Package 'SARP.moodle'

July 21, 2025

Type Package

Title XML Output Functions for Easy Creation of Moodle Questions

Version 1.2.3

Date 2025-06-18

Maintainer Emmanuel Curis <emmanuel.curis@parisdescartes.fr>

License Artistic-2.0

Description Provides a set of basic functions for creating Moodle XML output files suited for importing questions in Moodle (a learning management system, see <https://moodle.org/> for more information).

Suggests readODS,openxlsx

Depends base64enc,magick

Imports stats,utils

Encoding UTF-8

Language fr

NeedsCompilation no

Author Emmanuel Curis [aut, cre, cph] (ORCID: <https://orcid.org/0000-0001-8382-1493>), Virginie Lasserre [ctb]

Repository CRAN

Date/Publication 2025-06-25 11:50:07 UTC

Contents

affichage.moodle 2
annoter_image.moodle 5
categorie.moodle 7
csv.moodle 8
csv_glossaire.moodle 15
csv_optique.moodle 16
debuter_xml.moodle 19
definir_dossier.image.moodle 20

description.moodle 23
df.moodle 24
glisser_deposer.moodle 29
glisser_textes.moodle 32
glossaire.moodle 33
inserer_formule.moodle 35
inserer_SMILES.moodle 37
legender_image.moodle 39
libre.moodle 41
lier_fichier.moodle 44
messages.moodle 46
numerique.moodle 47
ouverte.moodle 49
qcm.moodle 51
qroc.moodle 54
question.moodle 55
SARP.moodle 58
sortie_R.moodle 59
vf.moodle 60

Index **63**

`affichage.moodle` *Aide aux affichages dans les textes des questions*

Description

Ces fonctions permettent de créer une chaîne de caractère au format HTML, qui peut ensuite être intégrée au texte d'une question (énoncé, réponse, commentaire...).

Usage

```
afficher_echantillon.moodle( x, tableau = TRUE, trier = FALSE,
                             n.chiffres = get( "nombre.chiffres",
                                               envir = SARP.Moodle.env ),
                             marge = c( 10, 10, 1, 1 ), ... )

afficher_echantillons.moodle( x, trier = FALSE,
                              n.chiffres = get( "nombre.chiffres",
                                                envir = SARP.Moodle.env ),
                              marge = c( 10, 10, 1, 1 ),
                              pre.texte = paste0( "\u00c9chantillon\u00a0",
                                                  1:length( x ) ),
                              couleur.trait = "Black", lg.trait = "2",
                              ... )

afficher_nombre.moodle( x,
                        dec = get( "decimal", envir = SARP.Moodle.env ),
```

```
                        n.chiffres = get( "nombre.chiffres",
                                          envir = SARP.Moodle.env ),
                        unite = "" )

afficher_poly.moodle( degre, variable, a, ... )
```

Arguments

x	L'élément à convertir au format HTML. Voyez les détails pour plus de précisions.
n.chiffres	Le nombre de chiffres à conserver pour l'affichage. Si le nombre est plus petit, en valeur absolue, que 10^-n.chiffres, il ne peut pas être affiché avec ce nombre de décimales : c'est alors le nombre de chiffres significatifs. Sinon, c'est le nombre de décimales.
trier	Si TRUE, les valeurs de l'échantillon sont triées par ordre croissant avant d'être converties en code HTML. S'il y a plusieurs échantillons, chacun est trié séparément.
marge	Les marges à laisser au sein de chaque case du tableau affichant les résultats. Ce doit être un vecteur de quatre valeurs entières, correspondant dans l'ordre aux marges gauche, droite, haute et basse, exprimées en pixels.
...	Autres options pour sortie_R.moodle.numeric, qui est appelé pour l'affichage sous forme de tableau.

Options pour afficher_echantillon.moodle

tableau	Si TRUE, un tableau HTML est construit. Sinon, une simple énumération des valeurs est préparée.

Options pour afficher_echantillons.moodle

pre.texte	Les textes à afficher au début de chaque ligne, permettant d'identifier les échantillons.
couleur.trait	La couleur des filets du tableau (ce doit être une couleur valable en CSS)
lg.trait	L'épaisseur des filets du tableau (exprimée en pixels)

Options pour afficher_nombre.moodle

dec	Le symbole à utiliser comme séparateur décimal. Par défaut, celui indiqué à la création du questionnaire.
unite	L'unité à utiliser pour l'affichage de la valeur.

Options pour afficher_poly.moodle

degre	Le degré du polynome.
variable	Le texte correspondant à la variable du polynome. Ce texte peut être quelconque et inclure des balises HTML. S'il correspond à une unique lettre minuscule (typiquement, x), les balises de mise en italique sont automatiquement ajoutées, afin de se conformer aux règles typographiques françaises.
a	Vecteur de longueur degre + 1 donnant les coefficients du polynôme, par degré décroissant.

Details

Pour `afficher_echantillon.moodle`, `x` peut être un vecteur ou une liste; chaque élément est converti en texte avec `afficher_nombre.moodle`. La liste des valeurs est alors créée, soit sous la forme (x1 ; x2 ; ...), soit sous la forme d'un tableau d'une ligne, avec un filet au-dessus et en-dessous.

Pour `afficher_echantillon.moodle`, `x` doit être une liste, dont chaque élément est un échantillon. Un tableau est alors créé, avec une ligne par échantillon (la ligne commençant par le texte fourni dans `pre.texte` et permettant d'identifier l'échantillon). Chaque valeur de chaque échantillon est convertie en texte avec `afficher_nombre.moodle`.

Pour `afficher_nombre.moodle`, `x` doit être un vecteur contenant au plus une seule valeur (dans le cas contraire, seule la première est utilisée, les autres sont ignorées avec un avertissement). S'il est de longueur nulle, le symbole de l'ensemble vide (entité `∅`) est renvoyé, mis en forme avec le style utilisé pour les messages d'erreur. Les valeurs manquantes (`NA`) ou non-numériques (`NaN`) sont affichées avec ces caractères, et le style des messages d'erreur. Les valeurs infinies (`Inf` et `-Inf`) sont affichées avec le symbole ∞ (entité `∞`). Les chaînes de caractère sont affichées inchangées. Enfin, les nombres sont mis en forme avec le symbole `dec` comme séparateur décimal, un signe moins correct (entité `−`) et, si nécessaire, une puissance de 10 correcte (soit ` × 10<sup>n</sup>`, où n est la puissance, pour avoir $\times 10^n$ avec des espacements corrects).

La fonction `afficher_poly.moodle` permet de créer le code HTML d'affichage d'un polynome de degré quelconque ; l'affichage se fait par puissances décroissantes ($a\,x + b$ pour un polynôme de degré 1, par exemple).

Value

Ces fonctions renvoient une unique chaîne de caractères, au format HTML, qui peut être utilisée pour construire le texte d'un énoncé, d'une réponse affichée, d'un commentaire...

Author(s)

Emmanuel Curis `<emmanuel.curis@parisdescartes.fr>`

Examples

```
## Not run:
# Commencer un nouveau questionnaire
q <- debuter_xml.moodle( "exemple.affichages" )

# Catégorie : exemples
categorie.moodle( "Exemples SARP/Affichages" )

# Signe moins, décimale
vrai_faux.moodle( "Quelle est l'écriture correcte de l'opposé de cinq huitièmes ?",
                  afficher_nombre.moodle( -5/8 ), -5/8 )

# Affichage des puissances
vrai_faux.moodle( paste( "L'inverse de", afficher_nombre.moodle( 10^8 ),
                         "est", afficher_nombre.moodle( 10^-8 ) ) )

# Affichage d'un échantillon
```

```
e <- rnorm( 10 ) * 2 + 20
numerique.moodle( paste0( "Donnez la plus petite valeur de l'échantillon ci-dessous.",
                          afficher_echantillon.moodle( e ) ),
                  min( e ) )

# Affichage de deux échantillons
e1 <- rnorm( 10 ) * 2 + 20
e2 <- rnorm( 5 ) * 1 + 5
d <- mean( e1 ) - mean( e2 )
numerique.moodle( paste0( "Donnez la différence des moyennes des deux échantillons ci-dessous.",
                          afficher_echantillons.moodle( list( e1, e2 ) ) ),
                  c( d, -d ), notes = c( 100, 100 ) )

# Affichage d'un polynôme
numerique.moodle( paste0( "Quelle est la solution de l'équation ",
                          afficher_poly.moodle( 1, 'x', c( 2, -3 ) ),
                          " = 0 ?" ),
                  3/2 )

# On termine le questionnaire
finir_xml.moodle( )

# Importez le fichier exemple.affichages.xml dans Moodle pour voir le résultat...
print(
"Importez le fichier exemple.affichages.xml dans Moodle pour voir le résultat..." )

## End(Not run)
```

---

annoter_image.moodle *Créer une question demande d'annoter une image*

Description

Cette fonction permet de créer une question affichant une image et des champs de texte à compléter pour annoter cette image.

Usage

```
annoter_image.moodle( texte, titre = "Annoter la figure...",
                      f.creer_figure,
                      fichier.image,
                      legendes,
                      x.numeros, y.numeros,
                      numeros = if ( superposer ) rep( "", length( legendes ) )
          else 1:length( legendes ),
                      superposer = all( !missing( x.numeros ),
                                        !missing( y.numeros ) ),
                      description.image = NULL, n.colonnes = 1,
                     commentaire.global = NA, penalite = NA, note.question = NA,
```

```
  idnum = NA, temps, tags = NULL,
fichier.xml = get( "fichier.xml", envir = SARP.Moodle.env ),
  ... )
```

Arguments

`texte`	L'énoncé de la question (une chaîne de caractères). Cet énoncé peut contenir n'importe quel caractère (codage en UTF-8) et n'importe quelles balises HTML.
`titre`	L'intitulé de la question, en HTML.
`f.creer_figure`	La fonction qui va construire l'image et les zones sur lesquelles déposer les éléments (voyez les détails, ci-dessous).
`fichier.image`	Le nom du fichier contenant l'image à légender. Ignoré si `f.creer_figure` est fournie.
`legendes`	Les textes à entrer dans les champs (réponses attendues)
`x.numeros`, `y.numeros`	Les coordonnées des points où placer le champ texte, sur l'image (exprimées en pixels). Si l'un, ou les deux, sont absents, les champs textes apparaîtront en-dessous de l'image. Si les deux sont présents, les champs apparaîtront sur l'image.
`numeros`	Le texte à faire apparaître avant chaque champ (typiquement, un numéro de champ)
`superposer`	Veut-on superposer les champs à l'image (`TRUE`) ou non (`FALSE`)?
`description.image`	Le texte alternatif à l'image à utiliser dans la balise HTML insérant l'image
`n.colonnes`	Si les champs sont placés sous l'image, les uns en-dessous des autres, sur combien de colonnes doivent-ils être répartis?
`commentaire.global`, `penalite`, `note.question`, `idnum`, `tags`	Options communes à toutes les questions Moodle, voyez `debut_question.moodle`.
`temps`	Le temps conseillé pour répondre à la question, en minutes.
`fichier.xml`	Le fichier correspondant au questionnaire qui contiendra cette question (par défaut, le dernier créé).
`...`	Arguments additionnels qui seront passés à `f.creer_figure`.

Details

À venir...

Value

Aucune valeur n'est renvoyée par cette fonction.

Author(s)

Emmanuel Curis <emmanuel.curis@parisdescartes.fr>

`categorie.moodle` *Créer une catégorie de questions*

Description

Cette fonction permet de créer une nouvelle catégorie de questions dans le questionnaire XML.

Usage

```
categorie.moodle( nom.categorie, autoriser.dollar = FALSE,
                  temps = NULL,
                  fichier.xml = get( "fichier.xml", envir = SARP.Moodle.env ) )
```

Arguments

`nom.categorie`	Le nom de la catégorie. Il est possible de donner une hiérarchie de catégories, en utilisant la barre oblique (`/`) comme séparateur des différents niveaux — comme dans un chemin incluant des dossiers.
`autoriser.dollar`	Le dollar (`$`) est un caractère réservé, qui permet d'ancrer les catégories dans les divers cours. Pour éviter des difficultés, les dollars sont donc remplacés par des soulignés (_). Cette conversion peut être désactivée en indiquant `TRUE` pour cette option.
`temps`	Le temps conseillé pour répondre aux questions de la catégorie. Ce temps sera utilisé pour toutes les questions créées jusqu'au prochain appel de cette fonction, si aucun temps conseillé n'est indiqué lors de l'appel à la fonction créant la fonction.
`fichier.xml`	Le fichier correspondant au questionnaire qui contiendra cette question (par défaut, le dernier créé).

Details

La catégorie proposée est toujours intégrée dans le cours actuel, en ajoutant `$course$` en début de hiérarchie.

Pour éviter des sous-catégories sans nom, les séparateurs multiples sont simplifiés en un seul séparateur et ceux en fin de nom de catégorie sont supprimés. Si vous voulez vraiment des sous-catégories sans nom visible, utilisez des espaces entre les séparateurs ou après le dernier.

Value

Aucune valeur n'est renvoyée par cette fonction.

Attention

Les catégories ne seront prises en compte que si l'option correspondante est cochée lors de l'import du fichier (case « Obtenir la catégorie à partir du fichier »)

Author(s)

Emmanuel Curis <emmanuel.curis@parisdescartes.fr>

Examples

```
## Not run:
# Commencer un nouveau questionnaire
q <- debuter_xml.moodle( "exemple.categorie" )

# Catégorie des questions numériques pour les exemples de cette documentation
categorie.moodle( "Exemples SARP/Numériques" )
numerique.moodle( "Combien vaut 1 + 1 ?", 2 )
numerique.moodle( "Combien vaut 1 &times 1 ?", 1 )
numerique.moodle( "Combien vaut 1 + 1 en binaire ?", 10 )

# Catégorie des QROC pour les exemples de cette documentation
categorie.moodle( "Exemples SARP/QROC" )
qroc.moodle( "Comment s'appelle cette biblioth&egrave;que ?", "SARP.Moodle" )

# On termine le questionnaire
finir_xml.moodle( )

# Importez le fichier exemple.categorie.xml dans Moodle pour voir le résultat...
print(
"Importez le fichier exemple.categorie.xml dans Moodle pour voir le résultat..." )

## End(Not run)
```

---

csv.moodle *Convertir un fichier CSV, ODS ou XLSX en questions Moodle*

Description

Cette fonction permet de convertir un fichier convenablement construit, en un fichier XML de questions pour Moodle

Usage

```
csv.moodle(fichier.csv,
           colonne.texte = NA, colonne.reponse = NA,
           colonne.note = NA, colonne.note_question = NA,
           colonne.titre = NA,
           colonne.code = NA, colonne.type = NA,
           colonne.retour = NA, colonne.global = NA,
           colonne.penalite = NA,
           colonne.temps = NA, colonne.decimale = NA,
           fichier.xml = if ( TRUE == nv.fichier ) gsub(
                              "\\.[Cc][Ss][Vv]$",
```

```
                              ".xml", fichier.csv )
                          else get( "fichier.xml",
                                    envir = SARP.Moodle.env ),
            nv.fichier = TRUE,
            creer.titre = TRUE, lg.titre = 30,
            embellir = TRUE, deja.HTML = FALSE,
            forcer.multiple = TRUE, melanger.reponses = TRUE,
            somme.nulle = FALSE, precision = 3,
            categorie.base = "",
            dossier.images = dirname( fichier.csv ),
            sep.images = c( '@@', '@@' ), inserer.images = TRUE,
            sep.formules = c( '@\\$', '\\$@' ),
            sep.SMILES = c( '@\\{', '\\}@' ),
            sep = if ( extension == "txt" ) "" else ";",
            header = TRUE, quote = '\"',
            ... )

ods.moodle( fichier.ods, onglet = NA,
            colonne.texte  = NA, colonne.reponse = NA,
            colonne.note   = NA, colonne.note_question = NA,
            colonne.titre  = NA, colonne.code = NA, colonne.type = NA,
            colonne.retour = NA, colonne.global = NA, colonne.penalite = NA,
            colonne.temps  = NA, colonne.decimale = NA,
            fichier.xml = if ( TRUE == nv.fichier ) gsub( "\\.[Oo][Dd][Ss]$",
                                                          ".xml",
                                                          fichier.ods )
                          else get( "fichier.xml", envir = SARP.Moodle.env ),
            nv.fichier = TRUE,
           creer.titre = TRUE, lg.titre = 30, embellir = TRUE, deja.HTML = FALSE,
           forcer.multiple = TRUE, melanger.reponses = TRUE, somme.nulle = FALSE,
            precision = 3,
            categorie.base = "",
            dossier.images = dirname( fichier.ods ),
            sep.images = c( '@@', '@@' ), inserer.images = TRUE,
            sep.formules = c( '@\\$', '\\$@' ),
            sep.SMILES = c( '@\\{', '\\}@' ),
            ... )

xlsx.moodle( fichier.xlsx, onglet = NA,
             colonne.texte  = NA, colonne.reponse = NA,
             colonne.note   = NA, colonne.note_question = NA,
             colonne.titre  = NA, colonne.code = NA, colonne.type = NA,
             colonne.retour = NA, colonne.global = NA, colonne.penalite = NA,
             colonne.temps  = NA, colonne.decimale = NA,
            fichier.xml = if ( TRUE == nv.fichier ) gsub( "\\.[Xx][Ll][Ss][Xx]$",
                                                          ".xml",
                                                          fichier.xlsx )
                          else get( "fichier.xml", envir = SARP.Moodle.env ),
```

```
  nv.fichier = TRUE,
creer.titre = TRUE, lg.titre = 30, embellir = TRUE, deja.HTML = FALSE,
forcer.multiple = TRUE, melanger.reponses = TRUE, somme.nulle = FALSE,
  precision = 3,
  categorie.base = "",
  dossier.images = dirname( fichier.xlsx ),
  sep.images = c( '@@', '@@' ), inserer.images = TRUE,
  sep.formules = c( '@\\$', '\\$@' ),
  sep.SMILES = c( '@\\{', '\\}@' ),
  ... )
```

Arguments

`fichier.csv`, `fichier.ods`, `fichier.xlsx`
: Un vecteur de chaînes de caractères contenant les noms des fichiers à convertir. Chacun des fichiers est traité indépendamment, mais avec les mêmes valeurs des options qui suivent.

