Syntaxe

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

Où *expression* doit être une expression python valide. Elle peut contenir des références à des champs d'analyse COLD (@FLDX).

Description

@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())

Exemple

Nous avons un champ d'analyse “Montant TTC” dont l'identifiant est 32 et la valeur trouvée est 88.12 . Nous souhaitons que le champ d'analyse “Montant sup. 100” (dont le champ de destination est un booléen) prenne la valeur 1 si le montant ttc est supérieur à 100. Inversement il prendra la valeur 0 si le montant ttc est inférieur à 100.

Voici ce que nous écrirons dans “Fonction avancée sur le champ”:

```
@condition(float(@FLD32)>100)
```

Notez que la valeur de @FLD32 est explicitement transformée en float. Si nous ne l'avions pas fait l'expression serait toujours évaluée à vraie et retournerait toujours 1.

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

Permanent link:
http://wiki.ezdev.fr/doku.php?id=cold:reference:coldfldextra:condition

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