

1.1 Données

La console Python se comporte comme une « grosse » calculatrice dans laquelle on peut effectuer des opérations sur différents **types** de données :

- Les **entiers**, qui correspondent aux entiers relatifs, comme par exemple 74 et -2.
- Les **flottants** qui correspondent aux nombres décimaux dans lesquels devront obligatoirement apparaître leurs parties décimales, comme par exemple 12.97 et -14.0.
- Les **chaînes de caractères** qui sont des successions de caractères encadrés par des guillemets, comme par exemple "Ok!", "456" et "Bonjour Gaëtan, Damien et Marius".



Activité de découverte

Sur la console Python, si on souhaite calculer $5 + 6 \times (4 - 1)$, on saisit puis valide l'expression $5+6*(4-1)$, on obtiendra alors 23 :

```
>>> 5+6*(4-1)
23
```

En effet, $5 + 6 \times (4 - 1) = 5 + 6 \times 3 = 5 + 18 = 23$.

1) De même, si on saisit puis valide les expressions suivantes, que va-t-on obtenir ? Vérifier les résultats sur la console Python.

a) $60/6-1+3*2$ b) $(60/(6-1)+3)*2$

2) Sur la console Python, saisir puis valider chacune des expressions suivantes et donner les résultats obtenus.

a) $23/4$ b) $23//4$ c) $23\%4$

Effectuer d'autres saisies de $n//p$ et $n\%p$ pour différentes valeurs entières de n et p , avec p non nul.

1.1 Données

Si n et p sont des entiers naturels, avec p non nul, que retourne $n//p$ et $n\%p$?

3) On peut faire des opérations sur les chaînes de caractères.

Sur la console Python, saisir puis valider chacune des expressions suivantes et donner les résultats obtenus.

- | | |
|-------------------|----------------------|
| a) "azerty"+"123" | b) "12"+"14"+" "+"1" |
| c) 2*"30" | d) 3*"5"+4*"7" |
| e) len("bonjour") | f) len("az"+"123") |
| g) "bonjour"[0] | h) "bonjour"[2] |
| i) "bonjour"[0:3] | j) "bonjour"[2:4] |

Si ch , $ch1$ et $ch2$ sont des chaînes de caractères et i , j , n des entiers avec $i < j$, que retourne $ch1+ch2$, $n*ch$, $len(ch)$, $ch[i]$ et $ch[i:j]$?

4) Sur la console Python, saisir puis valider chacune des expressions suivantes et donner les résultats obtenus.

- | | |
|---------------|------------------|
| a) str(12) | b) str(7.54) |
| c) int("12") | d) float("7.54") |
| e) int(43.67) | f) int(7/2) |

Quelle sont les fonctions des opérateurs `str`, `int` et `float` ?

5) Dans un programme ou un algorithme on peut être amené à faire des tests. Par exemple, en saisissant puis validant « $2^3==8$ » sur la console Python, qui signifie « 2^3 est bien égal à 8 », elle retourne `True`, c'est-à-dire VRAI.

Réaliser sur la console Python les tests suivants et interpréter les résultats obtenus.

- | | |
|----------------|-------------------|
| a) $5.1==5.01$ | b) $5.1!=5.01$ |
| c) $3<4$ | d) $3<=4$ |
| e) $4>=4$ | f) $3>=4$ |
| g) "az"=="za" | h) $(1+2)/3==4/4$ |

Corrigé

1) a) $60/6-1+3*2$ retourne 15.

En effet : $\frac{60}{6} - 1 + 3 \times 2 = 10 - 1 + 6 = 15$

b) $(60/(6-1)+3)*2$ retourne 30.

En effet : $\left(\frac{60}{6-1} + 3\right) \times 2 = \left(\frac{60}{5} + 3\right) \times 2 = (12 + 3) \times 2 = 30$

Vérifications sur la console Python :

```
>>> 60/6-1+3*2
15.0
>>> (60/(6-1)+3)*2
30.0
```

2) a) $23/4$ retourne 5.75.

b) $23//4$ retourne 5.

c) $23\%4$ retourne 3.

Si n et p sont des entiers naturels avec p non nul :

- $n//p$ retourne le **quotient** dans la division euclidienne de n par p .
- $n\%p$ retourne le **reste** dans la division euclidienne de n par p .