`onglet`
: Le numéro, ou le nom, de la feuille (onglet) du classeur LibreOffice ou Excel à convertir. S'il n'est pas indiqué, toutes les feuilles seront converties.

`colonne.texte`
: Le nom ou le numéro de la colonne qui contient l'énoncé des questions.

`colonne.reponse`
: Le nom ou le numéro de la colonne qui contient la ou les réponses aux questions.

`colonne.note`
: Le nom ou le numéro de la colonne qui contient la note associée à chaque réponse. Si elle n'est pas indiquée, une colonne intitulée « Note » (insensible à la casse) est cherchée dans le fichier. Cette colonne est obligatoire en cas de question à réponses multiples. Elle peut alors contenir soit le pourcentage de la note (voir les détails), soit « Vrai » ou « Faux » (insensible à la casse) et les pourcentages sont calculés en conséquence (avec toujours 0 pour « Faux »: indiquez explicitement un pourcentage négatif si nécessaire).

`colonne.note_question`
: Le nom ou le numéro de la colonne qui contient la note globale de la question. Si elle n'est pas indiquée, une colonne intitulée « Note question » (insensible à la casse) est cherchée dans le fichier. Cette colonne est facultative. Si elle existe, elle doit contenir un entier strictement positif donnant la note globale de la question, ou rester vide. Toute note non indiquée sera supposée égale à 1 (valeur par défaut de Moodle).

`colonne.titre`
: Le nom ou le numéro de la colonne qui contient le titre à donner à la question dans Moodle. Si elle est manquante, le titre est construit, sous la forme `xx yy` où `yy` correspond aux `lg.titre` premiers caractères de la question, et `xx` au code de la question.

`colonne.code`
: Le nom ou le numéro de la colonne qui contient le code de la question. Cette colonne n'est obligatoire que pour créer des questions « cloze ». Si elle est manquante, un code interne de la forme `[Qnn]`, où `nn` est le numéro de la question, est construit, chaque question correspondant à une ligne avec les deux colonnes `colonne.texte` et `colonne.reponse` renseignées. Voyez les détails pour davantage de précisions.

`colonne.type` Le nom ou le numéro de la colonne qui contient le type de la question. Cette colonne peut être utilisée pour indiquer qu'une question avec plusieurs réponses proposées n'a qu'une seule réponse possible, en indiquant QCU pour au moins une des réponses (question à choix unique); QCM permet d'indiquer que l'on peut cocher plusieurs réponses (question à choix multiples). Dans ce cas, `forcer.multiple` est ignoré. QCI permet d'indiquer que l'on veut un QCM avec une notation de style « concours de l'Internat » — attention, cela crée un type de question spécial qui ne fonctionnera que si le plugin approprié est installé...

`colonne.retour` Le nom ou le numéro de la colonne qui contient le commentaire à afficher pour chaque réponse. Cette colonne est facultative.

`colonne.global` Le nom ou le numéro de la colonne qui contient le commentaire à afficher globalement pour la question. Cette colonne est facultative.

`colonne.penalite`
Le nom ou le numéro de la colonne qui contient la pénalité à mettre en cas de nouvelle tentative de la question. Si elle n'est pas indiquée, une colonne intitulée « Pénalité » (insensible à la casse) est cherchée dans le fichier. Cette colonne est facultative. Si elle existe, elle doit contenir la fraction de la note globale de la question qui sera prise en compte pour noter la nouvelle tentative (par exemple, si la question a une note globale de 2 et la pénalité est de 0,5, à la seconde tentative la note maximale possible sera de 1) ou rester vide. Toute pénalité non indiquée sera supposée égale à 1 (pas de pénalité en cas de nouvelle tentative).

`colonne.temps` Le nom ou le numéro de la colonne qui contient le temps conseillé pour la question. Cette colonne est facultative.

`colonne.decimale`
Le nom ou le numéro de la colonne qui contient le nombre de chiffres après la virgule attendu pour la réponse à la question (pour les questions numériques uniquement). Cette colonne est facultative.

`fichier.xml` Le nom du fichier XML à créer ou un fichier XML déjà créé avec `debuter_xml.moodle`. Par défaut, la fonction crée un fichier de même nom que le premier fichier fourni, en remplaçant l'extension `.csv` par `.xml`.

`nv.fichier` Une valeur logique indiquant si l'on veut créer un nouveau fichier (`TRUE`) ou utiliser un fichier XML déjà ouvert avec `debuter_xml.moodle` (`FALSE`)

`creer.titre` Une valeur logique indiquant si l'on veut créer un titre pour chaque question, à partir de son code et du début de son énoncé (`TRUE`) ou non. Ignoré si une colonne de titres a été fournie.

`lg.titre` Le nombre de caractères de l'énoncé d'une question à conserver pour construire son titre. Ignoré si une colonne de titres a été fournie ou si `creer.titre=FALSE`.

`embellir`, `deja.HTML`
Actuellement inutilisés, prévus pour une version future

`forcer.multiple`
Si `TRUE`, les questions avec plusieurs réponses sont toujours considérées comme des réponses à choix multiples (l'étudiant pourra cocher plusieurs réponses). Sinon, si une seule bonne réponse est proposée (plus exactement, s'il existe au moins une réponse donnant 100 % des points), la question est à choix unique (l'étudiant ne pourra choisir qu'une seule réponse). Cette option est ignorée pour les questions qui auraient une valeur indiquée dans la colonne précisée dans `colonne.type`.

`melanger.reponses`
: Si `TRUE`, autorise Moodle à permuter aléatoirement l'ordre des réponses lorsqu'il pose la question. Sinon, l'ordre du fichier est tout le temps utilisé.

`somme.nulle`
: Cette option précise comment gérer les réponses incorrectes dans un QCM. Si `FALSE`, la note de ces réponses est déterminée à partir du fichier, le code FAUX étant associé à une note nulle. Si `TRUE`, les réponses incorrectes se voient attribuer une note négative telle que la somme de ces notes vaut -1, de sorte que si l'étudiant coche toutes les cases du QCM, il a 0 (sauf bien sûr si toutes les cases correspondent à des questions correctes...).

`precision`
: Le nombre de décimales à donner dans la réponse, pour une réponse numérique (utilisez `NA` pour une valeur exacte et ne pas avoir de message indiquant le nombre de décimales ajouté au texte de la question).

`categorie.base`
: La catégorie de base à utiliser pour créer les questions. Elle contiendra toutes les sous-catégories du fichier, s'il y en a.

`dossier.images`
: Le dossier d'image où trouver les images à intégrer au fichier XML, ou l'URL de base pour les trouver dans Moodle (voir `definir_dossier.image.moodle`).

`sep.images`
: Les codes servant à encadrer un nom d'image dans les textes du fichier CSV.

`inserer.images`
: Si `TRUE`, les images sont incluses dans le fichier XML. Sinon, seul le lien est construit.

`sep.formules`
: Les codes servant à encadrer une formule mathématique à convertir grâce à latex dans les textes du fichier CSV.

`sep.SMILES`
: Les codes servant à encadrer un code SMILES à convertir en formule chimique grâce à openbabel dans les textes du fichier CSV.

`sep`, `header`, `quote`
: Options pour `read.table`, correspondant au format d'un fichier CSV par défaut lorsqu'il est créé par Libre office ou Excel.

`...`
: Autres options pour `read.table`, si nécessaire.

Details

Ces fonctions réalisent la conversion d'un ou plusieurs fichiers structurés de questions pour Moodle en un fichier XML. La nature des questions est déduite de la structure du fichier. Le fichier peut être au format CSV (format recommandé), Libre Office Calc (ODS) ou Excel (XLSX). Dans les deux derniers cas, il est possible de convertir toutes les feuilles du classeur ou juste une partie. Chaque feuille à convertir doir avoir la structure décrite ci-après, correspondant au format du fichier CSV.

Le fichier doit comporter au moins deux colonnes : l'une avec les énoncés des questions et l'autre avec les réponses. Chaque ligne correspond à une réponse possible.

Pour les questions simples, la question tient sur une ligne. La nature de la question est déduite de la réponse. Si la réponse peut être convertie en nombre, la question est supposée être une réponse numérique; elle sera créée avec un appel à `numerique.moodle`. Si la réponse est identifiée comme une valeur logique (un des textes « V », « F », « T », « Vrai », « Faux », « True » ou « False », quelle que soit la casse), la question est supposée être une question binaire avec les deux réponses « Vrai » et « Faux »; elle sera créée avec un appel à `vrai_faux.moodle`. Dans tous les autres cas, la question est supposée être à réponse ouverte et courte et sera créée par un appel à `qroc.moodle`. Par défaut, la réponse est sensible à la casse; cela peut être modifié en indiquant « I » dans la colonne précisant le type de question.

S'il y a un énoncé, mais pas de réponse, cet énoncé est supposé être une indication de catégorie (sauf si la question a le même code qu'une question « cloze », voir plus loin, ou si la colonne précisant le type de question indique un type particulier) qui sera créée comme une sous-catégorie de la catégorie de base, par un appel à `categorie.moodle`. Si une colonne précise le type, la question peut être de type rédactionnel (réponse libre dans un éditeur de texte, pas de correction automatique; type « R », `question_ouverte.moodle`), description (un texte s'affiche, mais aucune réponse n'est attendue; type « D », `description.moodle`) ou texte à trou (il faudra replacer les textes dans les trous de l'énoncé, les trous étant construits à partir des termes encadrés par des [[doubles crochets]]); type « T », `glisser_textes.moodle`)

S'il y a une réponse, mais pas d'énoncé, cette réponse est supposée être une autre réponse possible à une question à choix multiple, dont l'énoncé est dans la première ligne précédente qui en contient un. Dans ce cas, le fichier doit contenir une colonne qui contient les notes associées à chaque réponse. Cette note peut être précise, exprimée en pourcentage suivant la convention Moodle, ou indicative, avec la simple mention « Vrai » pour les bonnes réponses et « Faux » pour les mauvaises. Dans le deuxième cas, les bonnes réponses sont associées à l'inverse du nombre de bonnes réponses, de sorte que cocher toutes les bonnes réponses donne la note totale; les mauvaises réponses sont associées à la note 0 (si `somme.nulle=FALSE`: cocher ou non ces réponses ne change rien à la note) ou à l'opposé de l'inverse du nombre de mauvaises réponses (si `somme.nulle=TRUE`: cocher ces réponses diminue la note et cocher toutes les réponses donne une somme nulle). Par défaut, la question construite est toujours un QCM (plusieurs réponses peuvent être choisies), même s'il y a une seule bonne réponse. Pour demander qu'une seule réponse puisse être choisie (QCU), indiquez le type QCU dans la colonne indiquée par `colonne.type` pour la question considérée. Si vous voulez que toutes les questions avec une seule bonne réponse soient en QCU, vous pouvez aussi utiliser l'option `forcer.multiple = FALSE`.

Construire des questions "*cloze*"

Pour pouvoir construire des questions "*cloze*", le fichier doit obligatoirement contenir une colonne qui indique le code de chaque question (y compris les catégories). Ce code doit être unique pour une même question; toutes les lignes qui auront le même code seront donc considérées comme faisant partie de la même question.

De ce fait, si plusieurs lignes d'énoncé renseignées ont le même code, elles sont considérées faisant partie de la même question et sont utilisées pour construire une question "*cloze*", selon le même principe que la fonction `question_libre.moodle`: après chaque partie d'énoncé, un champ de réponse est créé, dont la bonne réponse est celle indiquée dans la colonne `colonne.reponse` et sert à définir le type de champ de réponse (selon la logique des questions en une ligne). Pour avoir un texte après le dernier champ de réponse, laissez la colonne `colonne.reponse` vide tout en gardant le même code de question.

Pour avoir des questions avec réponse à choisir dans une liste, mettez une réponse par ligne sans mettre d'énoncé avant, comme pour un QCM. La colonne `colonne.type` permet de préciser le type d'affichage, avec les codes Moodle (`MULTICHOICE`, `MULTICHOICE_S`...). Il est possible d'abréger le code: M correspond à un choix dans un menu déroulant (une seule réponse possible ; `MULTICHOICE`) ; H, à des réponses juxtaposés (`MULTICHOICE_H`) ; V, à des réponses superposées (`MULTICHOICE_V`) ; SA, à un champ libre attendant une réponse courte, de type Q. R. O. C. (`SHORTANSWER`). Pour les réponses superposées ou juxtaposés, par défaut, une seule réponse peut être choisie et l'ordre est celui indiqué dans le fichier. Il est possible d'indiquer que plusieurs réponses peuvent être choisies en ajoutant le code M (VM, HM) ; il est possible de demander à Moodle de tirer au sort l'ordre des réponses en ajoutant le code A (MA, VA, VMA...) ou S (MS, VS...). Pour le champ libre, par défaut la casse est ignorée. Il est possible d'indiquer que la casse doit être contrôlée grâce au code

SAC (SHORTANSWER_C). L'ordre des codes n'a pas d'importance (HM et MH donneront tous deux une question de type MULTICHOICE_H).

Insérer des images

Il est possible de faire référence à une image dans les textes (énoncés, réponses, commentaires) en encadrant le nom de fichier de l'image par les séparateurs choisis, par défaut @@. Il est possible de redimensionner l'image en faisant suivre le nom de l'image d'un point d'exclamation, puis de la largeur, du caractère x et de la hauteur. Attention, il semblerait que Moodle ne déforme pas les images, quelles que soient les dimensions demandées. Par défaut, l'image sera incluse au fichier XML.

Deux types d'image peuvent être créés au vol : les formules mathématiques et les formules chimiques. Voir les fonctions inserer_formule.moodle et inserer_SMILES.moodle pour plus de détails.

Value

De façon invisible, une liste des data.frames des questions après traitement de chaque fichier. Elle n'a guère d'autre intérêt que pour détecter l'origine d'un problème de conversion.

Attention!

Moodle est assez peu permissif pour les notes exprimées en pourcentage; en particulier, lorsque ce pourcentage ne tombe pas juste (33% pour trois bonnes réponses, par exemple), à cause de l'arrondi il peut rejeter le fichier créé. Dans ce cas, désactivez l'arrêt en cas d'erreur et demandez l'arrondi à la note la plus proche pour pouvoir importer les questions.

Author(s)

Emmanuel Curis <emmanuel.curis@parisdescartes.fr>

See Also

Les fonctions de base de création de questions pour plus de souplesse, et en particulier pour créer des questions non gérées par le format de fichier ci-dessus.

csv_optique.moodle pour convertir des fichiers CSV au format utilisé par les lecteurs optiques.

Examples

```
  # Conversion du fichier d'exemple fourni
  #  (à placer dans le répertoire de travail)
## Not run:
  csv.moodle( "exemple_Moodle.csv", colonne.code = "Code" )

## End(Not run)
```

csv_glossaire.moodle *Convertir un fichier CSV en glossaire Moodle*

Description

Cette fonction permet de convertir un fichier CSV, convenablement construit, en un fichier XML de glossaire pour Moodle

Usage

```
csv_glossaire.moodle( fichier.csv,
                      colonne.terme = "Mot",
                      colonne.definition = "D\u00e9finition",
                      fichier.xml = if ( TRUE == nv.fichier ) gsub(
                            "\\.[Cc][Ss][Vv]$",
                            ".xml", fichier.csv )
                                    else get( "fichier.xml",
                                              envir = SARP.Moodle.env ),
                      nv.fichier = TRUE,
                      embellir = TRUE, deja.HTML = FALSE,
                      sep = ";", header = TRUE, quote = "\"",
                      ... )
```

Arguments

fichier.csv	Un vecteur de chaînes de caractères contenant les noms des fichiers à convertir. Chacun des fichiers est traité indépendamment, mais avec les mêmes valeurs des options qui suivent.
colonne.terme	Le nom ou le numéro de la colonne qui contient les termes à définir dans le glossaire.
colonne.definition	Le nom ou le numéro de la colonne qui contient les définitions des termes du glossaire.
fichier.xml	Le nom du fichier XML à créer ou un fichier XML déjà créé avec creer_glossaire.moodle. Par défaut, la fonction crée un fichier de même nom que le premier fichier fourni, en remplaçant l'extension .csv par .xml.
nv.fichier	Une valeur logique indiquant si l'on veut créer un nouveau fichier (TRUE) ou utiliser un fichier XML déjà ouvert avec creer_glossaire.moodle (FALSE)
embellir, deja.HTML	Actuellement inutilisés, prévus pour une version future
sep, header, quote	Options pour read.table, correspondant au format d'un fichier CSV par défaut lorsqu'il est créé par Libre office ou Excel.
...	Autres options pour read.table, si nécessaire.

Details

Cette fonction réalise la conversion d'un ou plusieurs fichiers CSV de définitions d'un glossaire pour Moodle en un fichier XML.

Ce fichier doit comporter au moins deux colonnes : l'une avec les termes à définir et l'autre avec les définitions. Chaque ligne correspond à une définition.

Value

De façon invisible, une liste des data.frames des définitions après traitement de chaque fichier. Elle n'a guère d'autre intérêt que pour détecter l'origine d'un problème de conversion.

Author(s)

Emmanuel Curis <emmanuel.curis@parisdescartes.fr>

See Also

Les fonctions de base de création de glossaire pour plus de souplesse.

