

Python

Python est un langage de programmation populaire et polyvalent, adapté aux débutants et aux développeurs expérimentés.

Voici quelques points clés à retenir :



Types de variables - fonctions de conversions

Type integer Nombre entier

int() convertit si possible un décimal ou texte en entier

Type float Nombre décimal

float() convertit si possible un entier ou texte en décimal

Type string Chaîne de caractères (texte) suite
de signes définie en la délimitant par des guillemets

str() convertit un nombre en chaîne

Type boolean Logique

ne prend que deux valeurs : **True** et **False**

Affectation=

x =... lire « x prend la valeur....»

```
x="12" # x est de type string
y=3    # y est du type integer
z=x+y   # erreur
z=int(x)+y # donne 15
z=x+str(y) # donne "123"
```

```
a=(1+1>2) # a booléen qui vaut False
b=(2*3==6) # b booléen qui vaut True
```

```
a,b=12,15 # équivaut à a=12 et b =15
x+=2      # équivaut à x=x+2
x-=1      # équivaut à x=x-1
```



Entrées, sorties console, opérations numériques

Entrée `input` (" message") : lit un texte saisi au clavier .

👉 Renvoie donc toujours une chaîne de caractères.

👉 conversion possible en nombre par `int()` ou `float()`

Sortie en console `print(, , ...)` : affiche en console les valeurs de tout type en les séparant par une tabulation.

Opérations sur les nombres

/ division décimale

// quotient de la division entière

% reste de la division entière

****** puissance (remarque : `a**0.5` a)

`abs()` valeur absolue

`round(x,d)` arrondi le nombre x à d décimales

```
a=input("Texte?")    # a    type string  
b=float(input("Nombre?")) # b type float
```

```
a=45*2  
print("Coût=",a,"€") # affiche "Coût= 90 €"
```

```
d=11/4    # d vaut 2.75  
q=11//4    # q vaut 2  
r=11%4    # r vaut 3 car 11= 2*4+3  
p=4**3    # p vaut 64  
r=16**0.5 # r vaut racine(16) soit 4  
x=abs(7-10) # x vaut 3  
y=round(4/3,4) # y vaut 1,3333
```



Chaînes de caractères

Concaténation + attache les textes pour n'en former qu'un **Caractères d'échappement**

le signe **** permet de transformer le caractère qui suit **\n** saut de ligne (new). **\t** tabulation **\"** ou **\'** guillemet qui ne ferme pas la chaîne
longueur d'une chaîne : **len()** renvoie le nombre de caractères d'une chaîne, espaces compris.

Indexation

Chaque caractère de la chaîne est indexé (numéroté) en commençant par 0 **Chaîne[i]** renvoie le caractère de rang i

👉 astuces : **MaChaîne[-1]** dernier caractère

MaChaîne[-2] avant dernier caractère, etc...

MaChaîne[i : j] caractères indexés de i à j-1

👉 Attention : on en peut pas modifier un caractère d'une chaîne par son index, seulement le lire !

```
T1="Ceci est" # variable type string
T2=" un test." # variable type string
T=T1+T2      # T vaut "Ceci est un test"
```

```
T="ceci est \n un test"# Imprime: ceci est
print(T)                #                un test
T="Il m\'a dit:\"Attention\""
print(T) # imprime : Il m'a dit:"Attention"
```

```
T="Ceci est un test"
L=len(T)      # entier valant 16
print(T[0])   # écrit 'C'
x=T[2]        # x vaut 'c'
y=T[L]        # erreur de dépassement
z=T[L-1]      # z vaut 't' (dernière lettre)
z=T[-1]       # z vaut 't' (dernier index)
z=T[-2]       # z vaut 't' (avant dernier index)
z=T[1:6]      # z vaut 'eci e' (indexe 1 à 5)
```

```
T="Pithon"
T[1]="y" # erreur: affectation non autorisée
```



Tests : Les conditions

if test 1 : # un test est une valeur booléenne(logique)

| bloc si test1 vérifié

elif test 2 : # (facultatif). Sinon si :

| bloc si test 1 non vérifié mais test2 vérifié

else : # (facultatif). Sinon

| bloc si aucun des tests précédent n'est vérifié suite du programme

comparateurs

== # égal

!= # différent

> # supérieur

>= # sup ou égal

```
a=float(input("Donne un nombre"))
if a==0:
    texte ="nul"
elif a>=0:
    texte= "positif"
else:
    texte = "négatif"
print(texte)
```

Suite

DES EXERCICES

Suite

DES EXERCICES

1. Écrivez un programme qui demande à l'utilisateur de saisir un nombre, puis vérifie s'il est positif, négatif ou nul.
2. Créez un programme qui vérifie si un nombre donné est pair ou impair.
3. Demandez à l'utilisateur de saisir son âge, puis déterminez s'il est majeur (âge ≥ 18) ou mineur.
4. Écrivez un programme qui demande le prénom de l'utilisateur et affiche un message de bienvenue personnalisé.
5. Demandez à l'utilisateur de saisir deux nombres et effectuez une opération (addition, soustraction, multiplication, division) de son choix sur ces nombres.
6. Créez un programme qui demande à l'utilisateur de saisir une longueur en mètres et convertit cette longueur en pieds (1 mètre équivaut à environ 3,281 pieds).
7. Écrivez un programme qui permet à l'utilisateur de saisir un nombre entier, puis affiche son carré et son cube.
8. Demandez à l'utilisateur de saisir deux chaînes de caractères (mots) et concaténez-les pour former une nouvelle chaîne.
9. Créez un programme qui permet à l'utilisateur de saisir la base et la hauteur d'un triangle, puis calcule et affiche sa superficie ($0,5 * \text{base} * \text{hauteur}$).
10. Écrivez un programme qui prend une phrase en entrée et compte le nombre de mots dans la phrase.
11. Demandez à l'utilisateur de saisir un mot et affichez-le à l'envers (par exemple, "Python" devrait devenir "nohtyP").
12. Créez un programme qui prend un prénom en entrée et affiche un message de salutation personnalisé en utilisant la fonction `.format()`.