3) a) `"azerty"+"123"` retourne `"azerty123"`.

b) `"12"+"14"+" "+"1"` retourne `"1214 1"`.

c) `2*"30"` retourne `"3030"`.

d) `3*"5"+4*"7"` retourne `"5557777"`.

e) `len("bonjour")` retourne 7.

f) `len("az"+"123")` retourne 5.

g) `"bonjour"[0]` retourne `"b"`.

h) `"bonjour"[2]` retourne `"n"`.

i) `"bonjour"[0:3]` retourne `"bon"`.

j) `"bonjour"[2:4]` retourne `"nj"`.

1.1 Données

Si ch , $ch1$ et $ch2$ sont des chaînes de caractères et i, j, n des entiers avec $i < j$:

- $ch1+ch2$ retourne la **concaténation** des chaînes $ch1$ et $ch2$,
- $n*ch$ retourne la concaténation de n chaînes ch ,
- $len(ch)$ retourne le nombre de caractères de la chaîne ch , que l'on appelle **longueur** de ch ,
- $ch[i]$ retourne dans une chaîne le $i+1^e$ caractère de la chaîne ch ,
- $ch[i:j]$ retourne la chaîne constituée des caractères allant du $i+1^e$ caractère au j^e caractère de la chaîne ch .

4) a) $str(12)$ retourne "12".

b) $str(7.54)$ retourne "7.54".

c) $int("12")$ retourne 12.

d) $float("7.54")$ retourne 7.54.

e) $int(43.67)$ retourne 43.

f) $int(7/2)$ retourne 3.

- L'opérateur `str` transforme un nombre en une chaîne de caractères.
- L'opérateur `int` extrait lorsque c'est possible, un entier d'une chaîne de caractères. De plus, il renvoie la partie entière d'un nombre.
- L'opérateur `float` extrait lorsque c'est possible, un flottant d'une chaîne de caractères.

5) a) $5.1==5.01$ retourne `False`. Interprétation : $5,1 = 5,01$ est faux.

b) $5.1!=5.01$ retourne `True`. Interprétation : $5,1 \neq 5,01$ est vrai.

c) $3<4$ retourne `True`. Interprétation : $3 < 4$ est vrai.

d) $3<=4$ retourne `True`. Interprétation : $3 \leq 4$ est vrai.

e) $4>=4$ retourne `True`. Interprétation : $4 \geq 4$ est vrai.

f) $3>=4$ retourne `False`. Interprétation : $3 \geq 4$ est faux.

g) $"az"=="za"$ retourne `False`. Interprétation : $"az"="za"$ est faux.

h) $(1+2)/3==4/4$ retourne `True`. Interprétation : $\frac{1+2}{3} = \frac{4}{4}$ est vrai.

➤ *A retenir*

Condition	Opération	En Python	Dans un algorithme
a, b, n, p nombres avec n, p entiers.	$a+b$	<code>a+b</code>	$a+b$
	$a-b$	<code>a-b</code>	$a-b$
	$a \times b$	<code>a*b</code>	$a*b$
	$a \div b$	<code>a/b</code>	a/b
	a^n	<code>a**n</code>	a^n
	Quotient de la division euclidienne de n par p	<code>n//p</code>	$n//p$
	Reste de la division euclidienne de n par p	<code>n%p</code>	$n\%p$
	Partie entière de a	<code>int(a)</code>	$\text{int}(a)$
	Transformation du nombre a en une chaîne de caractères	<code>str(a)</code>	$\text{str}(a)$
ch, ch1, ch2 chaînes de caractères. i, j, n entiers.	Concaténation des chaînes ch1 et ch2	<code>ch1+ch2</code>	$ch1+ch2$
	Concaténation de n chaînes ch	<code>n*ch</code>	$n*ch$
	Nombre de caractères de la chaîne ch	<code>len(ch)</code>	$\text{longueur}(ch)$
	Chaîne constituée du $i+1^{\text{e}}$ caractère de la chaîne ch	<code>ch[i]</code>	$ch[i]$
	Chaîne constituée des caractères allant du $i+1^{\text{e}}$ caractère au j^{e} caractère de la chaîne ch	<code>ch[i:j]</code>	$ch[i:j]$
	Extraction d'un entier de ch (si possible)	<code>int(ch)</code>	$\text{int}(ch)$
	Extraction d'un flottant de ch (si possible)	<code>float(ch)</code>	$\text{float}(ch)$