Examples

```
  # Conversion du fichier d'exemple fourni
  #  (à placer dans le répertoire de travail)
## Not run:
  csv_questionnaire.moodle( "exemple_glossaire.csv" )

## End(Not run)
```

---

csv_optique.moodle *Convertir un fichier CSV en questions Moodle*

Description

Cette fonction permet de convertir un fichier CSV, au format « lecture optique », en un fichier XML de questions pour Moodle

Usage

```
csv_optique.moodle( fichier.csv,
                    forcer.multiple = FALSE, melanger.reponses = TRUE,
                    somme.nulle = FALSE,
                    fichier.xml = if ( TRUE == nv.fichier )
                                    gsub( "\\.[Cc][Ss][Vv]$",
                                          ".xml", fichier.csv )
                                  else get( "fichier.xml",
                                            envir = SARP.Moodle.env ),
                    nv.fichier = TRUE,
```

```
                    embellir = TRUE, deja.HTML = FALSE,
                    categorie.base = "",
                    dossier.images = ".",
                    sep.images = c( '@@', '@@' ), inserer.images = TRUE,
                    sep.formules = c( '@\\$', '\\$@' ),
                    sep.SMILES = c( '@\\{', '\\}@' ),
                    sep = ";", header = TRUE, quote = '\"',
                    ... )
```

Arguments

`fichier.csv`	Un vecteur de chaînes de caractères contenant les noms des fichiers à convertir. Chacun des fichiers est traité indépendamment, mais avec les mêmes valeurs des options qui suivent.
`forcer.multiple`	Si `TRUE`, les questions sont toujours considérées comme des réponses à choix multiples (l'étudiant pourra cocher plusieurs réponses). Sinon, si une seule bonne réponse est proposée, la question est à choix unique (l'étudiant ne pourra choisir qu'une seule réponse).
`melanger.reponses`	Si `TRUE`, autorise Moodle à permuter aléatoirement l'ordre des réponses lorsqu'il pose la question. Sinon, l'ordre du fichier est tout le temps utilisé.
`somme.nulle`	Cette option précise comment gérer les réponses incorrectes du QCM. Si `FALSE`, les réponses incorrectes ont une note nulle. Si `TRUE`, les réponses incorrectes se voient attribuer une note négative telle que la somme de ces notes vaut -1, de sorte que si l'étudiant coche toutes les cases du QCM, il a 0 (sauf bien sûr si toutes les cases correspondent à des questions correctes...).
`fichier.xml`	Le nom du fichier XML à créer ou un fichier XML déjà créé avec `debuter_xml.moodle`. Par défaut, la fonction crée un fichier de même nom que le premier fichier fourni, en remplaçant l'extension `.csv` par `.xml`.
`nv.fichier`	Une valeur logique indiquant si l'on veut créer un nouveau fichier (`TRUE`) ou utiliser un fichier XML déjà ouvert avec `debuter_xml.moodle` (`FALSE`)
`embellir`, `deja.HTML`	Actuellement inutilisés, prévus pour une version future
`categorie.base`	La catégorie de base à utiliser pour créer les questions. Elle contiendra toutes les questions du fichier.
`dossier.images`	Le dossier d'image où trouver les images à intégrer au fichier XML, ou l'URL de base pour les trouver dans Moodle (voir `definir_dossier.image.moodle`).
`sep.images`	Les codes servant à encadrer un nom d'image dans les textes du fichier CSV.
`inserer.images`	Si `TRUE`, les images sont incluses dans le fichier XML. Sinon, seul le lien est construit.
`sep.formules`	Les codes servant à encadrer une formule mathématique à convertir grâce à latex dans les textes du fichier CSV.
`sep.SMILES`	Les codes servant à encadrer un code SMILES à convertir en formule chimique grâce à openbabel dans les textes du fichier CSV.

`sep`, `header`, `quote`
: Options pour `read.table`, correspondant au format d'un fichier CSV par défaut lorsqu'il est créé par Libre office ou Excel.

`...`
: Autres options pour `read.table`, si nécessaire.

Details

Ces fonctions réalisent la conversion d'un ou plusieurs fichiers CSV de questions pour Moodle en un fichier XML. Les questions sont nécessairement des QCM.

Ce fichier doit être au format utilisé par les lecteurs optiques de QCM. La première ligne, qui donne le numéro de question, est ignorée, ainsi que la première colonne. Chacune des autres colonnes correspond à une question différente.

Pour chaque question, la première ligne indique le titre, la suivante l'énoncé. Ensuite, les réponses possibles et leur justesse alternent.

Insérer des images

Il est possible de faire référence à une image dans les textes (énoncés, réponses, commentaires) en encadrant le nom de fichier de l'image par les séparateurs choisis, par défaut `@@`. Il est possible de redimensionner l'image en faisant suivre le nom de l'image d'un point d'exclamation, puis de la largeur, du caractère x et de la hauteur. Attention, il semblerait que Moodle ne déforme pas les images, quelles que soient les dimensions demandées. Par défaut, l'image sera incluse au fichier XML.

Deux types d'image peuvent être créés au vol : les formules mathématiques et les formules chimiques. Voir les fonctions `inserer_formule.moodle` et `inserer_SMILES.moodle` pour plus de détails.

Value

De façon invisible, une liste des data.frames des questions après traitement de chaque fichier. Elle n'a guère d'autre intérêt que pour détecter l'origine d'un problème de conversion.

Attention!

Moodle est assez peu permissif pour les notes exprimées en pourcentage; en particulier, lorsque ce pourcentage ne tombe pas juste (33% pour trois bonnes réponses, par exemple), à cause de l'arrondi il peut rejeter le fichier créé. Dans ce cas, désactivez l'arrêt en cas d'erreur et demandez l'arrondi à la note la plus proche pour pouvoir importer les questions.

Author(s)

Emmanuel Curis <emmanuel.curis@parisdescartes.fr> (sur une idée de Wilfrid Cariou)

See Also

Les fonctions de base de création de questions pour plus de souplesse, et en particulier pour créer des questions non gérées par le format de fichier ci-dessus.

`csv.moodle` pour convertir un fichier CSV au format proposé par SARP.moodle, permettant de gérer davantage de sortes de question.

Examples

```
  # Conversion du fichier d'exemple fourni
  #  (à placer dans le répertoire de travail)
## Not run:
  csv_optique.moodle( "exemple_optique.csv" )

## End(Not run)
```

`debuter_xml.moodle` *Commencer et terminer un questionnaire Moodle*

Description

Ces deux fonctions permettent de commencer et de terminer la génération d'un questionnaire Moodle au format XML.

Usage

```
debuter_xml.moodle( fichier.xml,
                    n.chiffres = 2, dec = ",",
                    racine = 2004197487,
                    glossaire = FALSE )

finir_xml.moodle(fichier.xml = get( "fichier.xml", envir = SARP.Moodle.env ))
```

Arguments

`fichier.xml`	Pour `debuter_xml.moodle`, le nom du fichier qui contiendra le questionnaire moodle au format XML. L'extension .xml est automatiquement rajoutée, sauf si elle est déjà présente dans ce nom. Pour `finir_xml.moodle`, le fichier correspondant au questionnaire à terminer (par défaut, le dernier créé).
`dec`	Le séparateur décimal à utiliser dans les textes. Par défaut, la virgule (comme utilisé en français). Il est modifié par `options`.
`n.chiffres`	Le nombre de décimales à utiliser pour les sorties au format XML. Attention, cela n'affecte que les valeurs numériques explicitement converties en texte par un appel à la fonction `afficher_nombre.moodle`.
`racine`	La racine (graine) à utiliser pour le générateur de nombres aléatoires, afin d'avoir une génération de questionnaires reproductible. Utilisez `NA` pour ne pas définir la graine.
`glossaire`	Si `TRUE`, le fichier XML créé est au format permettant de construire un glossaire. Sinon, c'est le format XML moodle « classique » qui est utilisé.

Details

Lorsque l'on débute un questionnaire avec `debuter_xml.moodle`, le fichier XML est créé avec son en-tête. Le descripteur de fichier est mémorisé dans l'environnement de la bibliothèque : toutes les fonctions ultérieures utiliseront ce fichier par défaut pour écrire les questions créées. Attention, en cas d'appels multiples pour travailler sur plusieurs questionnaires en parallèle, seul le dernier fichier créé est mémorisé.

Value

La fonction `debuter_xml.moodle` renvoie un descripteur de fichier correspondant au questionnaire nouvellement créé. Ce descripteur peut être utilisé par toutes les fonctions de la bibliothèque, mais aussi par la fonction `cat` de R directement, si nécessaire.

La fonction `finir_xml.moodle` ne renvoie rien.

Author(s)

Emmanuel Curis <emmanuel.curis@parisdescartes.fr>

See Also

`options( OutDec )` pour le choix du séparateur décimal à l'affichage.

Examples

```
## Not run:
# Commencer un nouveau questionnaire
q <- debuter_xml.moodle( "exemple" )

# Une question Moodle basique
vrai_faux.moodle( "Cette question a deux réponses" )

# On termine le questionnaire
finir_xml.moodle( )

# Importez le fichier exemple.xml dans Moodle pour voir le résultat...
print(
"Importez le fichier exemple.xml dans Moodle pour voir le résultat..." )

## End(Not run)
```

---

definir_dossier.image.moodle

Insérer une image dans une question

Description

Ces fonctions permettent de créer des liens vers des images dans les questions Moodle (texte, mais aussi réponses).

Usage

```
definir_dossier.image.moodle( URL, local = FALSE, ajouter = TRUE,
                              silencieux = FALSE )

lier_image.moodle( nom.image, largeur = -1, hauteur = -1,
                   description = NULL, interne = FALSE )

coder_image.moodle( nom.image,
                    dossier.image = get( "dossier.images",
                                         envir = SARP.Moodle.env ) )
```

Arguments

URL	L'URL du dossier ou de l'activité Moodle contenant les images à lier. Voyez les détails pour plus d'information. Ce doit être un vecteur de type character, ne contenant qu'un seul élément.
local	Précise si l'URL indiquée est locale (TRUE) ou sur le serveur Moodle (FALSE).
ajouter	Pour une URL locale, indique elle doit être ajoutée à la liste des dossiers de recherche des fichiers (TRUE) ou remplacer cette liste (FALSE).
silencieux	Si TRUE, les messages indiquant les URL enregistrées sont supprimés.
nom.image	Le nom du fichier contenant l'image. Si l'image est externe au fichier XML, ce doit être le nom tel qu'il apparaîtra dans le dossier moodle : attention aux conversions de caractères que peut faire Moodle lors de l'import d'un fichier... Ce doit être un vecteur de type character contenant un seul élément.
dossier.image	Le nom du dossier contenant l'image, en local.
largeur, hauteur	Les dimensions d'affichage de l'image dans la question (équivalent des champs height et width de la balise <img>). Seules les dimensions numériques, finies, positives sont prises en compte. Toutes les autres valeurs sont ignorées et le champ correspondant n'est pas créé dans le code HTML. Ces deux valeurs doivent être des vecteurs numériques contenant un seul élément.
description	Un vecteur de chaîne de caractère, ne contenant qu'un seul élément. S'il existe, ce texte sera utilisé comme description textuelle de l'image (champ alt, fortement conseillé).
interne	Indique si l'image est déjà sur Moodle (FALSE) ou si elle est incluse dans le fichier XML (TRUE). Dans le premier cas, l'URL de base doit avoir été définie par definir_dossier.image.moodle (voyez les détails, ci-après).

Details

La fonction lier_image.moodle crée une balise HTML <img> avec une URL permettant de trouver l'image. Cette URL est obtenue en concaténant une racine et le nom du fichier image indiqué.

Pour une image extérieure au fichier XML, la racine, définie avec la fonction definir_dossier.image.moodle, peut être n'importe quel schéma d'URL valide. Le plus souvent, ce sera soit une URL vers un site externe qui contient les images soit, encore plus fréquemment, une URL référençant l'image dans Moodle.

La méthode la plus simple pour obtenir cette racine de l'URL est la suivante :

1. Créez une activité « Dossier » dans Moodle, cachée aux utilisateurs.
2. Dans ce dossier, importez la ou les images, puis enregistrez.
3. Affichez le contenu de ce dossier, comme un utilisateur.
4. Affichez le code source de la page HTML correspondance (Ctrl-U avec Firefox)
5. Dans ce code source, repérez le nom d'une image, dans un élément `<img src="">`. Ce nom devrait ressembler à https://votre.site.moodle/pluginfile.php/00000/mod_folder/content/0/votre_image.jpg?forcedownloa où 0000 est un nombre quelconque qui identifie ce dossier dans la base Moodle.
6. Mémorisez cette URL et utilisez-la comme base:

   ```
   definir_dossier.image.moodle("https://votre.site.moodle/pluginfile.php/0000/mod_folder/content/
   ```

Toutes les images situées dans ce dossier (au moment de l'import du fichier XML ou ajoutées ultérieurement) sont alors directement utilisables dans la fonction `lier_image.moodle`, en indiquant juste leur nom.

Il est aussi possible d'inclure l'image dans le fichier XML. Pour cela, précisez `interne = TRUE` lors de l'appel à `lier_image.moodle`. Lors de la création de la question, les balises ainsi générées seront identifiées et les images encodées dans le fichier XML, avec la fonction `coder_image.moodle`. Attention cependant, les chemins d'accès locaux étant perdus, les fichiers contenant les images doivent se trouver dans le répertoire de travail au moment de la création de la question. Il est cependant possible d'indiquer un ou plusieurs dossiers locaux d'images différents, avec `definir_dossier.image.moodle`, en précisant `local = TRUE`. Si un fichier correspondant à l'image est trouvé dans plusieurs de ces dossiers, seule le premier est utilisé avec un avertissement. Les dossiers sont cherchés dans l'ordre d'ajout à la liste des dossiers de recherche, le dernier ajouté étant exploré en premier; un appel à `definir_dossier.image.moodle` pour un dossier déjà dans la liste le replace en haut de la liste. La liste des dossiers peut être remplacée par un seul dossier avec l'option `ajouter = FALSE` de `definir_dossier.image.moodle`.

Value

La fonction `lier_image.moodle` renvoie une chaîne de caractère qui contient le code HTML créé. Cette chaîne peut ensuite être utilisée, directement ou après concaténation avec d'autres éléments, dans la constitution des textes ou des réponses de n'importe quelle question moodle.

La fonction `coder_image.moodle` renvoie une chaîne de caractère qui contient le code XML permettant d'intégrer l'image dans le fichier XML, y compris son encodage en base 64. Cette chaîne peut ensuite être utilisée dans la constitution des élements d'une question moodle. Attention cependant, savoir où le mettre dans le fichier XML n'est pas trivial: normalement, l'utilisation de cette fonction ne devrait pas être nécessaire, si `lier_image.moodle` a été utilisée correctement.

Attention

Vous devez appeler au moins une fois la fonction `definir_dossier.image.moodle` pour pouvoir ensuite lier des images externes au fichier XML.

Au moment de la création du fichier XML, il n'y a aucun moyen de vérifier que l'URL indiquée est correcte...

Astuce

Pour importer plusieurs images en une fois, vous avez tout intérêt à les regrouper dans une archive au format zip, que vous décompresserez après l'avoir importée dans le dossier moodle ciblé.

Author(s)

Emmanuel Curis <emmanuel.curis@parisdescartes.fr>

See Also

inserer_formule.moodle pour insérer une formule mathématique à l'aide d'un code latex; inserer_SMILES.moodle pour insérer une formule chimique à l'aide d'un code SMILES.

Examples

```
## Not run:
# Commencer un nouveau questionnaire
q <- debuter_xml.moodle( "exemple.image" )

# On crée une image quelconque
png( "essai.png", width = 640, height = 400 )
qqnorm( rnorm( 20 ), pch = 19 )
dev.off()

# On pose une question là-dessus
vrai_faux.moodle( paste0( "Le graphe ci-dessous est un diagramme de Henry",
                          "<br />\n",
                          lier_image.moodle( "essai.png", interne = TRUE ) ) )

# On termine le questionnaire
finir_xml.moodle( )

# Importez le fichier exemple.image.xml dans Moodle pour voir le résultat...
print( "Importez le fichier exemple.image.xml dans Moodle pour voir le résultat..." )

# Le code XML...
coder_image.moodle( "essai.png" )

## End(Not run)
```

---

description.moodle *Créer une question Moodle contenant simplement du texte*

Description

Cette fonction permet de créer une question de type description pour Moodle.

Usage

```
description.moodle( texte,
                    titre = "Description",
                    commentaire.global = NA,
                    idnum = NA, tags = NULL,
                   fichier.xml = get( "fichier.xml", envir = SARP.Moodle.env ) )
```

Arguments

texte	Le texte de la question (une chaîne de caractères). Cet énoncé peut contenir n'importe quel caractère (codage en UTF-8) et n'importe quelles balises HTML.
titre	L'intitulé de la question, en HTML.
fichier.xml	Le fichier correspondant au questionnaire qui contiendra cette question (par défaut, le dernier créé).
commentaire.global, idnum, tags	Option commune à toutes les questions Moodle, voyez debut_question.moodle.

