



LES NOMBRES



01

DÉFINITION

Qu'est ce qu'une donnée numérique?

03

CONVERSION DE TYPE

Passer d'un type numérique à un autre.

05

FONCTIONS ALÉATOIRES

Comment générer un nombre aléatoire?

02

TYPES NUMÉRIQUES

Quels sont les différents types numériques?

04


FONCTIONS MATHÉMATIQUES

Quelles sont les opérations mathématiques disponibles?

06

FONCTIONS TRIGONOMÉTRIQUES

Quelles sont les opérations trigonométriques disponibles?





DÉFINITION

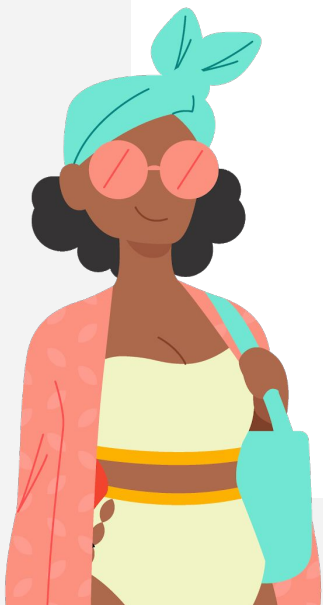
Les **types de données numériques** stockent des **valeurs numériques**.
Ce sont des types de données **immuables**, ce qui signifie que la modification de la valeur entraîne la création d'un nouvel objet.

Exemples :

```
nb_ananas = 1  
print( id(nb_ananas) )  
nb_ananas = 3  
print( id(nb_ananas) )
```

Resultats :

```
140226850670304  
  
140226850670336
```





TYPES NUMÉRIQUES

Il existe quatre types numériques différents.

int

Ce sont des nombres entiers positifs ou négatifs sans décimale.

Exemple : 3, 10, -78, -15

long

Ce sont des entiers de taille illimitée, écrits comme des entiers et suivis d'un L majuscule ou minuscule.

Exemple : 0122L, 24361L, -4782L

float

Ils représentent des nombres réels (décimaux) et sont écrits avec une virgule. Ils peuvent être en notation scientifique, e (ou E) indiquant la puissance de 10 ($2,5e^2 = 2,5 \times 10^2 = 250$).

Exemple : 0.0, 45., 12.36, -7.1, -99

complex

Ils sont de la forme $a + bj$, où a et b sont des floats et j (ou J) est la racine carrée de -1.

La partie réelle du nombre est a, et la partie imaginaire est b.

Exemple : 3.14j, 10+2.3j, -78+7j, 1.5-3.0j



CONVERSION DE TYPE

Python convertit les nombres **implicitement** pour évaluer les opérations. Mais parfois, il faut convertir **explicitement** l'un des membres de l'opération afin de satisfaire les exigences de l'opérateur.

int(x) : pour convertir x en un nombre entier simple.

long(x) : pour convertir x en un entier long.

float(x) : pour convertir x en un nombre à virgule flottante.

complex(x) : pour convertir x en un nombre complexe avec une partie réelle x et une partie imaginaire zéro.

complex(x, y) : pour convertir x et y en un nombre complexe avec une partie réelle x et une partie imaginaire y. x et y sont des expressions numériques.





FONCTIONS MATHÉMATIQUES

abs(x)

La valeur absolue de x.

La distance (positive) entre x et zéro.

ceil(x)

Le plus petit nombre entier supérieur à x.

exp(x)

Exponentielle de x : e^x .

```
import math
```

```
print("abs(-15) : ", abs(-15))
```

```
print("abs(18) : ", abs(18))
```

```
print("math.ceil(math.pi) : ", math.ceil(math.pi))
```

```
print "math.ceil(18) : ", math.ceil(18)
```

```
print( "math.exp(1) : ", math.exp(1))
```

Résultats :

```
abs(-15) : 15
```

```
abs(18) : 18
```

```
math.ceil(math.pi) : 4.0
```

```
math.ceil(18) : 18.0
```

```
math.exp(1) : 2.718281828459045
```



FONCTIONS MATHÉMATIQUES

fabs(x)

