

Instruction conditionnelle

NUMWORKS

Au sein d'une fonction, il peut être intéressant d'exécuter des instructions sous certaines conditions.

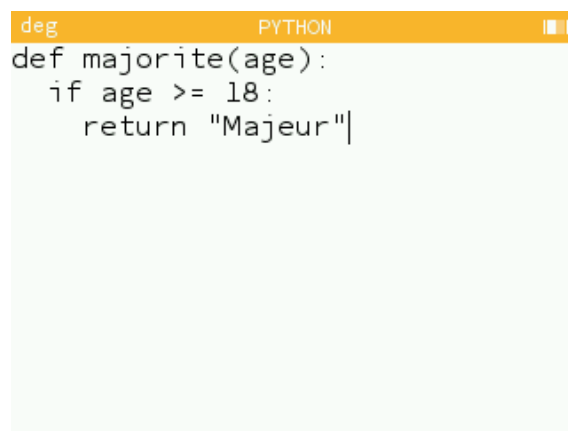
1 If ... elif ... else

Par exemple, si vous souhaitez écrire une fonction `majorite(age)` qui renvoie "Majeur"/"Mineur" selon l'âge renseigné, il est nécessaire de séparer les cas où l'âge est supérieur à 18 et celui où il est inférieur.

En Python, un bloc d'instruction conditionnelle s'écrit en indiquant :

```
if condition:
    instruction
```

L'instruction n'est exécutée que si la condition suivant `if` est vérifiée. Les deux points servent à indiquer à Python que vous commencez un bloc d'instructions.



```
def majorite(age):
    if age >= 18:
        return "Majeur"
```

Pour notre exemple, on peut écrire :

```
deg PYTHON
>>> from exemple import *
>>> majorite(25)
'Majeur'
>>> |
```

La fonction ci-dessus répond donc le texte "Majeur" si l'âge indiqué entre parenthèses est supérieur ou égal à 18. Voir l'application ci-contre.

```
deg PYTHON
def majorite(age):
    if age >= 18:
        return "Majeur"
    else:
        return "Mineur"
```

Si l'on souhaite ajouter la possibilité de répondre "Mineur", il faut ajouter un cas. On souhaite que la fonction réponde "Majeur" si l'âge est supérieur à 18 et "Mineur" sinon. Pour signifier ce SINON à Python, on utilise **else** suivi de deux points avec la même indentation que le premier **if**.

En résumé :

- S'il n'y a qu'un seul cas à distinguer, on utilisera :

```
if condition:
    instruction
```

- S'il n'y a que deux cas à distinguer, on utilisera :

```
if condition:
    instruction_1
else:
    instruction_2
```

- S'il y a plus de deux cas, on utilisera **elif** pour ajouter des conditions :

```
if condition_1:
    instruction_1
elif condition_2:
```

```
        instruction_2
elif condition_3:
        instruction_3
...
```

2 Les conditions

Voici la syntaxe des conditions dans Python :

Condition	Syntaxe Python
Si x est égal à y	<code>if x == y:</code>
Si x est différent de y	<code>if x != y:</code>
Si x est strictement supérieur à y	<code>if x > y:</code>
Si x est strictement inférieur à y	<code>if x < y:</code>
Si x est supérieur ou égal à y	<code>if x >= y:</code>
Si x est inférieur ou égal à y	<code>if x <= y:</code>

Si l'on souhaite vérifier deux conditions on utilise **and** entre les deux conditions. Par exemple :

```
if x == 1 and y > 0:
```

Si l'on souhaite vérifier l'une ou l'autre des deux conditions, on utilise **or** entre les deux conditions. Par exemple :

```
if x >= 1 or x == 0:
```

Appuyez sur la touche Toolbox de votre calculatrice pour faire apparaître un menu de raccourcis. Dans **Boucles et tests** vous trouverez des blocs d'instructions pré-remplis pour vous éviter d'écrire lettre par lettre au clavier.

3 Exercice

Écrire une fonction `vabsolu(x)` qui prend un réel en argument et renvoie sa valeur absolue.

4 Autre exercice

Écrire une fonction `mediane` qui prend une série de nombres de taille quelconque en argument et qui renvoie la médiane de la série.