---

df.moodle *Convertir une data.frame en questions Moodle*

Description

Cette fonction permet de convertir une data.frame, convenablement construit, en un fichier XML de questions pour Moodle

Usage

```
df.moodle( d,
           colonne.texte = NA, colonne.reponse = NA,
           colonne.note = NA, colonne.note_question = NA,
           colonne.titre = NA,
           colonne.code = NA, colonne.type = NA,
           colonne.retour = NA, colonne.global = NA,
           colonne.penalite = NA,
           colonne.temps = NA, colonne.decimale = NA,
           fichier.xml,
           creer.titre = TRUE, lg.titre = 30,
           embellir = TRUE, deja.HTML = FALSE,
           forcer.multiple = TRUE, melanger.reponses = TRUE,
           somme.nulle = FALSE, precision = 3,
           categorie.base = "",
           dossier.images,
           sep.images = c( '@@', '@@' ), inserer.images = TRUE,
           sep.formules = c( '@\\$', '\\$@' ),
           sep.SMILES = c( '@\\{', '\\}@' ),
           nom.fichier = deparse( substitute( d ) ) )
```

Arguments

d	La data.frame à convertir.
colonne.texte	Le nom ou le numéro de la colonne qui contient l'énoncé des questions. Si elle n'est pas indiquée, une colonne intitulée « Texte » est cherchée dans le fichier. Cette colonne est obligatoire.
colonne.reponse	Le nom ou le numéro de la colonne qui contient la ou les réponses aux questions. Si elle n'est pas indiquée, une colonne intitulée « Réponse » est cherchée dans le fichier. Cette colonne est obligatoire.
colonne.note	Le nom ou le numéro de la colonne qui contient la note associée à chaque réponse. Si elle n'est pas indiquée, une colonne intitulée « Note » (insensible à la casse) est cherchée dans le fichier. Cette colonne est obligatoire en cas de question à réponses multiples. Elle peut alors contenir soit le pourcentage de la note (voir les détails), soit « Vrai » ou « Faux » (insensible à la casse) et les pourcentages sont calculés en conséquence (avec toujours 0 pour « Faux » : indiquez explicitement un pourcentage négatif si nécessaire).
colonne.note_question	Le nom ou le numéro de la colonne qui contient la note globale de la question. Si elle n'est pas indiquée, une colonne intitulée « Note question » (insensible à la casse) est cherchée dans le fichier. Cette colonne est facultative. Si elle existe, elle doit contenir un entier strictement positif donnant la note globale de la question, ou rester vide. Toute note non indiquée sera supposée égale à 1 (valeur par défaut de Moodle).
colonne.titre	Le nom ou le numéro de la colonne qui contient le titre à donner à la question dans Moodle. Si elle est manquante, le titre est construit, sous la forme xx yy où yy correspond aux lg.titre premiers caractères de la question, et xx au code de la question.
colonne.code	Le nom ou le numéro de la colonne qui contient le code de la question. Cette colonne n'est obligatoire que pour créer des questions « cloze ». Si elle est manquante, un code interne de la forme [Qnn], où nn est le numéro de la question, est construit, chaque question correspondant à une ligne avec les deux colonnes colonne.texte et colonne.reponse renseignées. Voyez les détails pour davantage de précisions.
colonne.type	Le nom ou le numéro de la colonne qui contient le type de la question. Cette colonne peut être utilisée pour indiquer qu'une question avec plusieurs réponses proposées n'a qu'une seule réponse possible, en indiquant QCU pour au moins une des réponses (question à choix unique); QCM permet d'indiquer que l'on peut cocher plusieurs réponses (question à choix multiples). Dans ce cas, forcer.multiple est ignoré.
colonne.retour	Le nom ou le numéro de la colonne qui contient le commentaire à afficher pour chaque réponse. Cette colonne est facultative.
colonne.global	Le nom ou le numéro de la colonne qui contient le commentaire à afficher globalement pour la question. Cette colonne est facultative.
colonne.penalite	Le nom ou le numéro de la colonne qui contient la pénalité à mettre en cas de nouvelle tentative de la question. Si elle n'est pas indiquée, une colonne intitulée

« Pénalité » (insensible à la casse) est cherchée dans le fichier. Cette colonne est facultative. Si elle existe, elle doit contenir la fraction de la note globale de la question qui sera prise en compte pour noter la nouvelle tentative (par exemple, si la question a une note globale de 2 et la pénalité est de 0,5, à la seconde tentative la note maximale possible sera de 1) ou rester vide. Toute pénalité non indiquée sera supposée égale à 1 (pas de pénalité en cas de nouvelle tentative).

`colonne.temps` Le nom ou le numéro de la colonne qui contient le temps conseillé pour la question. Cette colonne est facultative. Lorsqu'elle est présente, et renseignée, un texte indiquant le temps conseillé pour répondre est ajouté à l'énoncé de la question.

`colonne.decimale`
Le nom ou le numéro de la colonne qui contient le nombre de chiffres après la virgule attendu pour la réponse à la question (pour les questions numériques uniquement). Cette colonne est facultative.

`fichier.xml` Le nom du fichier XML, créé avec `debuter_xml.moodle`, dans lequel créer les questions que contient la data.frame.

`creer.titre` Une valeur logique indiquant si l'on veut créer un titre pour chaque question, à partir de son code et du début de son énoncé (`TRUE`) ou non. Ignoré si une colonne de titres a été fournie.

`lg.titre` Le nombre de caractères de l'énoncé d'une question à conserver pour construire son titre. Ignoré si une colonne de titres a été fournie ou si `creer.titre=FALSE`.

`embellir`, `deja.HTML`
Actuellement inutilisés, prévus pour une version future

`forcer.multiple`
Si `TRUE`, les questions avec plusieurs réponses sont toujours considérées comme des réponses à choix multiples (l'étudiant pourra cocher plusieurs réponses). Sinon, si une seule bonne réponse est proposée (plus exactement, s'il existe au moins une réponse donnant 100 % des points), la question est à choix unique (l'étudiant ne pourra choisir qu'une seule réponse). Cette option est ignorée pour les questions qui auraient une valeur indiquée dans la colonne précisée dans `colonne.type`.

`melanger.reponses`
Si `TRUE`, autorise Moodle à permuter aléatoirement l'ordre des réponses lorsqu'il pose la question. Sinon, l'ordre du fichier est tout le temps utilisé.

`somme.nulle` Cette option précise comment gérer les réponses incorrectes dans un QCM. Si `FALSE`, la note de ces réponses est déterminée à partir du fichier, le code FAUX étant associé à une note nulle. Si `TRUE`, les réponses incorrectes se voient attribuer une note négative telle que la somme de ces notes vaut -1, de sorte que si l'étudiant coche toutes les cases du QCM, il a 0 (sauf bien sûr si toutes les cases correspondent à des questions correctes...).

`precision` Le nombre de décimales à donner dans la réponse, pour une réponse numérique (utilisez `NA` pour une valeur exacte et ne pas avoir de message indiquant le nombre de décimales ajouté au texte de la question).

`categorie.base` La catégorie de base à utiliser pour créer les questions. Elle contiendra toutes les sous-catégories du fichier, s'il y en a.

`dossier.images`	Le dossier d'image où trouver les images à intégrer au fichier XML, ou l'URL de base pour les trouver dans Moodle (voir `definir_dossier.image.moodle`).
`sep.images`	Les codes servant à encadrer un nom d'image dans les textes du fichier CSV.
`inserer.images`	Si `TRUE`, les images sont incluses dans le fichier XML. Sinon, seul le lien est construit.
`sep.formules`	Les codes servant à encadrer une formule mathématique à convertir grâce à latex dans les textes du fichier CSV.
`sep.SMILES`	Les codes servant à encadrer un code SMILES à convertir en formule chimique grâce à openbabel dans les textes du fichier CSV.
`nom.fichier`	Le nom à utiliser pour créer les titres et la catégorie de base des questions générées à partir de cette data.frame.

Details

Cette fonctions réalise la conversion d'une data.frame de questions pour Moodle en un fichier XML. La nature des questions est déduite de la structure de la data.frame.

Cette data.frame doit comporter au moins deux colonnes : l'une avec les énoncés des questions et l'autre avec les réponses. Chaque ligne correspond à une réponse possible.

S'il y a un énoncé, mais pas de réponse, cet énoncé est supposé être une indication de catégorie (sauf si la question a le même code qu'une question « cloze », voir plus loin, ou si la colonne précisant le type de question indique un type particulier) qui sera créée comme une sous-catégorie de la catégorie de base, par un appel à `categorie.moodle`.

Pour les questions simples, la question tient sur une ligne. La nature de la question est déduite de la réponse. Si la réponse peut être convertie en nombre, la question est supposée être une réponse numérique; elle sera créée avec un appel à `numerique.moodle`. Si la réponse est identifiée comme une valeur logique (un des textes « V », « F », « T », « Vrai », « Faux », « True » ou « False », quelle que soit la casse), la question est supposée être une question binaire avec les deux réponses « Vrai » et « Faux »; elle sera créée avec un appel à `vrai_faux.moodle`. Dans tous les autres cas, la question est supposée être à réponse ouverte et courte et sera créée par un appel à `qroc.moodle`.

S'il y a une réponse, mais pas d'énoncé, cette réponse est supposée être une autre réponse possible à une question à choix multiple, dont l'énoncé est dans la première ligne précédente qui en contient un. Dans ce cas, le fichier doit contenir une colonne qui contient les notes associées à chaque réponse. Cette note peut être précise, exprimée en pourcentage suivant la convention Moodle, ou indicative, avec la simple mention « Vrai » pour les bonnes réponses et « Faux » pour les mauvaises. Dans le deuxième cas, les bonnes réponses sont associées à l'inverse du nombre de bonnes réponses, de sorte que cocher toutes les bonnes réponses donne la note totale; les mauvaises réponses sont associées à la note 0 (si `somme.nulle=FALSE`: cocher ou non ces réponses ne change rien à la note) ou à l'opposé de l'inverse du nombre de mauvaises réponses (si `somme.nulle=TRUE`: cocher ces réponses diminue la note et cocher toutes les réponses donne une somme nulle). Par défaut, la question construite est toujours un QCM (plusieurs réponses peuvent être choisies), même s'il y a une seule bonne réponse. Pour demander qu'une seule réponse puisse être choisie (QCU), indiquez le type QCU dans la colonne indiquée par `colonne.type` pour la question considérée. Si vous voulez que toutes les questions avec une seule bonne réponse soient en QCU, vous pouvez aussi utiliser l'option `forcer.multiple = FALSE`.

Construire des questions "*cloze*"

Pour pouvoir construire des questions "*cloze*", le fichier doit obligatoirement contenir une colonne qui indique le code de chaque question (y compris les catégories). Ce code doit être unique pour une même question; toutes les lignes qui auront le même code seront donc considérées comme faisant partie de la même question.

De ce fait, si plusieurs lignes d'énoncé renseignées ont le même code, elles sont considérées faisant partie de la même question et sont utilisées pour construire une question "*cloze*", selon le même principe que la fonction `question_libre.moodle`: après chaque partie d'énoncé, un champ de réponse est créé, dont la bonne réponse est celle indiquée dans la colonne `colonne.reponse` et sert à définir le type de champ de réponse (selon la logique des questions en une ligne). Pour avoir un texte après le dernier champ de réponse, laissez la colonne `colonne.reponse` vide tout en gardant le même code de question.

Pour avoir des questions avec réponse à choisir dans une liste, mettez une réponse par ligne sans mettre d'énoncé avant, comme pour un QCM. La colonne `colonne.type` permet de préciser le type d'affichage, avec les codes Moodle (`MULTICHOICE`, `MULTICHOICE_S`...). Il est possible d'abréger le code: M correspond à un choix dans un menu déroulant (une seule réponse possible ; `MULTICHOICE`) ; H, à des réponses juxtaposés (`MULTICHOICE_H`) ; V, à des réponses superposées (`MULTICHOICE_V`) ; SA, à un champ libre attendant une réponse courte, de type Q. R. O. C. (`SHORTANSWER`). Pour les réponses superposées ou juxtaposés, par défaut, une seule réponse peut être choisie et l'ordre est celui indiqué dans le fichier. Il est possible d'indiquer que plusieurs réponses peuvent être choisies en ajoutant le code M (VM, HM) ; il est possible de demander à Moodle de tirer au sort l'ordre des réponses en ajoutant le code A (MA, VA, VMA...) ou S (MS, VS...). Pour le champ libre, par défaut la casse est ignorée. Il est possible d'indiquer que la casse doit être contrôlée grâce au code SAC (`SHORTANSWER_C`). L'ordre des codes n'a pas d'importance (HM et MH donneront tous deux une question de type `MULTICHOICE_H`).

Insérer des images

Il est possible de faire référence à une image dans les textes (énoncés, réponses, commentaires) en encadrant le nom de fichier de l'image par les séparateurs choisis, par défaut `@@`. Il est possible de redimensionner l'image en faisant suivre le nom de l'image d'un point d'exclamation, puis de la largeur, du caractère x et de la hauteur. Attention, il semblerait que Moodle ne déforme pas les images, quelles que soient les dimensions demandées. Par défaut, l'image sera incluse au fichier XML.

Deux types d'image peuvent être créés au vol : les formules mathématiques et les formules chimiques. Voir les fonctions `inserer_formule.moodle` et `inserer_SMILES.moodle` pour plus de détails.

Value

De façon invisible, une liste des data.frames des questions après traitement de chaque fichier. Elle n'a guère d'autre intérêt que pour détecter l'origine d'un problème de conversion.

Attention!

Moodle est assez peu permissif pour les notes exprimées en pourcentage; en particulier, lorsque ce pourcentage ne tombe pas juste (33% pour trois bonnes réponses, par exemple), à cause de l'arrondi il peut rejeter le fichier créé. Dans ce cas, désactivez l'arrêt en cas d'erreur et demandez l'arrondi à la note la plus proche pour pouvoir importer les questions.

Author(s)

Emmanuel Curis <emmanuel.curis@parisdescartes.fr>

See Also

Les fonctions de base de création de questions pour plus de souplesse, et en particulier pour créer des questions non gérées par le format de fichier ci-dessus.

Examples

```
  # Conversion du fichier d'exemple fourni
  #  (à placer dans le répertoire de travail)
## Not run:
  csv.moodle( "exemple_Moodle.csv", colonne.code = "Code" )

## End(Not run)
```

---

glisser_deposer.moodle

Créer une question demandant de placer des éléments à des positions précises d'une image

---

Description

Cette fonction permet de créer une question dans laquelle le but est de placer des éléments sur une image.

Usage

```
glisser_deposer.moodle( texte, titre = "Glisser-d\u00e9poser...",
                        f.creer_figure,
                        fichier.image,
                        x.zones, y.zones, txt.zones,
                        indications = paste0( "Zone ", 1:n.zones ),
                        img.zones = NULL, zone.unique = TRUE,
                        grp.zones = rep( 1, n.zones ),
                        ordre.aleatoire = TRUE,
                        commentaire.global = NA,
                        penalite = NA, note.question = NA, idnum = NA,
                        temps, tags = NULL,
                        fichier.xml = get( "fichier.xml",
                                           envir = SARP.Moodle.env ),
                        ... )
```

Arguments

`texte`	L'énoncé de la question (une chaîne de caractères). Cet énoncé peut contenir n'importe quel caractère (codage en UTF-8) et n'importe quelles balises HTML.
`titre`	L'intitulé de la question, en HTML.
`f.creer_figure`	La fonction qui va construire l'image et les zones sur lesquelles déposer les éléments (voyez les détails, ci-dessous).
`fichier.image`	Le nom du fichier contenant l'image à légender. Ignoré si `f.creer_figure` est fournie.
`x.zones`, `y.zones`	Les coordonnées des coins supérieurs, gauches des zones à définir sur l'image. Ignoré si `f.creer_figure` est fournie.
`txt.zones`	Les textes à utiliser pour les marques à placer sur les zones (dans le même ordre que les coordonnées). Ignoré si `f.creer_figure` est fournie.
`indications`	Les textes à afficher sur les zones de l'image pour les identifier.
`img.zones`	Les noms des fichiers contenant les images à utiliser pour chaque marque. Ignoré si `f.creer_figure` est fournie.
`zone.unique`	Un vecteur aussi long que le nombre de zones, indiquant si l'étiquette associée peut être réutilisée après avoir été placée dans une zone (`FALSE`) ou non (`TRUE`). Ignoré si `f.creer_figure` est fournie.
`grp.zones`	Un vecteur d'entiers aussi long que le nombre de zones, indiquant à quel groupe d'étiquette appartient chacune des étiquettes. L'apparence de l'étiquette (forme, couleur) dépend de ce groupe. Ignoré si `f.creer_figure` est fournie.
`ordre.aleatoire`	Si `TRUE`, l'ordre des marques est permuté aléatoirement dans le fichier
`commentaire.global`, `penalite`, `note.question`, `idnum`, `tags`	Options communes à toutes les questions Moodle, voyez `debut_question.moodle`.
`temps`	Le temps conseillé pour répondre à la question, en minutes.
`fichier.xml`	Le fichier correspondant au questionnaire qui contiendra cette question (par défaut, le dernier créé).
`...`	Arguments additionnels qui seront passés à `f.creer_figure`.

Details

Ce type de question affiche une image et, en-dessous, des étiquettes textuelles ou formées d'une image à placer sur l'image. Les étiquettes ont toutes la même apparence (taille, forme, couleur). Les zones où les placer sont indiquées sur l'image en surimpression et font la même taille que les étiquettes. Il est possible de créer des groupes d'étiquettes (mais ce n'est intéressant que s'il y a au moins deux étiquettes par groupe, sans quoi la taille de la zone affichée sur l'image donne la réponse...); la forme et la couleur des étiquettes varie d'un groupe à l'autre. Attention, les étiquettes d'un groupe donné ne peuvent pas être placées sur les zones des étiquettes des autres groupes. Par défaut, toutes les étiquettes appartiennent au premier groupe (étiquettes rectangulaires, blanches). La taille des étiquettes est identique dans un même groupe, définie pour pouvoir contenir le texte ou l'image le plus grand du groupe.

Utilisation de la fonction `f.creer_figure`

Avec cette méthode, la fonction se charge de créer le fichier d'image (au format PNG, dans le dossier temporaire de R) avant d'appeler `f.creer_figure`. La fonction `f.creer_figure` peut alors utiliser tous les outils graphiques basiques de R pour créer une figure et les zones à légender sur cette figure. Elle *ne* doit *pas* clore la figure (pas d'appel à `dev.off()` ou à `graphics.off`).

La fonction `f.creer_figure` doit renvoyer une `data.frame` dont chaque ligne correspond à une zone, et contenant au moins trois colonnes:

- 'X' et 'Y', indiquant les coordonnées du coin supérieur, gauche de la zone, exprimées dans le système d'unité natif du graphe créé avec R (elles seront ensuite converties en pixels grâce aux fonctions `grconvertX` et `grconvertY`).;
- 'Texte', indiquant le texte de l'étiquette associée à la zone.

La taille de la zone sera déterminée par Moodle en fonction de la taille de l'étiquette associée à la zone.

De façon optionnelle, il peut y avoir une colonne 'Indications' qui contient une indication textuelle sur la zone, une colonne 'Images' qui contient le nom du fichier contenant l'image à associer à la zone, une colonne 'Groupes' qui précise à quelle groupe d'étiquette appartient l'étiquette associée à la zone et une colonne 'Unique' qui indique si l'étiquette disparaît des propositions après avoir été placée (`TRUE`) ou si elle reste, pouvant alors être placée sur plusieurs zones (`FALSE`). Si ces colonnes sont absentes, l'indication est "Zone n°" et le numéro de la zone (de sa ligne dans la `data.frame`), toutes les étiquettes ne contiennent que du texte, sont placées dans le même groupe et ne sont utilisables qu'une seule fois.