La valeur absolue de x.

floor(x)

Le plus grand nombre entier inférieur à x.

log(x)

Le logarithme naturel de x, pour $x > 0$.

log10(x)

Le logarithme de base 10 de x pour $x > 0$.

```
import math
```

```
print("math.fabs(-15.48) : ", math.fabs(-15.48))  
print("math.floor(18) : ", math.floor(18))  
print("math.floor(math.pi) : ", math.floor(math.pi))  
print("math.log(172.14) : ", math.log(172.14))  
print("math.log10(math.pi) : ", math.log10(math.pi))  
print("math.log10(-121) : ", math.log10(-121))
```

Résultats :

```
math.fabs(-15.48) : 15.48  
math.floor(18) : 18.0  
math.floor(math.pi) : 3.0  
math.log(172.14) : 5.148308099221328  
math.log10(math.pi) : 0.497149872694  
math.log10(121) : 2.08278537031645
```



FONCTIONS MATHÉMATIQUES

pow(x, y)

La valeur de x^y .

x a la puissance y.

round(x [,n])

x arrondi à n chiffres à partir de la virgule.

round(0,5) est 1,0 et round(-0,5) est -1,0.

sqrt(x)

La racine carrée de x pour $x > 0$.

```
import math
```

```
print("math.pow(100, -2) : ", math.pow(100, -2))
```

```
print("math.pow(2, 4) : ", math.pow(2, 4))
```

```
print("round(80.23456, 2) : ", round(80.23456, 2))
```

```
print("round(100.000056, 3) : ", round(100.000056, 3))
```

```
print("math.sqrt(7) : ", math.sqrt(7))
```

```
print("math.sqrt(math.pi) : ", math.sqrt(math.pi))
```

Résultats :

```
math.pow(100, -2) : 0.0001
```

```
math.pow(2, 4) : 16.0
```

```
round(80.23456, 2) : 80.23
```

```
round(100.000056, 3) : 100.0
```

```
math.sqrt(7) : 2.64575131106
```

```
math.sqrt(math.pi) : 1.77245385091
```




FONCTIONS MATHÉMATIQUES

max(x1, x2,...)

Le plus grand de ses arguments : la valeur la plus proche de l'infini positif.

min(x1, x2,...)

Le plus petit de ses arguments : la valeur la plus proche de l'infini négatif.

modf(x)

Les parties fractionnaires et entières de x dans un tuple à deux éléments. La partie entière est renvoyée sous forme de flottant.

```
import math
```

```
print("max(-20, 100, 400) : ", max(-20, 100, 400))  
print("max(-80, -20, -10) : ", max(-80, -20, -10))  
print("min(80, 100, 1000) : ", min(80, 100, 1000))  
print("min(0, 100, -400) : ", min(0, 100, -400))  
print("math.modf(math.pi) : ", math.modf(math.pi))  
print("math.modf(100.12) : ", math.modf(100.12))
```

Résultats :

```
max(-20, 100, 400) : 400  
max(-80, -20, -10) : -10  
min(80, 100, 1000) : 80  
min(0, 100, -400) : -400  
math.modf(math.pi) : (0.14159265358979312, 3.0)  
math.modf(100.12) : (0.120000000000000455, 100.0)
```



FONCTIONS ALÉATOIRES

Les nombres aléatoires sont utilisés pour les jeux, les simulations, les tests, les applications de sécurité et de protection de la vie privée. Python comprend les fonctions suivantes qui sont couramment utilisées.

choice(seq)

Un élément aléatoire d'une liste, d'un tuple ou d'une chaîne de caractères.

randrange([start,] stop [,step])

Un élément choisi au hasard entre start inclus et stop exclu en itérant de step.

```
import random
```

```
print("choice([1, 3, 5, 9]) : ", random.choice([1, 3, 5, 9]))  
print("choice('Mango') : ", random.choice('Mango'))  
print("randrange(1, 10, 2) : ", random.randrange(1, 10, 2))  
print("randrange(1, 10, 3) : ", random.randrange(1, 10, 3))
```

Résultats :

```
choice([1, 3, 5, 9]) : 5  
choice('Mango') : n  
randrange(1, 10, 2) : 3  
randrange(1, 10, 3) : 7
```



