

Calculs

Les 4 opérations : + - * / **

Division entière : // %

Fonctions : `abs()` `round()`

Variables

Noms des variables : 1er caractère = `a...zA...Z_` suivi de `a...zA...Z_0...9` (accents possibles, distinction casse min/MAJ)

Affectations

```
n = 23          # Entier
x = 23.0        # Décimal flottant
s = "23"        # Chaîne de caractères
a = True        # Booléen (True ou False)
n += 1          # Incrémentation
```

Fonctions

```
def double(x):
    return x * 2
```

```
y = double(2)
assert double(2) == 4          # Rien si vrai, Erreur si faux
```

Entrées - sorties

```
age = int(input("entrez un entier: "))
x = float(input("entrez un décimal: "))
print("J'ai",age,"ans")
```

Tests

```
if a==0:
    print("a est nul")
elif a>0:
    print("a est strictement positif")
else:
    print("a est strictement négatif")
```

Comparaisons

`==` `!=` `<` `<=` `>` `>=` `and` `or` `not`

Ne pas confondre : `=` (affectation) avec `==` (comparaison)

Boucle for

```
for item in « itérable »: (= range(n), chaînes de caractères et listes)
    print(item)
```

Boucle while

```
a = 0          # Initialiser la variable de condition
while a<10:
    print(a)
    a += 1      # Et la modifier dans la boucle
```

Chaînes de caractères

```
nom = "Brad" + " " + "Pitt"      # Concaténation
nom = nom.upper()                 # Majuscules
nom = nom.lower()                 # Minuscules
print(f"{nom} a {age} ans")
```

Listes

```
liste_vide = []
prénoms = ["Pierre","Paul","Jacques"]
prénoms.append("Jean")
prénoms.pop()                     # ="Jean"
prénoms = sorted(prénoms)
élèves = [["Pierre",105], ["Paul",107]]
élèves[0][1]                     # =105
```

Itérables (= range(n), chaînes de caractères et listes)

```
nom = "Brad Pitt"
len(nom)                          # =9
nom[0]                            # ="B"
nom[-1]                           # ="t"
"P" in nom                        # =True
```

Module Math

```
from math import *               # Chargement du module en mémoire
sqrt(x)                          # Racine carrée
floor(x)                         # Arrondi à l'entier inférieur
ceil(x)                          # Arrondi à l'entier supérieur
sin(x), cos(x), tan(x), asin(x)
pi                               # Nombre pi
```

Module Random

```
from random import *
random()                          # Flottant sur [0 ; 1[
randint(a,b)                     # Entier entre a et b inclus
```