Il est possible de créer des étiquettes ne correspondant à aucune zone ("distracteurs") en indiquant `NA` pour leurs coordonnées.

Utilisation directe

Il est aussi possible de préciser une image quelconque, grâce à l'option `fichier.image`. Dans ce cas, les options `x.zones`, `y.zones`, `txt.zones`, `indications`, `img.zones`, `grp.zones` et `zone.unique` permettent de préciser les caractéristiques de chaque zone à définir. Les coordonnées doivent être indiquées en pixels.

Value

Aucune valeur n'est renvoyée par cette fonction.

Author(s)

Emmanuel Curis <emmanuel.curis@parisdescartes.fr>

See Also

`legender_image.moodle` pour une autre façon de demander à placer des éléments sur une image; `glisser_textes.moodle` pour faire glisser des textes dans un texte à trou.

`glisser_textes.moodle` *Créer une question demandant de replacer des mots dans un texte*

Description

Cette fonction permet de créer une question dans laquelle il faut faire glisser des mots pour les replacer dans un texte à trous.

Usage

```
glisser_textes.moodle( texte, titre = "Glisser les textes...",
                       groupe = rep( 1, n.zones ),
                       infini = FALSE,
                       distracteurs = NULL,
                       ordre.aleatoire = TRUE, afficher.erreurs = TRUE,
                       commentaire.global = NA,
                       penalite = NA, note.question = NA, idnum = NA,
                       temps, tags = NULL,
                       fichier.xml = get( "fichier.xml",
                                          envir = SARP.Moodle.env ) )
```

Arguments

`texte`	L'énoncé de la question (une chaîne de caractères). Cet énoncé peut contenir n'importe quel caractère (codage en UTF-8) et n'importe quelles balises HTML. Les trous à faire dans ce texte doivent être identifiés par des doubles-crochets (voir les détails).
`titre`	L'intitulé de la question, en HTML.
`groupe`	Pour chaque élément à replacer, le groupe auquel il appartient (voir les détails). L'ordre est l'ordre d'apparition dans le texte.
`infini`	Pour chaque élément à replacer, indique s'il peut être réutilisé (`TRUE`) ou s'il disparaît des éléments une fois placé quelque part (`FALSE`). L'ordre est l'ordre d'apparition dans le texte. Si une seule valeur est indiquée, elle sera utilisée pour tous les éléments. Sinon, il faut indiquer autant de valeurs que d'éléments.
`distracteurs`	Une liste de termes n'apparaissant pas dans le texte, permettant de générer des étiquettes ne correspondant à aucun trou (distracteurs). Ce peut être un vecteur de chaînes de caractères (tous les distracteurs seront dans le groupe 1) ou une data.frame avec trois colonnes : `Textes` contenant le texte à mettre dans l'étiquette, `Groupe` indiquant le numéro du groupe auquel appartient le distracteur et `Infini` indiquant s'il peut être réutilisé (`TRUE`) ou non (`FALSE`).
`ordre.aleatoire`	Les éléments sont par défaut proposés dans l'ordre de leur numéro. Ce numéro est ici imposé à l'ordre d'apparition dans le texte. Cette option permet, si elle vaut `TRUE` de permuter aléatoirement ces numéros.

`afficher.erreurs`
: Si `TRUE`, en cas de réponse incorrecte à la question, les réponses erronées sont identifiées.

`commentaire.global`, `penalite`, `note.question`, `idnum`, `tags`
: Options communes à toutes les questions Moodle, voyez `debut_question.moodle`.

`temps`
: Le temps conseillé pour répondre à la question, en minutes.

`fichier.xml`
: Le fichier correspondant au questionnaire qui contiendra cette question (par défaut, le dernier créé).

Details

Cette question permet de proposer un texte dans lequel certains éléments sont remplacés par un cadre vide, dans lequel doivent être glissés des éléments.

Dans l'énoncé fourni, les éléments qui devront être remplacés par des cadres vides doivent être entourés par des doubles crochets — par exemple, le texte "Le roi [[Henri IV]] de France et de Navarre" apparaîtra dans Moodle avec un trou à la place de "[[Henri IV]]" et, sous le texte, une étiquette intitulée "Henri IV", qu'il faudra replacer.

Les différents éléments à replacer peuvent être organisés en groupes, qui se distinguent par une couleur légèrement différente et une taille des étiquettes différentes.

Afin de ne pas donner d'indication, les trous (et leurs étiquettes) auront tous la même taille au sein d'un groupe, choisie pour que le plus long texte tienne dans l'étiquette.

Value

Aucune valeur n'est renvoyée par cette fonction.

Author(s)

Emmanuel Curis <emmanuel.curis@parisdescartes.fr>

See Also

`glisser_deposer.moodle` et `legender_image.moodle` pour faire glisser des étiquettes sur une image.

---

glossaire.moodle *Commencer et terminer la génération de glossaires Moodle*

Description

Ces fonctions permettent de préparer et de remplir un glossaire Moodle au format XML.

Usage

```
creer_glossaire.moodle( nom.fichier, nom.glossaire, texte.intro,
                        doublons = TRUE )

entree_glossaire.moodle( terme, definition,
                    fichier.xml = get( "fichier.xml", envir = SARP.Moodle.env ) )
```

Arguments

nom.fichier	Une chaîne de caractères indiquant le nom du fichier XML, contenant le glossaire, à créer.
nom.glossaire	Une chaîne de caractères indiquant le nom du glossaire.
texte.intro	Une chaîne de caractères indiquant le texte décrivant le glossaire.
doublons	Une valeur logique indiquant si les doublons sont autorisés.
terme	Une chaîne de caractères indiquant le terme à insérer dans le glossaire.
definition	Une chaîne de caractères indiquant la définition du terme à insérer dans le glossaire.
fichier.xml	Le fichier correspondant au glossaire qui contiendra cette entrée (par défaut, le dernier créé).

Details

Ces fonctions permettent de préparer un glossaire Moodle au format XML.

Value

Aucune valeur n'est renvoyée par cette fonction.

Attention

Le support des glossaires au format XML est expérimental, en l'absence de documentation officielle.

Author(s)

Emmanuel Curis <emmanuel.curis@parisdescartes.fr>

`inserer_formule.moodle` *Insérer une formule mathématique, comme image, grâce à latex*

Description

Cette fonction permet de convertir une formule mathématique au format Latex en image prête à insérer dans un texte.

Usage

```
inserer_formule.moodle( formule, filtre.Moodle = FALSE,
                        displaystyle = TRUE, marge = 2,
                        couleurs = TRUE, enjoliver = TRUE,
                        cmd.latex = getOption( "Sm.cmd_latex" ),
                        packages = c( "amsfonts", "amsmath",
                                      "amssymb" , "dsfont" ),
                        options.convert = list( 'density' = 150,
                                                'outext'  = ".png" ) )
```

Arguments

`formule`	Une chaîne de caractères contenant la formule mathématique à convertir, au format latex (sans les délimiteurs).
`filtre.Moodle`	Si `TRUE`, la formule est entourée des marqueurs permettant à Moodle de faire la conversion, en utilisant son filtre (qui va dépendre de la configuration faite sur le serveur). Cela a l'avantage de faire une base plus légère (pas d'image...) et un affichage plus harmonieux, mais aussi plus lent et avec moins de souplesse, puisque les bibliothèques disponibles sont fonction de l'installation faite. Sinon, elle est convertie en image localement avec latex et l'image est insérée dans le fichier XML.
`displaystyle`	Si `TRUE`, la commande latex `\displaystyle` est ajoutée en début de formule.
`marge`	La marge à laisser autour de la formule (en points).
`couleurs`	Si `TRUE`, le package "xcolor" est inclus et les noms de couleurs usuels sont utilisables.
`enjoliver`	Si `TRUE`, quelques adaptations de la formule sont faites pour avoir un résultat plus esthétique. Actuellement, cela correspond à protéger la virgule entre deux chiffres par des accolades (évitant ainsi une espace après la virgule) et à remplacer les symboles inégalité large par défaut ("\leq" et "\geq") par leur version avec l'égalité parallèle au signe ("\leqslant" et "\geqslant").
`cmd.latex`	Une chaîne de caractères indiquant la commande à utiliser pour lancer latex. Par défaut, `"latex -shell-escape -interaction=batchmode -halt-on-error"` qui suppose que latex est accessible dans le chemin par défaut des applications. Attention, une commande sans les options indiquées peut entrer dans une boucle d'attente infinie en cas d'erreur dans une formule latex.

`packages` Un vecteur de chaînes de caractères indiquant les noms des packages latex à inclure

`options.convert` Une liste d'options à passer au package "standalone", utilisé pour créer l'image.

Details

Cette fonction permet de créer une image d'une formule mathématique, à partir du code latex correspondant.

La taille de l'image dépend de la longueur de la formule, mais aussi de la définition choisie pour créer l'image. Il n'est pas très facile de prévoir la qualité du résultat, puisque la taille du texte dans Moodle dépend, elle, de la feuille de style choisie et des réglages du navigateur. L'option "density" (par défaut à 150) permet de modifier cette taille: plus elle est élevée, plus l'image sera grande.

La conversion est faite avec latex et le package "standalone": les deux doivent être installés et latex exécutable par R pour que la conversion puisse avoir lieu.

Les commandes des formules latex commencent par le caractère \, qui sert de code d'échappement dans R. Il faut donc les protéger en les doublant (\\) dans les formules.

Value

Une chaîne de caractère contenant le code XML nécessaire pour inclure une image (contenant la formule créée), et qu'elle soit incluse au fichier XML au moment de la création de la question.

Author(s)

Emmanuel Curis <emmanuel.curis@parisdescartes.fr>

See Also

`inserer_SMILES.moodle` pour convertir un code SMILES en une formule chimique et l'insérer dans une question Moodle.

`lier_image.moodle` pour insérer une image quelconque.

Examples

```
  # Création d'une question demandant la définition de la tangente
  #  (Non exécuté car latex non nécessairement installé)
## Not run:
  # Commencer un nouveau questionnaire
  q <- debuter_xml.moodle( "tangente" )

  # Catégorie : exemples
  categorie.moodle( "Exemples SARP/Latex" )

  # Le format par défaut, avec deux bonnes réponses et trois mauvaises,
  qcm.moodle( "Quelle est la définition de la tangente ?",
              inserer_formule.moodle( "\\frac{\\sin x}{\\cos x}" ),
              inserer_formule.moodle( "\\frac{\\cos x}{\\sin x}" ) )

  # On termine le questionnaire
```

```
  finir_xml.moodle( )

  # Importez le fichier tangente.xml dans Moodle pour voir le résultat...
  print( "Importez le fichier exemple.xml dans Moodle pour voir le résultat..." )

## End(Not run)
```

`inserer_SMILES.moodle` *Insérer une formule chimique, comme image, grâce à openbabel*

Description

Cette fonction permet de convertir une formule chimique définie par un code SMILES en une image prête à insérer dans un texte.

Usage

```
inserer_SMILES.moodle( code.SMILES, nom.molecule = code.SMILES,
                       largeur = 300, hauteur = 300,
                       couleur.atomes = TRUE,
                       couleur.fond = NA,
                       couleur.liaisons = NA,
                       double.liaisons.asymetrique = FALSE,
                       masquer.terminaux = TRUE,
                       dessiner.CH = FALSE,
                       marges = FALSE,
                       cmd.obabel = "obabel " )
```

Arguments

`code.SMILES`	Une chaîne de caractères contenant le code SMILES à convertir.
`nom.molecule`	Une chaîne de caractère qui sera utilisée comme description de l'image, dans le code HTML produit pour Moodle.
`largeur, hauteur`	Les dimensions de l'image, en pixels
`couleur.atomes`	Faut-il utiliser le code couleur habituel pour indiquer les atomes? (Option "-xu" d'open babel.)
`couleur.fond`	Couleur à utiliser pour le fond de l'image (Option "-xb" d'open babel; `NA`: fond par défaut.)
`couleur.liaisons`	Couleur à utiliser pour les liaisons chimiques (`NA`: fond par défaut d'open babel)
`double.liaisons.asymetrique`	Option "-xs" d'open babel.
`masquer.terminaux`	Faut-il afficher en toutes lettres les méthyles terminaux? (Option "-xC" d'open babel.)

`dessiner.CH`	Faut-il afficher en toutes lettres les atomes de carbone? (Option "-xa" d'open babel.)
`marges`	Si `FALSE`, les marges autour de la formule sont éliminées (option "-xm" d'open babel). Comme les marges dépendent aussi des dimensions d'image demandées et de la formule créée, open babel peut laisser des marges assez grandes malgré tout, aussi l'image est rognée grâce à l'utilitaire "convert" (option "-trim"), qui doit donc aussi être installé et exécutable par R.
`cmd.obabel`	Une chaîne de caractères indiquant la commande à utiliser pour lancer open babel

Details

Cette fonction permet de créer une image d'une formule chimique semi-développée, à partir du code SMILES correspondant.

La conversion est faite avec open babel, qui doit être installé et exécutable par R pour que la conversion puisse avoir lieu. Par défaut, pour ne pas avoir de marges, l'utilitaire convert est aussi utilisé.

Value

Une chaîne de caractère contenant le code XML nécessaire pour inclure une image (contenant la formule créée), et qu'elle soit incluse au fichier XML au moment de la création de la question.

Author(s)

Emmanuel Curis <emmanuel.curis@parisdescartes.fr>

See Also

`inserer_formule.moodle` pour convertir un code latex en une formule mathématique et l'insérer dans une question Moodle.

`lier_image.moodle` pour insérer une image quelconque.

Examples

```
  # Création d'une question demandant quelle est la formule du benzène.
  #  (Non exécuté car open babel et convert non nécessairement installés)
## Not run:
  # Commencer un nouveau questionnaire
  q <- debuter_xml.moodle( "benzene" )

  # Catégorie : exemples
  categorie.moodle( "Exemples SARP/SMILES" )

  # Le format par défaut, avec deux bonnes réponses et trois mauvaises,
  qcm.moodle( "Quelle est la formule du benzène ?",
              inserer_SMILES.moodle( "c1ccccc1" ),
              inserer_SMILES.moodle( "C1CCCCC1" ) )

  # On termine le questionnaire
```

```
  finir_xml.moodle( )

  # Importez le fichier benzene.xml dans Moodle pour voir le résultat...
  print( "Importez le fichier benzene.xml dans Moodle pour voir le résultat..." )

## End(Not run)
```

`legender_image.moodle` *Créer une question demandant de placer des éléments à des positions précises d'une image*

Description

Cette fonction permet de créer une question dans laquelle le but est de placer des éléments sur une image.

Usage

```
legender_image.moodle( texte, titre = "L\u00e9gender...",
                       f.creer_figure,
                       fichier.image,
                       zones, marques,
                       ordre.aleatoire = TRUE, afficher.erreurs = TRUE,
                       commentaire.global = NA, penalite = NA,
                       note.question = NA, idnum = NA,
                       temps, tags = NULL,
                       fichier.xml = get( "fichier.xml",
                                          envir = SARP.Moodle.env ),
                       ... )
```

Arguments

`texte`	L'énoncé de la question (une chaîne de caractères). Cet énoncé peut contenir n'importe quel caractère (codage en UTF-8) et n'importe quelles balises HTML.
`titre`	L'intitulé de la question, en HTML.
`f.creer_figure`	La fonction qui va construire l'image et les zones sur lesquelles déposer les éléments (voyez les détails, ci-dessous).
`fichier.image`	Le nom du fichier contenant l'image à légender. Ignoré si `f.creer_figure` est fournie.
`zones`	Une liste décrivant les zones de la figure. Le format est celui utilisé par f.creer_figure, voyez les détails ci-dessous; les coordonnées doivent cependant être données directement en pixels. Ignoré si `f.creer_figure` est fournie.
`marques`	Une `data.frame` décrivant les marques à placer sur la figure. Le format est celui utilisé par f.creer_figure, voyez les détails ci-dessous. Ignoré si `f.creer_figure` est fournie.

`ordre.aleatoire`
: Si `TRUE`, l'ordre des marques est permuté aléatoirement dans le fichier

`afficher.erreurs`
: Si `TRUE`, en cas de réponse incorrecte à la question, les réponses erronées sont identifiées.

`commentaire.global`, `penalite`, `note.question`, `idnum`, `tags`
: Options communes à toutes les questions Moodle, voyez `debut_question.moodle`.

`temps`
: Le temps conseillé pour répondre à la question, en minutes.

`fichier.xml`
: Le fichier correspondant au questionnaire qui contiendra cette question (par défaut, le dernier créé).

`...`
: Arguments additionnels qui seront passés à `f.creer_figure`.

Details

Ce type de question affiche une image et, en-dessous, des étiquettes textuelles (code HTML possible) à placer sur l'image. Le placement de l'étiquette est défini par son coin supérieur gauche (qui est affiché sous forme de cible quand on la déplace). Les zones où placer les étiquettes sont invisibles et couvrent une surface libre de l'image, qui peut être rectangulaire, circulaire ou définie à l'aide d'un polygone.

Utilisation de la fonction `f.creer_figure`

Avec cette méthode, la fonction se charge de créer le fichier d'image (au format PNG, dans le dossier temporaire de R) avant d'appeler `f.creer_figure`. La fonction `f.creer_figure` peut alors utiliser tous les outils graphiques basiques de R pour créer une figure et les zones à légender sur cette figure. Elle *ne* doit *pas* clore la figure (pas d'appel à `dev.off()` ou à `graphics.off`.

La fonction `f.creer_figure` doit renvoyer une liste de deux éléments:

- un élément nommé `Marques` qui définit les marques à placer sur la figure;
- un élément nommé `Zones` qui définit les zones possibles pour placer ces marques.