FONCTIONS ALÉATOIRES

random()

Un float compris entre 0 et 1.

seed([x])

Utilisé pour générer des nombres aléatoires avec un point de départ. Les prochains nombres aléatoires générés seront identiques si la seed ne change pas.

shuffle(list)

Mélange aléatoirement les éléments d'une liste.
Ne renvoie aucun élément.

uniform(x, y)

Un float compris entre x et y.

```
import random
```

```
list = [20, 16, 10, 5]
```

```
print("random() : ", random.random(), random.random())
```

```
random.seed(10)
```

```
print("Random number : ", random.random())
```

```
print("Random number : ", random.random())
```

```
random.shuffle(list)
```

```
print("Reshuffled list : ", list)
```

```
print("Random uniform(5, 10) : ", random.uniform(5, 10))
```

Résultats :

```
random() : 0.28196520525, 0.85235469178
```

```
Random number : 0.57140259469
```

```
Random number : 0.57140259469
```

```
Reshuffled list : [16, 5, 10, 20]
```

```
Random uniform(5, 10) : 5.52615217015
```



FONCTIONS TRIGONOMÉTRIQUES

acos(x)

Retourne l'arc cosinus de x, en radians.

asin(x)

Retourne l'arc sinusoidal de x, en radians.

atan(x)

Retourne l'arc tangent de x, en radians.

cos(x)

Retourne le cosinus de x, en radians.

```
import math
print("acos(0.64) :", math.acos(0.64))
print("acos(0) :", math.acos(0))
print("asin(0.64) :", math.asin(0.64))
print("asin(1) :", math.asin(1))
print("atan(0.64) :", math.atan(0.64))
print("atan(-1) :", math.atan(-1))
print("cos(3) :", math.cos(3))
```

Résultats :

```
acos(0.64) : 0.876298061168
acos(0) : 1.57079632679
asin(0.64) : 0.694498265627
asin(1) : 1.57079632679
atan(0.64) : 0.569313191101
atan(-1) : -0.785398163397
cos(3) : -0.9899924966
```



FONCTIONS TRIGONOMÉTRIQUES

sin(x)

Retourne le sinus de x, en radians.

tan(x)

Retourne la tangente de, x en radians.

degrees(x)

Convertit l'angle de x radians en degrés.

radians(x)

Convertit l'angle de x degrés en radians.

```
import math
print "sin(0) : ", math.sin(0)
print "sin(math.pi/2) : ", math.sin(math.pi/2)
print "tan(math.pi) : ", math.tan(math.pi)
print "tan(math.pi/4) : ", math.tan(math.pi/4)
print "degrees(math.pi/2) : ", math.degrees(math.pi/2)
print "degrees(math.pi/4) : ", math.degrees(math.pi/4)
print "radians(math.pi) : ", math.radians(math.pi)
```

Résultats :

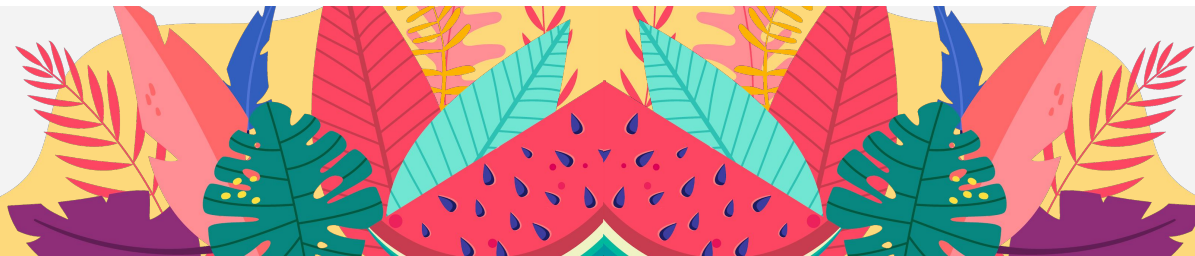
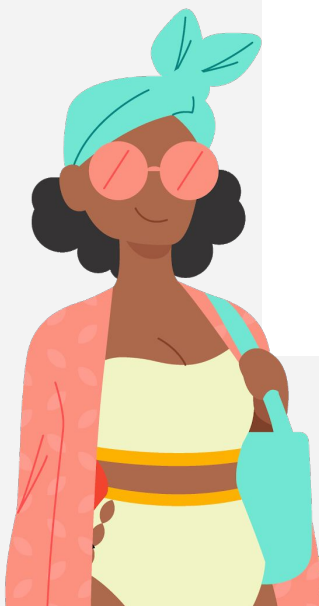
```
sin(0) : 0.0
sin(math.pi/2) : 1.0
tan(math.pi) : -1.22460635382e-16
tan(math.pi/4) : 1.0
degrees(math.pi/2) : 90.0
degrees(math.pi/4) : 45.0
radians(math.pi) : 0.0548311355616
```