1.1 Données

Opérateur de test	En Python	Dans un algorithme
est bien égal à	==	=
\neq	!=	\neq
<	<	<
>	>	>
\leq	<=	\leq
\geq	>=	\geq

Exercices

1. Dans la console Python, que retourne chacune des expressions suivantes ? Justifier.

a) $-6-1+2*4$

b) $1 / (0.2+0.3)+1$

c) $1/4-3*5/2$

d) $((6+1) / (6-1)+1) *2+1$

2. Dans la console Python, que retourne chacune des expressions suivantes ? Justifier.

a) $16//3+16\%3$

b) $43221\%3$

c) $(10+2)\%5-5$

d) $(2017//2016) *1000$

3. Dans la console Python, que retourne chacune des expressions suivantes ? Justifier.

a) `"5"+"101"`

b) `3*"5"+2*"101"`

c) `"Zida"+"ne"+3*!"`

d) `5*"101"+"5"`

e) `5*len("101")+5`

f) `len("5"+"101")`

g) `"Ronaldo"[1]+"k?"`

h) `"jouer"[0:3]+"ez"+3*!"`

Chapitre 1 : Outils

4. Déterminer le résultat de chacune des opérations suivantes en précisant son type (entier, flottant ou chaîne de caractères).

Vérifier ensuite ce résultat avec la console Python.

- | | |
|--|--|
| a) <code>11+2*3</code> | b) <code>"11"+2*"3"</code> |
| c) <code>str(11)+2*"3"</code> | d) <code>"11+2"[1:3]</code> |
| e) <code>11+longueur("2")*3</code> | f) <code>("11"+2*"3")[2]</code> |
| g) <code>str(11)+str(2*3)</code> | h) <code>2*str(11+3)</code> |
| i) <code>longueur(str(11))+2*3</code> | j) <code>float("14.78")</code> |
| k) <code>int("2017"[3])+2.5</code> | l) <code>int("84")+longueur("84")</code> |
| m) <code>3*str(longueur(str(11))+2)</code> | |

5. Voici des captures d'écran de la console Python sur lesquelles des taches cachent certaines valeurs. Retrouver **mentalement** chacune de ces valeurs. Vérifier ensuite les résultats par saisie des opérations.

a) Capture d'écran n°1

```
>>> 2*3+4-5+6//7
```



b) Capture d'écran n°2

```
>>> (5*"5j")[0]
```



c) Capture d'écran n°3

```
>>> len(str(10**7))
```



d) Capture d'écran n°4

```
>>> (7*233)%10
```



Corrigé

1. a) $-6-1+2 \times 4$ retourne 1.

En effet : $-6 - 1 + 2 \times 4 = -7 + 8 = 1$

b) $1 / (0.2+0.3) + 1$ retourne 3.0.

En effet : $\frac{1}{0,2+0,3} + 1 = \frac{1}{0,5} + 1 = 2 + 1 = 3$

c) $1/4-3 \times 5/2$ retourne -7.25.

En effet : $\frac{1}{4} - 3 \times \frac{5}{2} = 0,25 - 7,5 = -7,25$

d) $((6+1) / (6-1) + 1) \times 2 + 1$ retourne 5.8.

En effet : $\left(\frac{6+1}{6-1} + 1\right) \times 2 + 1 = \left(\frac{7}{5} + 1\right) \times 2 + 1 = 2,4 \times 2 + 1 = 5,8$

2. a) $16 // 3 + 16 \% 3$ retourne 6.

En effet, le quotient et le reste de la division euclidienne de 16 par 3 sont respectivement 5 et 1, et on a $5 + 1 = 6$.

b) $43221 \% 3$ retourne 0.

En effet, 43221 est divisible par 3 (la somme de ses chiffres est 12, qui est bien divisible par 3) ; le reste de la division euclidienne de 43221 par 3 est donc nul.

c) $(10+2) \% 5 - 5$ retourne -3.

En effet, le reste de la division euclidienne de $10 + 2 = 12$ par 5 est 2, et $2 - 5 = -3$.

d) $(2017 // 2016) \times 1000$ retourne 1000.

En effet, le quotient de la division euclidienne de 2017 par 2016 est 1, et $1 \times 1000 = 1000$.