L'élément `Zones` doit être une liste qui contient autant d'éléments que de zones à définir. Chaque élément est lui-même une liste qui doit contenir

- un élément nommé `Type` qui définit, sous forme d'une chaîne de caractères, la forme géométrique de la zone. Moodle connaît trois types de forme: rectangulaire (chaîne à utiliser: "Rectangle"), circulaire ("Cercle") et polygonale ("Polygone").
- un élément nommé `Coord`, qui doit être une data.frame contenant au moins deux colonnes appelées `X` et `Y`, et autant de lignes que de coordonnées nécessaires pour définir la zone. Pour une zone rectangulaire, la première ligne définit le coin supérieur, gauche et la seconde ligne le coin inférieur droit. Pour une zone polygonale, chaque ligne définit un sommet du polygone (il n'est pas nécessaire de répéter les coordonnées du premier sommet à la fin). Pour une zone circulaire, une seule ligne est nécessaire, définissant le centre du cercle. Le rayon du cercle doit être donné dans une troisième colonne, nommée `Rayon`.

Les coordonnées doivent être données en unités "utilisateur", c'est-à-dire les coordonnées naturelles de l'image (celles des échelles des axes). La conversion en pixels sera faite automatiquement, à l'aide des fonctions `grconvertX` et `grconvertY`. En général, les échelles ne sont pas les mêmes pour les deux axes, de ce fait la conversion du rayon d'un cercle ne conduit pas à la même valeur suivant qu'il est considéré parallèle à l'axe des abscisses ou à l'axe des ordonnées (ce qui revient à

dire qu'un cercle serait représenté par une ellipse sur la figure). La plus grande de ces deux valeurs est utilisée pour définir le rayon, en pixels, de la zone à créer. Si vous voulez contrôler parfaitement la forme de la zone, vous devez forcer le graphe à avoir les mêmes échelles dans les deux dimensions (option `asp = TRUE` dans `plot`).

L'élément `Marques` doit être une `data.frame` qui contient autant de lignes que de marques à placer. Cette `data.frame` doit avoir au moins une colonne nommée "`Marque`", qui contient le texte de chaque marque (tout code HTML est possible et sera mis en forme par Moodle). Par défaut, chaque marque ne peut être utilisée qu'une fois. Pour préciser un nombre d'utilisation plus important, la colonne "`Nombre`" peut être utilisée (y compris avec la valeur `+Inf`).

Utilisation directe

Il est aussi possible de préciser une image quelconque, grâce à l'option `fichier.image`. Dans ce cas, les options `zones` et `marques` doivent contenir, respectivement, les zones à définir et les marques à placer, en utilisant le format ci-dessus. Seule différence, les coordonnées des points définissant les zones (et le rayon, pour une zone circulaire) doivent être exprimés directement en pixels.

Value

Aucune valeur n'est renvoyée par cette fonction.

Author(s)

Emmanuel Curis <emmanuel.curis@parisdescartes.fr>

See Also

`glisser_deposer.moodle` pour une autre façon de demander à placer des éléments sur une image; `glisser_textes.moodle` pour faire glisser des textes dans un texte à trou.

---

libre.moodle *Créer une question libre (« cloze », texte à compléter)*

Description

Cette fonction permet de créer une question dont le format est libre, qui se présente sous forme d'un texte à trous — format dit « cloze ».

Usage

```
question_libre.moodle( texte.intro, textes.avant, texte.final,
                       reponses, notes = NULL, types = NULL,
                       commentaires = NULL,
                       titre = "Question libre",
                       commentaire.global = NA,
                       penalite = NA, note.question = NA, idnum = NA,
                       temps, tags = NULL,
                       fichier.xml = get( "fichier.xml",
                                          envir = SARP.Moodle.env ) )
```

Arguments

`texte.intro`	L'introduction de la question (une chaîne de caractères). Cet énoncé peut contenir n'importe quel caractère (codage en UTF-8) et n'importe quelles balises HTML. Il se trouvera au tout début de la question.
`textes.avant`	Un vecteur de chaîne de caractères. À chaque élément de ce vecteur sera associé un champ de réponse, qui sera *précédé* du texte contenu dans cette chaîne de caractères. Ces textes peuvent contenir n'importe quel caractère (codage en UTF-8) et n'importe quelles balises HTML. Le nombre d'éléments du vecteur doit être égal au nombre d'éléments de `reponses`.
`texte.final`	Le texte qui doit se placer après le dernier champ de réponse de la question. Il peut contenir n'importe quel caractère (codage en UTF-8) et n'importe quelles balises HTML.
`reponses`	Une liste dont chaque élément décrit la ou les réponses attendues pour chaque « trou » du texte. Le nombre d'éléments de cette liste définit le nombre de champs réponses dans la question et doit être égal à la longueur du vecteur `textes.avant`. Consultez les détails pour plus d'information.
`notes`	Le vecteur des notes associées à chaque réponse. S'il est fourni, il doit être de même longueur que `reponses`.
`types`	Le vecteur des types de champ réponse. S'il est fourni, il doit être de même longueur que `reponses`. Les types possibles sont les textes `NUMERICAL` (par défaut), `MULTICHOICE`, `MULTICHOICE_V`, `MULTICHOICE_H`, `MULTICHOICE_S`, `MULTICHOICE_VS`, `MULTICHOICE_HS`, `MULTIRESPONSE`, `MULTIRESPONSE_S`, `MULTIRESPONSE_H`, `MULTIRESPONSE_S`, `SHORTANSWER` et `SHORTANSWER_C`. Consultez les détails pour plus d'information.
`commentaires`	Une liste de commentaires associés aux réponses de chaque champ réponse. Si elle est fournie, elle doit être de même longueur que `reponses`. Consultez les détails pour plus d'information.
`titre`	L'intitulé de la question, en HTML.
`commentaire.global`, `penalite`, `note.question`, `idnum`, `tags`	Options communes à toutes les questions Moodle, voyez `debut_question.moodle`.
`temps`	Le temps conseillé pour répondre à la question, en minutes.
`fichier.xml`	Le fichier correspondant au questionnaire qui contiendra cette question (par défaut, le dernier créé).

Details

Cette fonction construit, à partir des éléments fournis, une question de type « cloze » : un texte à trous, les réponses étant à entrer dans les trous ou à choisir dans une liste. C'est donc une suite de « champs réponses », séparés par des textes.

Chaque champ réponse est défini par la ou les réponses possibles, son type, sa note et éventuellement les commentaires renvoyés à l'étudiant en fonction de ses réponses. Ces informations sont obtenues à partir des paramètres ci-dessus, le *i*-ème élément étant utilisé pour le *i*-ème champ réponse.

Le type `NUMERICAL` correspond à un champ attendant une réponse numérique — c'est l'équivalent d'une question numérique. Une ou plusieurs réponses sont possibles, qui seront de bonnes réponses, mais pas forcément donnant toute tous les points. Les réponses sont à fournir sous la forme d'une

liste comportant un vecteur des valeurs, nommé `Valeur` et un vecteur des tolérances associées, nommé `Tolerance`.

Les types `MULTICHOICE`, `MULTICHOICE_V`, `MULTICHOICE_H`, `MULTICHOICE_S`, `MULTICHOICE_VS`, `MULTICHOICE_HS`, `MULTIRESPONSE`, `MULTIRESPONSE_S`, `MULTIRESPONSE_S`, `MULTIRESPONSE_HS`, `SHORTANSWER` et `SHORTANSWER_C` correspondent à un champ permettant de choisir (cocher) une ou plusieurs réponses dans une liste prédéfinie — c'est l'équivalent d'un QCM. Tous ces types fonctionnent de façon similaire, seuls la façon d'afficher la question et le nombre de réponses sélectionnables varient (voyez le tableau récapitulatif ci-dessous). Dans tous les cas, `reponses` doit contenir pour ces champs une liste de deux éléments, l'un contenant le vecteur des textes des réponses (élément nommé `Textes`), le second indiquant pour chaque texte s'il correspond à une réponse correcte (`TRUE`) ou non (`FALSE`). S'il y a plusieurs réponses correctes, chacune donne une fraction égale des points. Pour préciser la note de chaque réponse, placez dans la liste un élément nommé `Notes`, contenant le vecteur des notes de chaque réponse (ces notes doivent être exprimées en pourcentage de la note, arrondi à l'entier le plus proche, parmi les valeurs autorisées par Moodle) ; dans ce cas, l'élément `Correct` est ignoré et peut être omis. Les commentaires de chaque réponse sont à donner dans un vecteur de même longueur que `reponses$Textes`; utilisez `NA` pour une réponse sans commentaire.

Code	Disposition	Remarque
`MULTICHOICE`	Menu déroulant	Une seule réponse possible
`MULTICHOICE_V`	Série de boutons radio verticaux	Une seule réponse
`MULTICHOICE_H`	Série de boutons radio horizontaux	Une seule réponse
`MULTIRESPONSE`	Série de cases à cocher verticales	Plusieurs réponse possibles
`MULTICHOICE_H`	Série de cases à cocher horizontales	Plusieurs réponses possibles
`SHORTANSWER`	Champ de texte libre	sensible à la casse
`SHORTANSWER_C`	Champ de texte libre	insensible à la casse

Les variantes avec le suffixe S conduisent à un ordre des réponses tiré au sort par Moodle lors de l'affichage de la question.

Une fois le texte complet de la question créé, au format « cloze », à partir de ces éléments, la question est créée grâce à la fonction `question.moodle`.

Value

Aucune valeur n'est renvoyée par cette fonction.

Author(s)

Emmanuel Curis <emmanuel.curis@parisdescartes.fr>

See Also

`numerique.moodle`, `qcm.moodle`, `qroc.moodle`, `question_ouverte.moodle`, `vrai_faux.moodle`

Examples

```
## Not run:
# Commencer un nouveau questionnaire
q <- debuter_xml.moodle( "exemple.libre" )
```

```
# Catégorie : exemples
categorie.moodle( "Exemples SARP/Question libre" )

# Une question Moodle libre avec champs numériques
question_libre.moodle( "Indiquez les chiffres du nombre 1974.",
                       list( "Unit&eacute; : ", " &mdash; dizaine : ",
                             " &mdash; centaine : ", " &mdash; milliers : " ), ".",
                       reponses = list( 4, 7, 9, 1 ) )

# Avec un champ numérique et des champs à choix multiples
question_libre.moodle( "<cite>Le Corbeau et le Renard</cite>.",
                       list( "<i>Ma&icirc;tre ",
                             " sur son arbre perch&eacute;<br />Tenait en son bec un ",
                             ".<br />Ma&icirc;tre ",
                             paste0( "</i>par l'odeur all&eacute;ch&eacute;<br />...",
                                     "<br />est un po&egrave;me compos&eacute; en " ) ),
                       " par le po&egrave;te Jean de la Fontaine.",
                       types = c( "MULTICHOICE", "MULTICHOICE_H",
                                  "SHORTANSWER", "NUMERICAL" ),
                       reponses = list( list( "Textes"  = c( "Renard", "Goupil",
                                                             "Corbeau", "Corneille" ),
                                              "Correct" = c( FALSE, FALSE, TRUE, FALSE ) ),
                                        list( "Textes"  = c( "Fromage", "Camembert",
                                                             "Bleu", "Livre" ),
                                              "Correct" = c( TRUE, FALSE, FALSE, FALSE ) ),
                                        list( "Textes"  = c( "Renard", "Goupil",
                                                             "Corbeau", "Corneille" ),
                                              "Correct" = c( TRUE, FALSE, FALSE, FALSE ) ),
                                        1668 ) )

# On termine le questionnaire
finir_xml.moodle( )

# Importez le fichier exemple.libre.xml dans Moodle pour voir le résultat...
print(
"Importez le fichier exemple.libre.xml dans Moodle pour voir le r\u00e9sultat..." )

## End(Not run)
```

`lier_fichier.moodle` *Insérer un lien vers un fichier dans une question*

Description

Cette fonction permet de créer un liens vers un fichier à télécharger dans les questions Moodle (texte, mais aussi réponses).

Usage

```
lier_fichier.moodle( nom.fichier, texte.lien = NULL, interne = TRUE )
```

Arguments

nom.fichier	Le nom du fichier vers lequel pointera le lien. Si le fichier est externe au fichier XML, ce doit être le nom tel qu'il apparaîtra dans le dossier moodle : attention aux conversions de caractères que peut faire Moodle lors de l'import d'un fichier... Ce doit être un vecteur de type character contenant un seul élément.
texte.lien	Un vecteur de chaîne de caractère, ne contenant qu'un seul élément, indiquant le texte à afficher comme lien vers le fichier. S'il est absent, le nom du fichier entre crochets est utilisé.
interne	Indique si le fichier est déjà sur Moodle (FALSE) ou si elle est incluse dans le fichier XML (TRUE). Dans le premier cas, l'URL de base doit avoir été définie par definir_dossier.image.moodle (voyez les détails, ci-après).

Details

La fonction lier_fichier.moodle crée une balise HTML <a href=> avec une URL pointant vers le fichier. Cette URL est obtenue en concaténant une racine et le nom du fichier indiqué.

Pour un fichier extérieur au fichier XML, la racine, définie avec la fonction definir_dossier.image.moodle, peut être n'importe quel schéma d'URL valide. Le plus souvent, ce sera soit une URL vers un site externe qui contient les fichiers soit, encore plus fréquent, une URL référençant le fichier dans Moodle. Il faut alors préciser interne = FALSE lors de l'appel à lier_fichier.moodle

Par défaut, le fichier est inclus dans le fichier XML. Lors de la création de la question, les balises ainsi générées seront identifiées et les fichiers encodés dans le fichier XML, avec la fonction coder_image.moodle. Attention cependant, les chemins d'accès locaux étant perdus, les fichiers doivent se trouver dans le répertoire de travail au moment de la création de la question. Il est cependant possible d'indiquer un dossier local différent, avec definir_dossier.image.moodle, en précisant local = TRUE.

Value

La fonction lier_fichier.moodle renvoie une chaîne de caractère qui contient le code HTML créé. Cette chaîne peut ensuite être utilisée, directement ou après concaténation avec d'autres éléments, dans la constitution des textes ou des réponses de n'importe quelle question moodle.

Attention

Vous devez appeler au moins une fois la fonction definir_dossier.image.moodle pour pouvoir ensuite lier des fichiers externes au fichier XML. Au moment de la création du fichier XML, il n'y a aucun moyen de vérifier que l'URL indiquée est correcte...

Astuce

Pour importer plusieurs fichiers en une fois, sans les inclure au fichier XML, vous avez tout intérêt à les regrouper dans une archive au format zip, que vous décompresserez après l'avoir importée dans le dossier moodle ciblé.

Author(s)

Emmanuel Curis <emmanuel.curis@parisdescartes.fr>

See Also

inserer_formule.moodle pour insérer une formule mathématique à l'aide d'un code latex; inserer_SMILES.moodle pour insérer une formule chimique à l'aide d'un code SMILES ; lier_image.moodle pour insérer une image (balise img)

---

messages.moodle *Aide aux affichages dans les textes des questions*

Description

Ces fonctions permettent de créer une chaîne de caractère au format HTML, pour des messages courant d'indication dans les questions.

Usage

```
temps_necessaire.moodle( temps, couleur = getOption( "Sm.temps_couleur" ),
                         nv.ligne = TRUE,
                         masque = getOption( "Sm.temps_masque" ) )
```

Arguments

temps	Le temps conseillé pour la question. Ce peut être un nombre, le temps est alors supposé exprimé en minutes, ou une chaîne de caractères à un format de temps classique (par exemple 45s, 2'30...).
couleur	La couleur à utiliser pour le message. N'importe quelle couleur reconnue en CSS peut être utilisée. Par défaut, utilise la valeur dans l'option Sm.temps_couleur (qui correspond à une consigne en bleue si elle n'est pas modifiée).
nv.ligne	Si TRUE, le message est précédé d'un retour à la ligne (balise HTML).
masque	Le masque à utiliser pour la consigne indiquant le temps nécessaire; dans ce masque, le code #T sera remplacé par le temps construit. Par défaut, utilise la valeur dans l'option Sm.temps_masque (qui correspond à "Temps conseillé pour répondre : #T").

Details

Ces fonctions sont utilisées par les fonctions générant les questions classiques pour ajouter des consignes en fin d'énoncé. Elles peuvent être utilisées directement pour davantage de souplesse.

Le message créé par temps_necessaire.moodle est par défaut "Temps conseillé pour cette question : x min y s", où x et y sont construits à partir du temps indiqué en minutes (par exemple, si temps=1.5, x=1 et y=30). Pour les questions classiques, il correspond à l'option temps. Ce message peut être modifié grâce à l'option Sm.temps_masque, qui doit contenir une chaîne de caractère. Dans cette chaîne, #T sera remplacé par "x min y s".

Value

Ces fonctions renvoient une unique chaîne de caractères, en HTML, prête à être intégrée dans un énoncé.

Author(s)

Emmanuel Curis <emmanuel.curis@parisdescartes.fr>

Examples

```
# Indication d'un temps conseillé de 2 min 30
temps_necessaire.moodle( 2.5 )

# Indication d'un temps conseillé de 10 secondes
temps_necessaire.moodle( "10s" )
```

---

numerique.moodle *Créer une question à réponse numérique simple*

Description

Cette fonction permet de créer une question dont la réponse est une valeur numérique unique.

Usage

```
numerique.moodle( texte, bonne.reponse, notes = 100,
                  n.decimales = get( "nombre.chiffres", envir = SARP.Moodle.env ),
                    n.significatifs = NA,
                    titre = "Question num&eacute;rique...",
                    commentaire.global = NA, penalite = NA, note.question = NA,
                    idnum = NA,
                    tolerance.type = 2, tolerance = "auto",
                    unites = NULL, unite.avant = FALSE,
                    unite.penalite = 0.1, unite.visible = FALSE,
                    commentaires = NULL,
                    couleur.consigne = getOption( "Sm.arrondi_couleur" ),
                    temps, tags = NULL,
                    fichier.xml = get( "fichier.xml", envir = SARP.Moodle.env ) )
```

Arguments

texte	L'énoncé de la question (une chaîne de caractères). Cet énoncé peut contenir n'importe quel caractère (codage en UTF-8) et n'importe quelles balises HTML.
bonne.reponse	La valeur numérique correspondant à la bonne réponse. Depuis Moodle 3, plusieurs valeurs peuvent être proposées.
notes	Lorsque plusieurs valeurs peuvent être acceptées comme réponse, la note (en fraction de la note totale de la question) associée à chaque réponse.
n.decimales	Le nombre de décimales à conserver pour la bonne réponse (utiliser NA pour ne pas arrondir); par défaut, la valeur indiquée à la création du questionnaire est utilisée.

