



Syntaxe utile

- **#** permet d'écrire des commentaires.
- **//** donne le quotient de la division
- **%** donne le reste de la division
- Sans le module **math** on ne peut pas utiliser cosinus, pi, e ...
- Les chaînes de caractères doivent être écrites entre guillemets " "
- **print** permet d'afficher quelque chose

```
1 #a et b entiers naturels non nuls
2 def div_eucl(a,b):
3     q=a//b
4     r=a%b
5     return (q,r)
6
7
8
9
```



```
>>>div_eucl(5,23)
(0, 5)
>>>div_eucl(23,5)
(4, 3)
>>>div_eucl(10,3)
(3, 1)
>>>div_eucl(8,6)
(1, 2)
>>>
```

```
1 from math import *
2 print("racine:",sqrt(2))
3 print("cosinus:",cos(pi))
4
5
6
7
8
9
```



```
MicroPython v1.9.4
(CASIO COMPUTER CO., LTD.
>>>from module import *
racine: 1.414213562373095
cosinus: -1.0
>>>
```

Instruction conditionnelle

D'autres tests peuvent être intéressants comme le test d'égalité avec **==**

```
1 def tarif(age):
2     if age<5:
3         p=0
4     elif 5<=age<12:
5         p=5
6     else:
7         p=10
8     return p
9
```



```
CASIO COMPUTER CO., LTD.
>>>from module import *
>>>tarif(3)
0
>>>tarif(26)
10
>>>tarif(2)+tarif(6)+tarif(35)
15
>>>
```

Boucle bornée

- **range(k)** : de 0 à k-1
- **range(a,b,p)** : de a à b-1 avec un pas de p
- Un **print** dans une boucle permet de voir ce qu'il se passe à chaque étape

```
1 def somme(n,m):
2     s=0
3     for i in range(n,m):
4         s=s+i
5         print(s)
6     return s
7
8
9
```



```
>>>somme(3,8)
3
7
12
18
25
>>>
```

Boucle non bornée

! Les boucles non bornées peuvent être infinies si la condition n'est jamais atteinte. Pour stopper l'exécution d'une boucle infinie, appuyez sur **(AC)** !

```
1 def repete(x,p):
2     n=0
3     while x<p:
4         n=2*x
5         n=n+1
6         return x,n
7
8
9
```



```
>>>repete(1,1024)
(1024, 10)
>>>repete(1,10)
(16, 4)
>>>repete(10,10000)
(10240, 10)
>>>
```

Nombres aléatoires et listes

- **.append** ajoute un élément à la fin de la liste
- **randint** donne un nombre entier (int) aléatoire (random) entre deux valeurs

```
1 from random import *
2 def des(n):
3     l=[]
4     for i in range(n):
5         l.append(randint(1,6))
6     return l
7
8
9
```

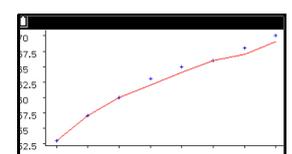


```
* SHELL Initialized *
>>>from module import *
>>>des(5)
[4, 1, 1, 5, 5]
>>>des(12)
[1, 1, 1, 1, 5, 3, 5, 6, 5, 3, 2, 6, 1]
>>>des(3)
[6, 3, 1]
* SHELL Initialized *
```

Matplotlib.pyplot

- **plot** trace une ligne brisée (en rouge dans l'exemple)
- **scatter** place les points (nuage de points en bleu dans l'exemple)

```
1 from matplotlib.pyplot import *
2
3 age=[1, 2, 3, 4, 5, 6, 7, 8]
4 taille=[53, 57, 60, 63, 65, 66, 68, 70]
5 moyenne=[53, 57, 60, 62, 64, 66, 67, 69]
6 plot(age, taille)
7 scatter(age, moyenne)
8 show()
9
```



Turtle

- **forward** fait avancer la tortue
- **left** fait tourner la tortue en degrés

```
1 from turtle import *
2 for j in range(12):
3     r=j/11
4     pencolor(r,0,1)
5     for i in range(8):
6         forward(30)
7         left(45)
8     left(30)
9
```