The background of the slide is a vibrant, colorful pattern of various tropical leaves and foliage. The leaves are in shades of green, yellow, orange, red, and purple, creating a dense and lively border around the central text. Some leaves are large and detailed, while others are smaller and more stylized. The overall effect is a tropical and energetic aesthetic.

EXERCICES

EXERCICES

1. Quels sont les 4 types numériques?
2. Créez une variable de type float et convertissez la en un entier.
3. Créez 3 variables x , y , z et attribuez leur les valeurs suivantes respectivement : 1.05, 12, 37.5
4. Affichez le plus petit nombre entier supérieur à x .
5. Affichez l'exponentielle de y .
6. Affichez le plus grand nombre entier inférieur à z .
7. Affichez la racine carré de y .



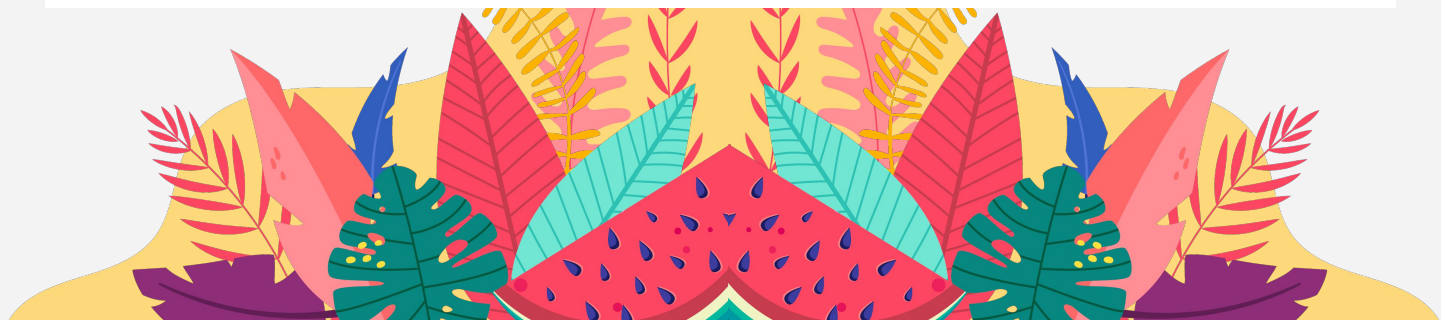
EXERCICES

8. Créez une variable `u`, et lui assigner une valeur aléatoire comprise entre 0 et 100 avec un pas de 20 en utilisant la fonction `randrange()`.

9. Si la variable `u` est supérieure ou égale à 50, affichez un float compris entre 50 et 100 en utilisant la fonction `uniform()`, sinon afficher un float compris entre 0 et 1 avec la fonction `random()`.

10. Assignez le résultat de l'addition entre la variable `x` élevée à la puissance 7 et la variable `u` dans une variable `v`.

11. Calculez le sinus de `v` et affichez le résultat en degrés.



A vibrant, colorful background featuring a dense pattern of various tropical leaves and foliage. The leaves are in shades of green, yellow, orange, red, and purple, creating a lush and energetic feel. Some leaves have detailed vein patterns, while others are solid colors. The overall composition is symmetrical and balanced, with the central text area clearly defined.