`n.significatifs`
: Le nombre de chiffres significatifs à conserver pour la bonne réponse (utiliser `NA` pour ne pas arrondir); par défaut, la valeur indiquée à la création du questionnaire est utilisée.

`titre`
: L'intitulé de la question, en HTML.

`fichier.xml`
: Le fichier correspondant au questionnaire qui contiendra cette question (par défaut, le dernier créé).

`commentaire.global`, `penalite`, `note.question`, `idnum`, `tags`
: Options communes à toutes les questions Moodle, voyez `debut_question.moodle`.

`tolerance.type`
: La façon dont est exprimée la tolérance. Les valeurs possibles sont 1 (tolérance relative: la valeur de la tolérance est un pourcentage de la réponse), 2 (tolérance absolue) et 3 (tolérance « géométrique »).

`tolerance`
: La tolérance associée à chaque réponse. Si une seule valeur est fournie, elle sera utilisée pour toutes les réponses. La valeur spéciale `"auto"` calcule une tolérance de $1{,}1\times10^{-n}$, où n est le nombre de décimales demandé pour l'arrondi si `n.decimales > 0`, et 0 sinon.

`unites`
: Un vecteur de multiplicateurs indiquant les conversions possible d'unités pour la réponse. Les noms de ce vecteurs définissent les unités.

`unite.avant`
: Si `TRUE`, l'unité sera affichée et attendue avant la valeur numérique (pour les unités monétaires anglosaxonnes, typiquement). Sinon, elle sera affichée et attendue après la valeur numérique.

`unite.penalite`
: La pénalité en cas d'erreur sur l'unité.

`unite.visible`
: Si `TRUE`, l'unité peut être choisie dans un menu déroulant. Sinon, elle doit être entrée dans le champ de la réponse, comme un texte.

`commentaires`
: Le commentaire à affiche selon la réponse proposée. Il doit y avoir autant de valeurs que de bonne réponses (certaines pouvant être `NA`, pour ne pas avoir de commentaire pour la réponse associée).

`couleur.consigne`
: La couleur à utiliser pour afficher le texte de la consigne générée, lorsque l'on précise l'arrondi. Doit être une chaîne de caractères décrivant une couleur reconnaissable en CSS. Par défaut, utilise la valeur dans l'option `Sm.arrondi_couleur` (qui correspond à une consigne en orange si elle n'est pas modifiée).

`temps`
: Le temps conseillé pour répondre à la question, en minutes.

Details

En l'absence de tolérance, la bonne réponse doit être donnée exactement: n'oubliez pas d'arrondir la valeur au bon nombre de décimales, soit en utilisant l'option `n.decimales` ou `n.significatifs`, soit avant d'appeler la fonction.

Si `n.decimales` est précisé et strictement positif, l'arrondi est fait avec la fonction `round` de R. Dans ce cas, un message est automatiquement ajouté à la fin du texte de la question, précisant le nombre de décimales attendu. Ce texte est sur sa propre ligne (précédé d'une balise HTML `<br />`), en italique (balises HTML `<i>` et `<i/>`) et avec la couleur indiquée par `couleur.consigne`.

Si `n.significatif` est précisé et strictement positif, l'arrondi est fait avec la fonction `signif` de R. Dans ce cas, un message est automatiquement ajouté à la fin du texte de la question, précisant le nombre de chiffres significatifs attendu. Ce texte est sur sa propre ligne (précédé d'une

balise HTML <br />), en italique (balises HTML <i> et <i/>) et avec la couleur indiquée par couleur.consigne.

Lorsque vous attendez une réponse entière, pour ne pas avoir d'arrondi, de tolérance ni de consigne associée, forcez le type de bonne.reponse à être un entier, et pas un double (fonction as.integer de R).

La tolérance τ permet d'accepter toutes les valeurs dans un intervalle $[a, b]$, calculé à partir de la réponse r. Le calcul dépend du type de tolérance demandé: pour le type 1, $a = r - \delta$ et $b = r + \delta$, avec $\delta = \tau\, r$. Pour le type 2, $\delta = \tau$. Pour le type 3, $a = r/(1 + \tau)$ et $b = r(1 + \tau)$.

Value

Aucune valeur n'est renvoyée par cette fonction.

Author(s)

Emmanuel Curis <emmanuel.curis@parisdescartes.fr>

See Also

qcm.moodle, qroc.moodle, question_ouverte.moodle, vrai_faux.moodle

Examples

```
## Not run:
# Commencer un nouveau questionnaire
q <- debuter_xml.moodle( "exemple.numerique" )

# Catégorie : exemples
categorie.moodle( "Exemples SARP/Numérique" )

# Une question Moodle numérique, avec 3 décimales
numerique.moodle( "Donnez la valeur de &pi;", pi, n.decimales = 3 )

# On termine le questionnaire
finir_xml.moodle( )

# Importez le fichier exemple.numerique.xml dans Moodle pour voir le résultat...
print(
"Importez le fichier exemple.numerique.xml dans Moodle pour voir le résultat..." )

## End(Not run)
```

---

ouverte.moodle *Créer une question à réponse ouverte (rédactionnelle)*

Description

Cette fonction permet de créer une question dont la réponse est libre (rédactionnelle).

Usage

```
question_ouverte.moodle( texte,
                         titre = "Question r&eacute;dactionnelle ouverte",
                         editeur = c( 'WYSIWIG', 'WYSIWIG+',
                                      'Texte', 'Chasse fixe', 'Aucun' ),
                         avec.texte = TRUE, n.lignes = 15,
                         n.annexes = 0, n.optionnelles = min( n.annexes, 3 ),
                         types = 'PDF',
                         modele = NULL, informations = NULL,
                       commentaire.global = NA, penalite = NA, note.question = NA,
                         idnum = NA,
                         temps, tags = NULL,
                      fichier.xml = get( "fichier.xml", envir = SARP.Moodle.env ) )
```

Arguments

`texte`	L'énoncé de la question (une chaîne de caractères). Cet énoncé peut contenir n'importe quel caractère (codage en UTF-8) et n'importe quelles balises HTML.
`titre`	L'intitulé de la question, en HTML.
`editeur`	Le type d'éditeur de texte à offrir à l'étudiant pour qu'il puisse rédiger sa réponse. "WYSIWIG+" correspond à un éditeur offrant la possibilité de choisir des fichiers.
`avec.texte`	Si `TRUE`, il est obligatoire de répondre à la question en mettant au moins quelques mots. Sinon, il est possible de répondre en ne remplissant rien.
`n.lignes`	Le nombre de lignes de texte que doit afficher l'éditeur, par défaut.
`n.annexes`	Le nombre maximal de fichiers que l'étudiant peut téléverser pour compléter sa réponse.
`n.optionnelles`	Parmi ces fichiers, combien sont-ils obligatoires?
`types`	Les formats de fichiers autorisés, pour le dépôt. Pour l'instant ignoré, car il semble que cette information ne soit pas exportée dans le fichier XML...
`modele`	Un modèle de réponse, à préafficher dans l'éditeur.
`informations`	Des informations qui seront affichées pour le relecteur, afin de le guider dans la correction par exemple.
`commentaire.global`, `penalite`, `note.question`, `idnum`, `tags`	Options communes à toutes les questions Moodle, voyez `debut_question.moodle`.
`temps`	Le temps conseillé pour répondre à la question, en minutes.
`fichier.xml`	Le fichier correspondant au questionnaire qui contiendra cette question (par défaut, le dernier créé).

Details

Cette fonction est une interface simplifiée pour la fonction `question.moodle`.

Value

Aucune valeur n'est renvoyée par cette fonction.

Author(s)

Emmanuel Curis <emmanuel.curis@parisdescartes.fr>

See Also

numerique.moodle, qcm.moodle, qroc.moodle, vrai_faux.moodle

Examples

```
## Not run:
# Commencer un nouveau questionnaire
q <- debuter_xml.moodle( "exemple.ouverte" )

# Catégorie : exemples
categorie.moodle( "Exemples SARP/Question ouverte" )

# Une question Moodle ouverte
question_ouverte.moodle( "Racontez votre journ&eacute;e." )

# On termine le questionnaire
finir_xml.moodle( )

# Importez le fichier exemple.ouverte.xml dans Moodle pour voir le résultat...
print(
"Importez le fichier exemple.ouverte.xml dans Moodle pour voir le résultat..." )

## End(Not run)
```

---

qcm.moodle *Créer une question à choix (réponse unique ou réponses multiples)*

Description

Cette fonction permet de créer une question de type QCM ou QCU, dans laquelle la ou les bonnes réponses sont à cocher dans une liste de réponses prédéfinie.

Usage

```
qcm.moodle( texte, bonnes.reponses, mauvaises.reponses,
        commentaires = NULL, fractions = list( "Bonnes" = NULL, "Fausses" = NULL ),
            unique = ( length( bonnes.reponses ) == 1 ), melanger = TRUE,
            titre = "QCM...",
            numerotation = c( "none", "abc", "ABCD", "123" ),
            fichier.xml = get( "fichier.xml", envir = SARP.Moodle.env ),
            commentaire.global = NA, penalite = NA, note.question = NA,
            idnum = NA,
            temps, instructions = TRUE, tags = NULL, internat = FALSE )
```

Arguments

`texte`	L'énoncé de la question (une chaîne de caractères). Cet énoncé peut contenir n'importe quel caractère (codage en UTF-8) et n'importe quelles balises HTML.
`bonnes.reponses`	Un vecteur de chaînes de caractères correspondant aux intitulés des bonnes réponses. Il doit y avoir au moins une bonne réponse.
`mauvaises.reponses`	Un vecteur de chaînes de caractères correspondant aux intitulés des mauvaises réponses.
`commentaires`	Un vecteur de chaînes de caractères, correspondant au retour à indiquer pour chacune des réponses, dans l'ordre dans lequel elles sont passées, en commençant par les bonnes réponses. Utilisez `NA` pour laisser un commentaire vide.
`fractions`	Une liste de deux éléments, contenant les pourcentage de note associés à chaque réponse. Ces deux éléments doivent être intitulés `Bonnes` (pour les bonnes réponses) et `Fausses` (pour les mauvaises réponses) ; s'ils sont indiqués, ce doivent être des vecteurs numériques ayant le même nombre d'éléments que la liste des réponses. Par défaut (si l'élément vaut `NULL`), chaque bonne réponse est affectée de la même fraction de note, de sorte que l'on atteigne 100 % en cochant toutes les bonnes réponses. De la même façon, les mauvaises réponses sont affectées d'une fraction négative de sorte que cocher toutes les mauvaises réponses annule le fait de cocher toutes les bonnes réponses, sans donner une note négative. Les fractions doivent être comprises entre 0 (exclus) et 100 pour les bonnes réponses ; elles doivent être négatives pour les mauvaises réponses. Si ce n'est pas le cas, ou si la somme des fractions des bonnes réponse ne vaut pas 100, un avertissement est donné, mais la question est tout de même créée. Toutefois, si toutes les fractions des mauvaises réponses sont positives, elles sont considérées négatives sans avertissement.
`unique`	Si `TRUE`, indique que l'on ne peut cocher qu'une seule bonne réponse.
`melanger`	Si `TRUE`, indique que l'ordre des réponses doit être choisi aléatoirement à chaque affichage de la question. Sinon, l'ordre est celui défini dans le fichier. Actuellement, cela correspond à l'ensemble des bonnes réponses dans l'ordre indiqué, suivi de l'ensemble des mauvaises réponses dans l'ordre indiqué.
`titre`	L'intitulé de la question, en HTML.
`numerotation`	Le type de numérotation à utiliser pour les différentes réponses. Ces numéros sont ajoutés par Moodle lors de l'affichage de la question. Seules les quatre valeurs indiquées sont reconnues par Moodle. Vous pouvez abréger en n'utilisant que la première lettre.
`fichier.xml`	Le fichier correspondant au questionnaire qui contiendra cette question (par défaut, le dernier créé).
`commentaire.global`, `penalite`, `note.question`, `idnum`, `tags`	Options communes à toutes les questions Moodle, voyez `debut_question.moodle`.
`temps`	Le temps conseillé pour répondre à la question, en minutes.
`instructions`	Faut-il afficher, avant les réponses possibles, le texte générique de Moodle rappelant les instructions (typiquement, "Veuillez choisir au moins une réponse")? (Oui si `TRUE`, non si `FALSE`).

`internat` Par défaut, le QCM créé est un QCM Moodle classique (type `multichoice`). Si `TRUE`, génère un QCM avec notation similaire à celle du concours de l'internat en pharmacie (notation en fonction du nombre de discordances entre les cases cochées et ce qu'il aurait fallu cocher ou non). Attention, ce type (`sngmultichoice`) ne fonctionnera que si le plugin moodle approprié est installé.

Details

La question générée avec cette fonction est une question de type `multichoice` (sauf si `internat = TRUE`, elle est alors de type `sngmultichoice`, qui nécessite un plugin additionnel).

Cette fonction est une interface simplifiée pour la fonction `question.moodle`.

Value

Aucune valeur n'est renvoyée par cette fonction.

Author(s)

Emmanuel Curis <emmanuel.curis@parisdescartes.fr>

See Also

`numerique.moodle`, `qroc.moodle`, `question_ouverte.moodle`, `vrai_faux.moodle`

Examples

```
## Not run:
# Commencer un nouveau questionnaire
q <- debuter_xml.moodle( "exemple.QCM" )

# Catégorie : exemples
categorie.moodle( "Exemples SARP/QCM" )

# Le format par défaut, avec deux bonnes réponses et trois mauvaises,
qcm.moodle( "Quels sont les fruits de la liste ci-dessous ?",
            c( "Orange", "Tomate" ), c( "Pomme de terre", "Carotte", "Navet" ) )

# Deux bonnes réponses, deux mauvaises réponses, avec commentaires
#   et fractions inégales
qcm.moodle( "Indiquez les romanciers de la liste ci-dessous.",
           c( "Eugène S<small>UE</small>", "Alexandre D<small>UMAS</small>" ),
            c( "Labiche", "Ronsard" ),
            commentaires = c( "Bravo !", NA,
                 "C'est un auteur de pièces de théâtre", "C'est un poète" ),
            fractions = list( "Bonnes"  = c( 75, 25 ),
                              "Fausses" = c( -50, -100 ) ) )

# On termine le questionnaire
finir_xml.moodle( )

# Importez le fichier exemple.QCM.xml dans Moodle pour voir le résultat...
```

```
print(
"Importez le fichier exemple.QCM.xml dans Moodle pour voir le résultat..." )

## End(Not run)
```

---

qroc.moodle *Créer une question à réponse ouverte, courte*

Description

Cette fonction permet de créer une question dont la réponse est un court texte (« QROC »)

Usage

```
qroc.moodle( texte, reponses, notes = rep( 100, length( reponses ) ),
             commentaires = NULL, casse = TRUE,
             titre = "QROC...",
             fichier.xml = get( "fichier.xml", envir = SARP.Moodle.env ),
             commentaire.global = NA, penalite = NA, note.question = NA,
             idnum = NA,
             temps, tags = NULL )
```

Arguments

texte	L'énoncé de la question (une chaîne de caractères). Cet énoncé peut contenir n'importe quel caractère (codage en UTF-8) et n'importe quelles balises HTML.
reponses	Un vecteur de chaînes de caractères, dont chaque élément correspond à une réponse correcte possible.
notes	Un vecteur de nombres, correspondant aux pourcentages de la note totale associés à chaque réponse. Par défaut, toutes les réponses sont supposées parfaitement correctes.
commentaires	Un vecteur de chaînes de caractères, correspondant au retour à afficher pour chacune des réponses entrées (dans le même ordre que les réponses). Le vecteur doit avoir la même longueur ; si certaines réponses n'appellent pas de commentaire, utilisez NA.
casse	Si TRUE, on distingue entre majuscules et minuscules pour savoir si la réponse est correcte. Sinon, cette distinction n'est pas faite.
titre	L'intitulé de la question, en HTML.
fichier.xml	Le fichier correspondant au questionnaire qui contiendra cette question (par défaut, le dernier créé).
commentaire.global, penalite, note.question, idnum, tags	Options communes à toutes les questions Moodle, voyez debut_question.moodle.
temps	Le temps conseillé pour répondre à la question, en minutes.

Details

Cette fonction est une interface simplifiée, et enrichie, pour la fonction question.moodle.

Value

Aucune valeur n'est renvoyée par cette fonction.

Author(s)

Emmanuel Curis <emmanuel.curis@parisdescartes.fr>

See Also

numerique.moodle, qcm.moodle, question_ouverte.moodle, vrai_faux.moodle

Examples

```
## Not run:
# Commencer un nouveau questionnaire
q <- debuter_xml.moodle( "exemple.qroc" )

# Catégorie : exemples
categorie.moodle( "Exemples SARP/QROC" )

# Une question Moodle en QROC, avec une seule bonne réponse
qroc.moodle( "Quel est le prénom de Victor H<small>UGO</small> ?",
             "Victor" )

# La même, en indiquant que l'initiale doit être en majuscule
qroc.moodle( "Quel est le prénom de Victor H<small>UGO</small> ?",
             c( "Victor", "victor" ),
             notes = c( 100, 50 ),
          commentaires = c( NA, "N'oubliez pas la majuscule aux noms propres !" ) )

# On termine le questionnaire
finir_xml.moodle( )

# Importez le fichier exemple.numerique.xml dans Moodle pour voir le résultat...
print(
"Importez le fichier exemple.numerique.xml dans Moodle pour voir le résultat..." )

## End(Not run)
```

---

question.moodle *Créer une question Moodle au format XML*

Description

Ces fonctions permettent de créer une question pour Moodle au format XML.

