

Cours - math

Bibliothèque `math`

La bibliothèque `math` en Python fournit de nombreuses fonctions mathématiques pour effectuer des calculs scientifiques et mathématiques. Cette bibliothèque inclut des fonctions pour les opérations arithmétiques, trigonométriques, logarithmiques, et plus encore.

1. Importer la bibliothèque `math`

Avant de pouvoir utiliser les fonctions de la bibliothèque `math`, vous devez l'importer dans votre script Python :

```
import math
```

2. Constantes Mathématiques

La bibliothèque `math` fournit plusieurs constantes utiles :

- `math.pi` : La constante π (pi), environ 3.14159.
- `math.e` : La constante e (base des logarithmes naturels), environ 2.71828.

```
print(math.pi)  # 3.141592653589793
print(math.e)   # 2.718281828459045
```

3. Fonctions de Base

3.1. Fonctions Arithmétiques

- `math.ceil(x)` : Renvoie le plus petit entier supérieur ou égal à x.
- `math.floor(x)` : Renvoie le plus grand entier inférieur ou égal à x.
- `math.sqrt(x)` : Renvoie la racine carrée de x.
- `math.factorial(x)` : Renvoie la factorielle de x (x!).
- `math.fabs(x)` : Renvoie la valeur absolue de x (en float).

```
print(math.ceil(4.2))    # 5
print(math.floor(4.8))   # 4
print(math.sqrt(16))     # 4.0
print(math.factorial(5)) # 120
print(math.fabs(-7.5))   # 7.5
```

3.2. Fonctions de Puissance et Logarithme

- `math.pow(x, y)` : Renvoie x élevé à la puissance y.
- `math.exp(x)` : Renvoie e élevé à la puissance x.
- `math.log(x, base)` : Renvoie le logarithme de x dans la base spécifiée. Si la base n'est pas spécifiée, le logarithme naturel est utilisé.
- `math.log10(x)` : Renvoie le logarithme base 10 de x.

```
print(math.pow(2, 3))    # 8.0
print(math.exp(2))       # 7.3890560989306495
print(math.log(8, 2))    # 3.0
print(math.log10(100))   # 2.0
```

4. Fonctions Trigonométriques

4.1. Fonctions de Base

- `math.sin(x)` : Renvoie le sinus de x (en radians).

- `math.cos(x)` : Renvoie le cosinus de x (en radians).
- `math.tan(x)` : Renvoie la tangente de x (en radians).

```
print(math.sin(math.pi/2)) # 1.0
print(math.cos(0))         # 1.0
print(math.tan(math.pi/4)) # 1.0
```

4.2. Fonctions Inverses

- `math.asin(x)` : Renvoie l'arc-sinus de x (en radians).
- `math.acos(x)` : Renvoie l'arc-cosinus de x (en radians).
- `math.atan(x)` : Renvoie l'arc-tangente de x (en radians).

```
print(math.asin(1)) # 1.5707963267948966 (π/2)
print(math.acos(1)) # 0.0
print(math.atan(1)) # 0.7853981633974483 (π/4)
```

4.3. Conversion entre Degrés et Radians

- `math.degrees(x)` : Convertit un angle de radians en degrés.
- `math.radians(x)` : Convertit un angle de degrés en radians.

```
print(math.degrees(math.pi)) # 180.0
print(math.radians(180))     # 3.141592653589793
```

5. Fonctions Hyperboliques

- `math.sinh(x)` : Renvoie le sinus hyperbolique de x.
- `math.cosh(x)` : Renvoie le cosinus hyperbolique de x.
- `math.tanh(x)` : Renvoie la tangente hyperbolique de x.

```
print(math.sinh(1)) # 1.1752011936438014
print(math.cosh(1)) # 1.5430806348152437
print(math.tanh(1)) # 0.7615941559557649
```

6. Autres Fonctions Utiles

- `math.gcd(a, b)` : Renvoie le plus grand commun diviseur de a et b.
- `math.isqrt(x)` : Renvoie la racine carrée entière de x.
- `math.lcm(a, b)` : Renvoie le plus petit commun multiple de a et b (disponible depuis Python 3.9).

```
print(math.gcd(48, 180)) # 12
print(math.isqrt(17))    # 4
print(math.lcm(4, 5))    # 20
```

Exemples Pratiques

1. Calculer l'aire d'un cercle :

Écrire une fonction qui prend le rayon d'un cercle et retourne son aire.

```
def aire_cercle(rayon):
    return math.pi * math.pow(rayon, 2)

print(aire_cercle(5)) # 78.53981633974483
```

2. Conversion de degrés en radians et vice versa :

Écrire deux fonctions, l'une pour convertir des degrés en radians et l'autre pour convertir des radians en degrés.

```
def degrees_en_radians(degrees):  
    return math.radians(degrees)  
  
def radians_en_degrees(radians):  
    return math.degrees(radians)  
  
print(degrees_en_radians(180)) # 3.141592653589793  
print(radians_en_degrees(math.pi)) # 180.0
```

3. Calculer la longueur de l'hypoténuse :

Utiliser le théorème de Pythagore pour écrire une fonction qui calcule la longueur de l'hypoténuse d'un triangle rectangle.

```
def hypotenuse(a, b):  
    return math.sqrt(math.pow(a, 2) + math.pow(b, 2))  
  
print(hypotenuse(3, 4)) # 5.0
```