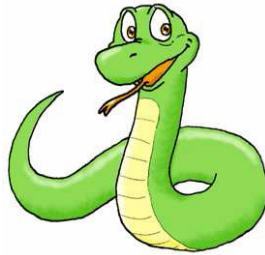


ECT1



TP 02: Notion de Variables

Exercice 1 : Première affectation

1- Évaluez dans la console `a` . Notez le message d'erreur obtenu.

2- Évaluez dans la console `a=2` . Puis de nouveau taper `a`. Comparer et expliquer ce résultat.

3- Évaluez dans la console `A` . Que pouvez-vous en conclure.

4- Évaluez dans la console `a==3` . Que pouvez-vous en conclure.

5- Évaluez dans la console `a=a-1` . Que contient la variable `a` ?

6- Évaluez dans la console `x=y=7` . Que contient la variable `x` ? Que contient la variable `y`?

7- Évaluez dans la console `b, c=1, 2` . Que contient la variable `b`? Que contient la variable `c`?



Bilan 1: Variable

Une variable est un emplacement de la mémoire ("boite") dans lequel on peut stocker un contenu .

Le symbole `=` n'est pas celui de l'égalité mathématique, il n'est d'ailleurs pas symétrique.

Il s'agit d'affecter une valeur à une variable : on stocke une valeur numérique ou du texte dans une mémoire.

La syntaxe est `NomDeLaVariable = valeur`.

Il est possible d'avoir une affectation simultanée

Les noms de variables sont des noms (identificateurs) qu'on choisit assez librement. De préférence assez courts, mais **aussi explicites que possible**, pour exprimer clairement ce que la variable est censée contenir : altitude, altit ou alt (au lieu de `x`) pour exprimer une altitude

`prix_unit` pour exprimer un prix unitaire, etc.

Quelques règles pour les noms de variables sous Python :

1. Un nom de variable (identificateur) est une séquence de lettres (a à z , A à Z) et de chiffres (0 à 9), qui doit toujours commencer par une lettre.

2. Seules les lettres ordinaires sont autorisées. Les lettres accentuées, les cédilles, les espaces, les caractères spéciaux tels que \$, #, @, etc. sont interdits, à l'exception du caractère `_` (souligné).

3. La casse est significative (les caractères majuscules et minuscules sont distingués). Attention : Somme, somme, SOMME sont donc des variables différentes. Soyez attentifs !

4. En plus de ces règles, il faut encore ajouter que vous ne pouvez pas utiliser comme noms de variables les 33 « mots réservés » ci-dessous (ils sont utilisés par le langage lui-même) :

<code>and</code>	<code>del</code>	<code>from</code>	<code>None</code>	<code>True</code>	<code>as</code>	<code>elif</code>	<code>global</code>	<code>nonlocal</code>	<code>try</code>
<code>assert</code>	<code>else</code>	<code>if</code>	<code>not</code>	<code>while</code>	<code>break</code>	<code>except</code>	<code>import</code>	<code>or</code>	<code>with</code>
<code>class</code>	<code>False</code>	<code>in</code>	<code>pass</code>	<code>yield</code>	<code>continue</code>	<code>finally</code>	<code>is</code>	<code>raise</code>	<code>def</code>
<code>for</code>	<code>lambda</code>		<code>return</code>						

Exercice 2 : Typage des variables

1- Évaluez dans la console

```
n=7
x=6/2
msg="Quoi de neuf ?"
n==8
```

Vous remarquerez que l'affectation n'est pas que numérique. Les variables sont de plusieurs types. Pour connaître le type de la variable on a l'instruction `type()`

2- Évaluez dans la console `type(n)`

3- Donner le type des différentes lignes



Bilan 2: Types de données

Les types de données les plus simples, ou primitifs, utilisés dans Python sont :

- ☐ Les entiers (Integers en anglais)
- ☐ Les virgules flottantes (Float en anglais): les nombres décimaux
- ☐ Les chaînes de caractères (Strings en anglais) : texte entre guillemets
- ☐ Les booléens: vrai ou faux

Nous verrons qu'il existe aussi un type matriciel (tableau)

Le résultat d'une opération est toujours de type float sauf pour les opérations de divisions euclidienne (`//` et `%`)

4- Donner le type des expressions suivantes sans utiliser Python

<i>a</i>	<i>type(a)</i>
<i>a=2</i>	
<i>a=2.0</i>	
<i>a=2+3</i>	
<i>a=2+3.0</i>	
<i>a= 'Bonjour'</i>	
<i>a=False</i>	
<i>a=2<3</i>	
<i>a="2<3"</i>	
<i>a=[2,3]</i>	
<i>a= 2.1</i>	

5- Évaluez dans la console `x=7/2;x;type(x)`

6- Évaluez dans la console `int(x),type(x)`

Remarque: Observer la différence entre , et ;

7- Comment rendre entier x



Bilan 3: Modification du type des variables

Il est possible de modifier explicitement le type des variables en utilisant les fonctions de conversion :

- ☐ `int(a)` Conversion de a en type entier
- ☐ `float(a)` Conversion de a en type flottant (décimale)
- ☐ `str(a)` Conversion de a en chaîne de caractère

Exercice 3 : A vous de deviner

Essayer de trouver ce que va afficher Python et que contient la variable x après les instructions suivantes :

Vérifier vos réponses avec la console

1- `x=4`

2- `(x+2)**2`

3- `2x`

4- `2*x`

5- `x=2*x`

--

6- `str(x)`

--

7- `a=5+1;b=2*a;a;b`

--

8- `b>a`

--

9- `a='je';b='travaille';a+b`

--

10- `a=a+b;a`

--

Exercice 4 : ordre des affectations

On considère deux programmes Python

```
1  # programme 1
2  a=2
3  b=-5
4  a,b=a+b,a-b
5  a
6  b
```

```
1  # programme 2
2  a=2
3  b=-5
4  a=a+b
5  b=a-b
6  a
7  b
```

Vous pouvez utiliser un tableau des affectations du type:

Variables	a	b
Affectation 1		
Affectation 2		
....		

Dire quelle sera l'affichage à l'issue de ces 2 programmes? Expliquer la différence des résultats

--

Exercice 5 : A vous de deviner

Dans les différents cas que contiennent les variables x , y et z à la fin des programmes

Vous pouvez utiliser un tableau des affectations du type:

Variables	x	y	z
Affectation 1			
Affectation 2			
....			

```
1  # Cas n°1
2  x=3
3  y=5
4  x=2*x+4
5  y=y-x
6  x
7  y
```

```
1  # Cas n°2
2  x=6
3  y=2
4  x=y
5  y=x
6  x
7  y
```

```
1  # Cas n°3
2  x=1
3  y=4
4  x=x+y
5  y=x-y
6  x=x-y
7  x
8  y
```

```
1  # Cas n°4
2  x,y=3,2
3  x=x-y
4  y,x=x+1,2*y
5  y=y%3
6  x=x-y//2
7  x
8  y
```

```
1  # Cas n°5
2  x,y,z=1,2,3
3  x,y=y,z
4  y,z=z,x
5  y=z
6  z=y
```