Usage

```
question.moodle( type = "cloze",
                 titre = "Question...", texte, reponses = NULL,
                 penalite = NA, note.question = NA, commentaire.global = NA,
                 idnum = NA,
                 autres.codes = NULL, tags = NULL,
                 fichier.xml = get( "fichier.xml",
                                    envir = SARP.Moodle.env ) )

debut_question.moodle( type,
                       titre, texte,
                       penalite = NA, note.question = NA,
                       commentaire.global = NA, idnum = NA,
                       fichier.xml = get( "fichier.xml",
                                          envir = SARP.Moodle.env ) )

fin_question.moodle( fichier.xml = get( "fichier.xml",
                                        envir = SARP.Moodle.env ),
                     tags = NULL )
```

Arguments

texte	Le texte de la question (une chaîne de caractères). Cet énoncé peut contenir n'importe quel caractère (codage en UTF-8) et n'importe quelles balises HTML. Il correspond à l'élément entre les balises <questiontext> et </questiontext> de la question. C'est le seul argument obligatoire.
type	Le type de question à créer (élément type de la balise <question>). Vecteur contenant une seule chaîne de caractères.
titre	L'intitulé de la question, en HTML. Il correspond à l'élément entre les balises <name> et </name> de la question. Indiquez NULL pour ne pas donner de titre.
reponses	Un vecteur. Chaque élément correspond à une réponse possible, c'est-à-dire à un champ délimité par les balises <answer> et </answer>. *Attention*, les valeurs de de ce vecteur sont indiquées telles qu'elles dans le fichier XML, sans être protégées : n'utilisez des formats particuliers qu'avec précaution, ou en les protégeant manuellement par une balise <![CDATA[...]]> — ce que font les fonctions de création de question prédéfinies, si nécessaire. Pour indiquer des commentaires (champs <feedback>), créez un vecteur de chaînes de caractères de même longueur que celui des réponses et associez-le aux réponses comme attribut, de nom commentaire. Utilisez NA pour laisser un commentaire vide. Par défaut, la première réponse est considérée comme parfaitement correcte (100% de la note) et toutes les autres comme fausses (0%). Pour indiquer des fractions pour chaque note, créez un vecteur numérique de même longueur que le nombre de réponses et associez-le aux réponses comme attribut, de nom fractions.
commentaire.global	Un vecteur de chaînes de caractère, de longueur 1, correspondant au retour à indiquer globalement (champ délimité par les balises <generalfeedback> et

	</generalfeedback>). Utilisez NA ou NULL pour ne pas indiquer de commentaire.
note.question	La note associée à cette question (champ délimité par les balises <defaultgrade> et </defaultgrade>). Doit être un vecteur numérique avec un seul élément. Utilisez NA pour ne pas indiquer de note. Cette note sera celle attribuée à la question lorsqu'elle sera intégrée à un test.
penalite	La pénalité à utiliser en cas de tentatives multiples, si le premier essai est incorrect (champ délimité par les balises <penalty> et </penalty>). Doit être un vecteur numérique avec un seul élément. Utilisez NA pour ne pas indiquer de pénalité.
idnum	Un entier qui identifie de façon unique la question. L'unicité s'entend au sein de la catégorie à laquelle appartient la question. SARP.moodle vérifie l'unicité pour la catégorie en cours, mais ne peut évidemment pas vérifier qu'il n'y a pas déjà dans la base des questions avec cet identifiant... Attention, à chaque appel à categorie.moodle, la liste des identifiants est remise à zéro.
tags	Un vecteur de chaînes de caractères dont chaque élément est un tag (une étiquette) qui sera associé à la question.
autres.codes	Un vecteur de chaînes de caractères, correspondant aux valeurs à donner à des champs optionnels de la question. Ils sont écrits dans l'ordre. Le nom des champs est déterminé à partir du nom des éléments du vecteur.
fichier.xml	Le fichier correspondant au questionnaire qui contiendra cette question (par défaut, le dernier créé).

Details

Cette fonction est la fonction principale, appelée par toutes les autres fonctions de création d'une question Moodle. Les autres fonctions étant d'usage plus simple, n'utilisez cette fonction que pour des cas non-prévus par les fonctions spécialisées.

Value

Aucune valeur n'est renvoyée par cette fonction.

Author(s)

Emmanuel Curis <emmanuel.curis@parisdescartes.fr>

See Also

numerique.moodle, qcm.moodle qroc.moodle, question_libre.moodle, question_ouverte.moodle, vrai_faux.moodle

Examples

```
## Not run:
# Commencer un nouveau questionnaire
q <- debuter_xml.moodle( "exemple.question" )
```

```
# Catégorie : exemples
categorie.moodle( "Exemples SARP/Question" )

# Une question vrai-faux (en utilisant le type binaire)
reponses <- c( "Vrai", "Faux" )
attr( reponses, "commentaire" ) <- c( "Bravo !", "Elle n'est pas plate" )
attr( reponses, "fractions" ) <- c( 100, 0 )
question.moodle( type  = "truefalse", titre = "Question binaire",
                 texte = "La Terre est ronde", reponses = reponses )

# On termine le questionnaire
finir_xml.moodle( )

# Importez le fichier exemple.question.xml dans Moodle pour voir le résultat...
print(
"Importez le fichier exemple.question.xml dans Moodle pour voir le résultat..." )

## End(Not run)
```

SARP.moodle *Création de questions Moodle au format XML avec R*

Description

Cette bibliothèque permet de générer un fichier de questions pour Moodle, au format XML, sans avoir à connaître ce format. Les questions peuvent être construites à partir d'un fichier CSV ou directement à partir de fonctions R.

Details

Cette bibliothèque permet de faciliter la création de questions pour les tests Moodle : elle génère un fichier XML qui peut être importé dans la base de questions d'un cours Moodle. Il est ensuite possible de piocher dans ces questions pour créer un test. Afin d'éviter toute confusion avec d'autres fonctions, toutes les fonctions de cette bibliothèque se terminent par le suffixe `.moodle`.

L'utilisation la plus simple est la conversion d'un fichier CSV (ou tout autre format texte lisible par `read.table`) avec une ligne par question au format XML (`csv.moodle`). Mais il est aussi possible de créer les questions directement dans R, en créant le fichier XML (`debuter_xml.moodle`) puis en appelant les fonctions appropriées. Dans ce cas, il ne faut pas oublier de clore le fichier avant de l'importer (`finir_xml.moodle`).

La plupart des types de questions de base de Moodle sont connus : vrai-faux (`vrai_faux.moodle`), questions à choix parmi des réponses prédéfinies (`qcm.moodle`), questions à réponse numérique (`numerique.moodle`), questions à réponse libre (`qroc.moodle`, `question_ouverte.moodle`) et le format libre permettant de poser plusieurs questions en une (`question_libre.moodle`). Il est possible de générer un type de question non explicitement supporté grâce à la fonction générique `question.moodle`, à condition que vous connaissiez les noms des balises XML nécessaires.

Il est possible de trier les questions en catégories afin de les repérer plus facilement dans la base de question (`categorie.moodle` ; c'est indispensable si vous prévoyez dans votre test le tirage au sort d'une question parmi une série de questions similaires)

Afin de faciliter la construction des énoncés des questions, quelques fonctions sont proposées pour convertir des objets fréquemment utilisés en code HTML tout en respectant les conventions typographiques françaises : nombres (afficher_nombre.moodle), échantillons de valeurs (vecteur de nombres, afficher_echantillon.moodle, construisant un tableau).

Author(s)

Emmanuel Curis <emmanuel.curis@parisdescartes.fr>

`sortie_R.moodle` *Insérer une sortie de R dans une question*

Description

Ces fonctions permettent de créer du code HTML à partir du résultat d'une fonction R, avec une mise en forme spécifique, pour insertion dans une question Moodle.

Usage

```
sortie_R.moodle( objet.R, ... )

## S3 method for class 'htest'
sortie_R.moodle( objet.R, precision = 4, ... )
## S3 method for class 'call'
sortie_R.moodle( objet.R, precision = 4, ... )
## S3 method for class 'numeric'
sortie_R.moodle( objet.R, precision = 4,
            enjoliver = !cadre, cadre = TRUE,
            couleur.trait = "Black", lg.trait = "2",
            marge = if ( enjoliver ) c( 10, 10, 1, 1 ) else c( 25, 25, 1, 1 ),
            pre.X = NA, pre.X.nom = NA, noms.gras = TRUE, ... )
```

Arguments

`objet.R`	L'objet R à convertir
`precision`	Nombre de décimales à conserver pour l'affichage des nombres (arrondis avec `round`).
`enjoliver`	Si `TRUE`, la sortie est mise en forme en utilisant des balises HTML appropriées. Sinon, elle est faite pour reproduire le plus possible l'affichage dans la console R.
`cadre`	Si `TRUE`, un cadre est dessiné (via les styles CSS appropriées) tout autour de la sortie.

Options pour l'affichage des vecteurs numériques

`couleur.trait`, `lg.trait`
: Couleur et épaisseur des filets des tableaux, pour la mise en forme enjolivée. La couleur peut être n'importe quelle couleur autorisée en CSS ; l'épaisseur des filets est exprimée en pixels.

`marge` Les marges à laisser au sein de chaque case du tableau affichant les résultats. Ce doit être un vecteur de quatre valeurs entières, correspondant dans l'ordre aux marges gauche, droite, haute et basse, exprimées en pixels.

`pre.X`, `pre.X.nom` Dans la version enjolivée, textes à afficher en première colonne du tableau, avant les premières valeurs.

`noms.gras` Si `TRUE`, les noms des éléments du vecteur sont affichés en gras dans la première ligne du tableau.

`...` Paramètres de mise en forme additionnels, en particulier indiquant la commande à afficher (`commande`), sa couleur (`couleur.commande`, nom de couleur HTML/CSS et non R, par défaut du bleu), s'il faut un cadre (`cadre`, présent par défaut) et dans ce cas son épaisseur (`largeur.cadre`, 2 pixels par défaut) et sa couleur (`couleur.cadre`, noir par défaut).

Value

Une chaîne de caractère contenant le code HTML approprié.

Author(s)

Emmanuel Curis <emmanuel.curis@parisdescartes.fr>

See Also

`afficher_echantillon.moodle` pour afficher un vecteur de valeurs sous forme d'un tableau.

---

vf.moodle *Créer une question à deux réponses possibles, dont une seule vraie*

Description

Cette fonction permet de créer une question de type vrai ou faux.

Usage

```
vrai_faux.moodle( texte,  texte.vrai = "Vrai", texte.faux = "Faux",
                  commentaires = NULL,
                  titre = "Question vrai-faux...",
            ordre = c( "aleatoire", NA, "random", "vrai premier", "faux premier" ),
                  melanger = FALSE,
                  fichier.xml = get( "fichier.xml", envir = SARP.Moodle.env ),
                  commentaire.global = NA, penalite = NA,
                  note.question = NA, idnum = NA, temps, tags = NULL,
                  multichoice = TRUE )
```

Arguments

texte	L'énoncé de la question (une chaîne de caractères). Cet énoncé peut contenir n'importe quel caractère (codage en UTF-8) et n'importe quelles balises HTML.
texte.vrai	Le texte à utiliser pour la bonne réponse.
texte.faux	Le texte à utiliser pour la mauvaise réponse.
commentaires	Un vecteur de deux chaînes de caractère, correspondant au retour à indiquer pour la bonne réponse (premier élément) et pour la mauvaise réponse (second élément). Utilisez NA pour laisser un commentaire vide.
titre	L'intitulé de la question, en HTML.
ordre	L'ordre dans lequel les deux réponses possibles doivent apparaître. Les trois premières possibilités décident d'un ordre aléatoire (fixe). Cette option n'a d'intérêt que si melanger=FALSE...
melanger	Si TRUE, indique que Moodle devra tirer au sort l'ordre d'affichage des réponses (qui changera donc à chaque fois que la question sera posée). Sinon, l'ordre sera toujours celui indiqué lors de la création de la question.
fichier.xml	Le fichier correspondant au questionnaire qui contiendra cette question (par défaut, le dernier créé).
commentaire.global, penalite, note.question, idnum, tags	Options communes à toutes les questions Moodle, voyez debut_question.moodle.
temps	Le temps conseillé pour répondre à la question, en minutes.
multichoice	Faut-il créer la question comme un QCU (TRUE, valeur par défaut) ou comme une vraie question « vrai-faux » (FALSE) ?

Details

Bien qu'un type particulier de question existe dans Moodle pour les questions « vrai ou faux » (type truefalse), la question générée avec cette fonction est par défaut une question de type multichoice, afin d'avoir la liberté des intitulés des bonnes et mauvaises réponses. Il est possible de forcer la création d'une vraie question « vrai-faux » (type truefalse) en utilisant multichoice=FALSE. Attention, les réponses ne peuvent alors contenir que vrai ou faux, aucun autre texte.

Cette fonction est une interface simplifiée pour la fonction question.moodle.

Value

Aucune valeur n'est renvoyée par cette fonction.

Author(s)

Emmanuel Curis <emmanuel.curis@parisdescartes.fr>

See Also

numerique.moodle, qcm.moodle, qroc.moodle, question_ouverte.moodle.

Examples

```
## Not run:
# Commencer un nouveau questionnaire
q <- debuter_xml.moodle( "exemple.VF" )

# Catégorie : exemples
categorie.moodle( "Exemples SARP/Vrai ou Faux" )

# Le format par défaut
vrai_faux.moodle( "Une question « vrai ou faux » offre le choix entre deux réponses" )

# et si la bonne réponse est « faux »
vrai_faux.moodle( "Le Soleil tourne autour de la Terre", "Faux", "Vrai" )

# avec des intitulés libres
vrai_faux.moodle( "La bataille de Marignan a eu lieu en...", "1515", "1914" )

# On termine le questionnaire
finir_xml.moodle( )

# Importez le fichier exemple.VF.xml dans Moodle pour voir le résultat...
print(
"Importez le fichier exemple.VF.xml dans Moodle pour voir le résultat..." )

## End(Not run)
```

Index

∗ **package**
SARP.moodle, 58

affichage.moodle, 2
afficher_echantillon.moodle, *59, 60*
afficher_echantillon.moodle *(*affichage.moodle*)*, 2
afficher_echantillons.moodle *(*affichage.moodle*)*, 2
afficher_nombre.moodle, *59*
afficher_nombre.moodle *(*affichage.moodle*)*, 2
afficher_poly.moodle *(*affichage.moodle*)*, 2
annoter_image.moodle, 5
arrondi.moodle *(*messages.moodle*)*, 46
as.integer, *49*

cat, *20*
categorie.moodle, 7, *13*, *27*, *57, 58*
coder_image.moodle *(*definir_dossier.image.moodle*)*, 20
conversion_csv.moodle *(*csv.moodle*)*, 8
creer_glossaire.moodle, *15*
creer_glossaire.moodle *(*glossaire.moodle*)*, 33
csv.moodle, 8, *18*, *58*
csv_glossaire.moodle, 15
csv_optique *(*csv_optique.moodle*)*, 16
csv_optique.moodle, *14*, 16

debut_question.moodle, *6*, *24*, *30*, *33*, *40*, *42*, *48*, *50*, *52*, *54*, *61*
debut_question.moodle *(*question.moodle*)*, 55
debuter_xml.moodle, *11*, *17*, 19, *26*, *58*
definir_dossier.image.moodle, *12*, *17*, 20, *27*, *45*
description.moodle, *13*, 23
df.moodle, 24

entree_glossaire.moodle *(*glossaire.moodle*)*, 33

fin_question.moodle *(*question.moodle*)*, 55
finir_xml.moodle, *58*
finir_xml.moodle *(*debuter_xml.moodle*)*, 19

glisser_deposer *(*glisser_deposer.moodle*)*, 29
glisser_deposer.moodle, 29, *33*, *41*
glisser_texte *(*glisser_textes.moodle*)*, 32
glisser_textes *(*glisser_textes.moodle*)*, 32
glisser_textes.moodle, *13*, *31*, 32, *41*
glossaire.moodle, 33
grconvertX, *31*, *40*
grconvertY, *31*, *40*

inserer_formule.moodle, *14*, *18*, *23*, *28*, 35, *38*, *46*
inserer_image.moodle *(*definir_dossier.image.moodle*)*, 20
inserer_SMILES.moodle, *14*, *18*, *23*, *28*, *36*, 37, *46*

legender_image *(*legender_image.moodle*)*, 39
legender_image.moodle, *31*, *33*, 39
libre.moodle, 41
lier_fichier.moodle, 44
lier_image.moodle, *36*, *38*, *46*
lier_image.moodle *(*definir_dossier.image.moodle*)*, 20

messages.moodle, 46

numerique.moodle, *12*, *27*, *43*, 47, *51*, *53*, *55*, *57, 58*, *61*

ods.moodle *(*csv.moodle*)*, 8
options, *20*
ouverte.moodle, 49

plot, *41*

qcm *(*qcm.moodle*)*, 51
qcm.moodle, *43*, *49*, *51*, 51, *55*, *57, 58*, *61*
qroc *(*qroc.moodle*)*, 54
qroc.moodle, *12*, *27*, *43*, *49*, *51*, *53*, 54, *57, 58*, *61*
question.moodle, *43*, *50*, *53*, *55*, 55, *58*, *61*
question_libre.moodle, *13*, *28*, *57, 58*
question_libre.moodle *(*libre.moodle*)*, 41
question_ouverte.moodle, *13*, *43*, *49*, *53*, *55*, *57, 58*, *61*
question_ouverte.moodle *(*ouverte.moodle*)*, 49

read.table, *12*, *15*, *18*, *58*
round, *48*

SARP.moodle, 58
SARP.moodle-package *(*SARP.moodle*)*, 58
signif, *48*
sortie_R.moodle, 59
sortie_R.moodle.numeric, *3*

temps_necessaire.moodle *(*messages.moodle*)*, 46

vf.moodle, 60
vrai_faux.moodle, *12*, *27*, *43*, *49*, *51*, *53*, *55*, *57, 58*
vrai_faux.moodle *(*vf.moodle*)*, 60

xlsx.moodle *(*csv.moodle*)*, 8