Fonctions d'évaluations

Fonctions

Convertir en entier (cast)

Double cliquez sur la fonction voulue donc “**Convertir en entier**”, cela fait apparaitre la fonction dans la zone texte au dessus.

La fonction **long()** prend en paramètre une variable qu'elle transforme en entier.

Dans la liste de champs en-dessous, il suffit donc de choisir le champs qui contient la valeur que vous souhaitez convertir. Faites un double clic.

Vous verrez alors apparaitre la pseudo-variable, correspondant au champs, dans la zone texte, en argument de la fonction. A moins d'une erreur, il n'y a plus qu'à cliquer sur Valider pour que votre choix soit prêt à être appliqué. Il est indiqué dans le formulaire, section “**Options de capture**”

Condition OU entre plusieurs champs

@or renvoi 1 (sous forme de chaîne de caractère) si les champs champ1 OU champ2 ont été retrouvés sur le document. Dans le cas contraire la chaîne vide sera renvoyée.

```
@or(champ1,champ2)
```

Où champ1 et champ2 sont des références de champs (@FLDX)

Condition évaluée

```
@condition(<expression>)
```

Où *expression* doit être une expression python valide. Elle peut contenir des références à des champs d'analyse COLD (@FLDX). **@condition** évalue l'expression python fournie en paramètre. Elle retourne 1 si l'expression est évaluée à vraie, et 0 si l'expression est évaluée à faux.

Avant évaluation les références @FLDX sont remplacées par leurs valeurs et sont encadrées par des simples quotes (sauf pour les valeurs 0, 1, True, False). Attention donc car un nombre devient une chaîne de caractère ! Il peut être alors nécessaire d'utiliser des fonctions de conversion (long(), int(), float())

Conversion du jour numérique en texte

@int2day est une macro qui traduit un jour exprimé sous forme numérique dans sa représentation

littérale.

```
@int2day(nombre)
```

Où nombre peut prendre les valeurs suivantes: * Une chaîne de caractères encadrée par des quotes simples.

- Un numérique.
- @self : une référence au champ lui-même.
- @<groupname> : une référence à une capture dans le filtre de champ.

```
@int2day(3) => mercredi
@int2day("7") => dimanche
```

En utilisant @self, si celui-ci vaut 5:

```
@int2day(@self) => vendredi
```

En utilisant une variable de capture <*jour*> qui vaut 1

```
@int2day(@<jour>) => lundi
```

Conversion du mois texte en mois numérique

```
@monthint(mois)
```

Où le paramètre *mois* est une chaîne de caractères qui représente l'écriture d'un mois sous sa forme littérale.

@monthint retourne le numéro (sur deux chiffres) du mois correspondant.

Ci dessous le tableau des correspondances entre un numéro et une représentation littérale du mois.

Numéro de mois en sortie	**Valeurs possibles en entrée**
01	ja, jan, janv, janu, janvi, janua, janvie, janvier, januar, january
02	f, fe, fev, feb, fevr, febr, fevri, febru, fevrie, februa, fevrier, februar, february
03	mar, mars, marc, march, marz
04	av, ap, avr, apr, avri, apri, avril, april
05	mai, may
06	juin, jun, june, juni
07	jul, july, juil, juill, juille, juillet
08	au, ao, aug, aou, aout, augu, augus, august
09	s, se, sept, septe, septem, septemb, septembe, septembr, septembre, september
10	o, oc, oct, octo, octob, octobe, octobr, october, octobre, okt, oktober
11	n, no, nov, nove, novem, novemb, novembr, novembe, novembre, november

Numéro de mois en sortie	Valeurs possibles en entrée
12	d, de, dec, dece, decem, decemb, decembr, decembe, decembre, december, dez, dezember

Conversion du mois numérique en texte

@int2month est une macro qui traduit un mois exprimé sous forme numérique dans sa représentation littérale.

Extraction d'une partie de la chaine

Cette macro permet de ne prendre qu'une partie d'une valeur.

Il faut savoir qu'une chaine en python est en réalité un tableau de caractères affichés les uns à la suite des autres.

C'est à dire : “glou glou”
est en fait un tableau qui contient : [“g”,“l”,“o”,“u”,“g”,“l”,“o”,“u”]

Donc utiliser la macro ainsi :

@champ[0] affichera “g” uniquement

Ou encore :

@champ[1:3] affichera “lou”

Rechercher la position d'une chaine xxx depuis le debut

@champ.find('z')

La macro parcoure la chaine en affectant un numéro à chaque caractère (espaces compris), depuis 0. Dès qu'elle trouve une première fois la valeur xxx, elle arrête la détection et renvoie la position du premier caractère.

Si vous souhaitez obtenir la position du dernier caractère, vous pouvez utiliser la macro inverse rfind.

Exemple :

si le champ contient “le cheval s'appelle Miguel” et que vous cherchez le premier a, utilisez @champ.find('a') Vous obtiendrez 7

La macro compte les espaces comme des caractères.

From:
http://wiki.ezdev.fr/ - **EzGED Wiki**

Permanent link:
http://wiki.ezdev.fr/doku.php?id=doc:v3:acquisition:apprentissage:fonctionsevaluation&rev=1514540844

Last update: **2023/03/17 09:56**