SOLUTIONS

CORRECTION

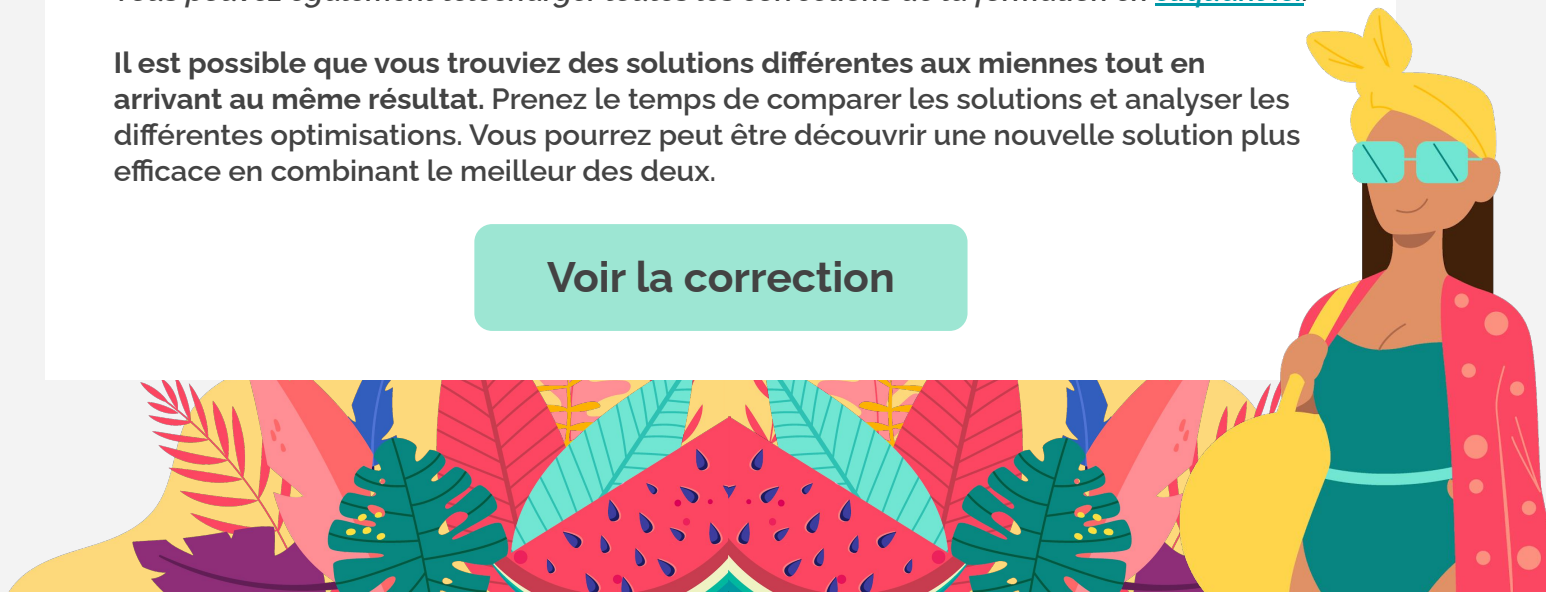
Pour visualiser la correction du chapitre cliquer sur le bouton ci-dessous.

*Le fichier de la correction s'ouvrira dans un nouvel onglet de votre navigateur préféré.
Pour cela vous devez avoir accès à une connexion Internet.*

Vous pouvez également télécharger toutes les corrections de la formation en [cliquant ici](#).

Il est possible que vous trouviez des solutions différentes aux miennes tout en arrivant au même résultat. Prenez le temps de comparer les solutions et analyser les différentes optimisations. Vous pourrez peut être découvrir une nouvelle solution plus efficace en combinant le meilleur des deux.

Voir la correction





**Félicitation vous avez terminé
le chapitre sur les nombres
avec Python!**

A decorative border of various tropical leaves in vibrant colors like red, orange, yellow, green, and purple surrounds the central white area. The leaves include Monstera, palm, and other exotic foliage.

CRÉDITS

- Modèle de la présentation par [Slidesgo](#)
- Icônes par [Flaticon](#)
- Images et infographies par [Freepik](#)