



## Utiliser les données - Ecrire l'expression d'une variable et d'une data

Nous montrons dans cette section comment écrire l'expression d'une variable et d'une data.

### 1. Ecrire l'expression d'une variable

L'expression d'une variable peut être un entier, une chaîne de [caractère](#), une fonction, une autre variable ou encore la concaténation de plusieurs variables. Nous illustrons ci-après tous ces cas de figures.

#### 1.1. I- L'expression de la variable est un entier

Si la valeur d'une variable est un entier, il suffit de la saisir dans le champ d'attribut "Expression" de la variable.

The screenshot displays a software interface with two panes. The left pane, titled 'Exemple variable et data', shows a hierarchical tree structure with nodes for 'Title', 'Content', and 'Paragraph'. A variable definition 'variableInteger = 123' is highlighted in a red box. The right pane, titled 'Project Explorer' and 'ToolBox', shows the 'Properties' window for the variable. The 'Expression' field is highlighted in a red box and contains the value '123'.

Properties	
<b>Variable</b>	
<input type="checkbox"/> Conditions	
IsEnabled	
<input type="checkbox"/> Definition	
DataType	integer
<input type="checkbox"/> Expression	
Expression	123
<input type="checkbox"/> Identification	
Name	variableInteger
<input type="checkbox"/> Validation	
Submit	False

## 1.2. II- L'expression de la variable est une chaîne de caractères

Si l'expression d'une variable est une chaîne de **caractère**, il suffit de l'écrire entre deux guillemets simples (").

**Exemple 1: Une simple chaîne de caractère.**

Variable	
Conditions	IsEnabled
Definition	DataType: string
Expression	Expression: 'Je suis une chaîne de caractères exprimée dans une variable'
Identification	Name: variablesString
Validation	Submit: False

**Exemple 2: La chaîne de caractère contient un apostrophe**

Cette image montre comment saisir l'expression d'une variable contenant l'**apostrophe**. Avant l'apostrophe, mettre un anti-slash.

Variable	
Conditions	IsEnabled
Definition	DataType: string
Expression	Expression: '\apprentissage'
Identification	Name: MyVariable
Validation	Submit: False

**Exemple 3: La Chaîne de caractères contient des chiffres**

Il suffit d'écrire la chaîne de caractères contenant des chiffres entre deux guillemets simples (").

Variable	
Conditions	IsEnabled
Definition	DataType: string
Expression	Expression: '07/10/2009'
Identification	Name: todayDate
Validation	Submit: False

### 1.3. III- L'expression de la variable est une fonction

L'expression d'une variable peut être une fonction avec ou sans argument

#### Exemple 1: La fonction a un argument de type string

Properties	
Variable	
Conditions	
IsEnabled	
Definition	
DataType	string
Expression	<code>getCurrentDate('dd/MM/yyyy HH:mm')</code>
Identification	
Name	todayDate2
Validation	
Submit	False

#### Exemple 2: La fonction a deux arguments (une variable et un String)

Properties	
Variable	
Conditions	
IsEnabled	
Definition	
DataType	string
Expression	<code>formatDate(\$(date_en_entier), 'dd/MM/yyyy')</code>
Identification	
Name	date_en_string
Validation	
Submit	False

#### Exemple 3: La fonction a plusieurs arguments (trois variables)

Properties	
Variable	
Conditions	
IsEnabled	
Definition	
DataType	string
Expression	<code>replace(\$(replace_txt1), \$(replace_txt2), \$(replace_txt3))</code>
Identification	
Name	replace_v
Validation	
Submit	False

### 1.4. IV- L'expression est la concaténation de plusieurs variables

L'expression d'une variable peut être une autre variable ou la concaténation de plusieurs variables. Pour l'exprimer, il suffit juste d'écrire le nom de(s) variable(s).

#### Exemple 1: Concaténation de deux variables de type integer

Dans cet exemple, la variable \$(A3) est égale à la valeur de \$(A1) plus celle de \$(A2).

\$(A1)= 10 , \$(A2)= 15 Le calcul donne se résultat: \$(A3)= .

Properties

Variable	
Conditions	IsEnabled
Definition	DataType <b>integer</b>
Expression	Expression <b>\$(A1) + \$(A2)</b>
Identification	Name <b>A3</b>
Validation	Submit <b>False</b>

**Exemple 2: Concaténation de deux variables de type string (avec espace entre elles)**

Properties

Variable	
Conditions	IsEnabled
Definition	DataType <b>string</b>
Expression	Expression <b>\$(A) + ' ' + \$(B)</b>
Identification	Name <b>C</b>
Validation	Submit <b>False</b>

**Exemple 3: Concaténation de trois variables de type string (avec un caractère entre elles)**

Properties

Variable	
Conditions	IsEnabled
Definition	DataType <b>string</b>
Expression	Expression <b>\$(varDay) + '/' + \$(varMonths) + '/' + \$(varYear)</b>
Identification	Name <b>varDate</b>
Validation	Submit <b>False</b>

## 2. Ecrire l'expression d'une data

L'expression d'une data peut être une variable, la concaténation de plusieurs variables ou encore une fonction. Nous illustrons ci-après tous ces cas de figures.

### 2.1. I- L'expression de la data est une variable

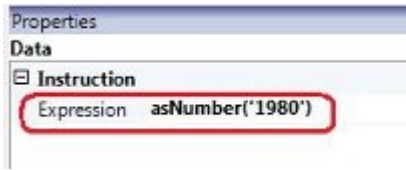
Dans cet exemple, la variable "\$(*extractsD\_v*)" est égale à "*extractsDigits(extractsD\_txt)*". La fonction *extractsDigits* renvoie les chiffres que contient un texte. Les chiffres que contient la variable "\$(*extractsD\_txt*)" sont retournés dans la variable "\$(*extractsD\_v*)". Ces derniers sont affichés par usage d'une data ayant comme expression la variable "\$(*extractsD\_v*)".



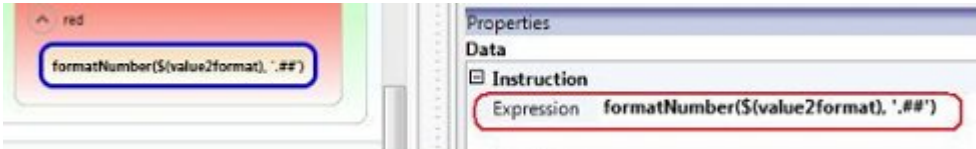
## 2.2. II- L'expression de la data est une fonction

Une fonction (avec ou sans argument) peut être utilisée pour exprimer la valeur d'une data.

### Exemple 1: La fonction a un argument string



### Exemple 2: La fonction a deux arguments (une variable et un string)



### Exemple 3: La fonction a plusieurs arguments (deux variables)

