

TP 2 : Les conditions

1 Les conditions « SI »

L'instruction `if` permet d'exécuter une partie d'un code **si** une condition est vraie. Elle permet, si on le souhaite, d'exécuter une autre partie d'un code si cette condition n'est pas vraie ou d'ajouter une nouvelle condition.

En programmation, il est parfois utile de réaliser un branchement : si une condition est réalisée alors le programme exécute les instructions de la première branche ; sinon le programme exécute les instructions de la deuxième branche. La syntaxe utilisée dans les langages de programmation, basée sur les mots clés `if`, `else`, est présentée ci-dessous. Voici la structure à respecter pour définir une condition en `Python` :

- Si ... alors ...

```
if condition :
    instructions
```
- Si ..., alors ..., sinon ...

```
if condition :
    instructions1
else :
    instructions2
```
- Si ..., alors ..., sinon si..., alors ..., sinon ...

```
if condition1 :
    instructions1
elif condition2 :
    instructions2
else :
    instructions3
```

Remarque : Dès qu'une condition est vérifiée, l'instruction correspondante est exécutée et les conditions suivantes ne sont pas exécutées.

Exemple : Dans l'éditeur, taper les lignes suivantes et exécuter ce programme :

```
Python
1 x = int(input('Entrer votre âge :'))
2 if x >= 18 :
3     print('vous êtes majeur')
4 else :
5     print('vous êtes mineur')
```

Que fait ce programme ?

2 Exercices

Exercice 1

Écrire une fonction qui prend comme argument un nombre réel x et renvoie la racine carrée de celui-ci quand cela est possible, ou un message d'erreur.

Exercice 2

Écrire une fonction qui prend pour argument :

- le tarif normal d'une place de cinéma
- l'âge d'une personne

Le programme doit ensuite afficher le prix à payer, sachant qu'une réduction de 50% est accordée pour les plus de 65 ans, et que c'est 4 euros pour les moins de 14 ans.

Exercice 3

Que fait le programme suivant :

 Python

```
1 def fonction(a,b):
2     if a >= b:
3         return a
4     else:
5         return b
```

Exercice 4

Que fait le programme suivant :

 Python

```
1 from math import *
2 def f(x) :
3     if abs(x) >= 2:
4         return exp(x)
5     elif x >= 0 and x < 2:
6         return 3**x
7     else:
8         return 4-x**2
```

Exercice 5

Que fait le programme suivant :

 Python

```
1 n = int(input("Entrer un nombre entier : "))
2 if n/2 == int(n/2):
3     print(n/2)
4 else:
5     print((n-1)/2)
```

Exercice 6

Écrire une fonction qui prend comme argument un nombre réel x et renvoie 1 si $x \in [0, 1]$ et 0 sinon.

Exercice 7

Écrire une fonction qui prend comme argument un nombre réel x et renvoie la valeur absolue de x .

Exercice 8

Écrire une fonction qui prend comme argument un nombre réel x et renvoie 1 si $x \in]-\infty, \ln(2)[\cup]5, +\infty[$ et 0 sinon.

Exercice 9

Écrire une fonction qui prend comme arguments trois paramètres a, b, c et qui renvoie les solutions réelles de l'équation $ax^2 + bx + c = 0$.

Exercice 10

Écrire un programme, sous Python, qui demande à l'utilisateur de saisir une année et affiche si l'année en question est une année bissextile.

Rappel : une année est bissextile si l'année est divisible par 4 et non divisible par 100 ou si l'année est divisible par 400.