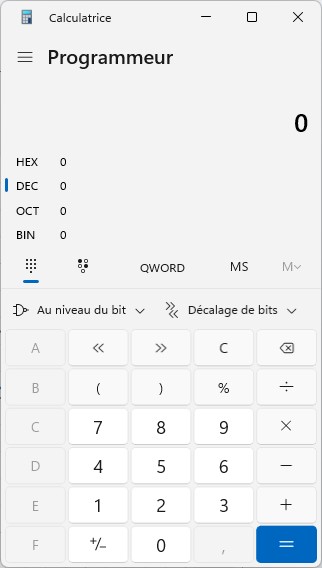
**CORRECTION – Représentation des nombres en machine**

# Partie 1 – Calculatrice Windows en mode "Programmeur"

## Mode programmeur



Ouvrez la calculatrice Windows et passez en mode programmeur.

## Identification des fonctionnalités de la calculatrice

|  |  |  |
| --- | --- | --- |
| Format | code un nombre binaire sur | |
| Byte | 8 | bits |
| Word | 16 |
| Dword | 32 |
| Qword | 64 |

## Exercices

Utilisez la calculatrice Windows pour faire les exercices suivants.

1. Convertir 1210 et -410 en binaire et en hexadécimal sur un **quartet**.

|  |  |  |
| --- | --- | --- |
|  | Binaire | Hexadécimal |
| 1210 | 11002 | C16 |
| -410 | 11002 | C16 |

Conclure

À une même représentation binaire peut correspondre un nombre décimal positif ou un nombre décimal négatif. Il faut préciser le codage : binaire naturel, en complément à 2, etc.

1. Convertir 1210 et -410 en binaire sur un **octet**.

|  |  |  |
| --- | --- | --- |
|  | Binaire | Hexadécimal |
| 1210 | 0000 11002 | 0C16 |
| -410 | 1111 11002 | FC16 |

1. Convertir 7F16 FF16 en binaire naturel sur un octet et en décimal. Convertir FFFF16 en binaire naturel sur 2 octets et en décimal.

|  |  |  |
| --- | --- | --- |
|  | Binaire | Décimal |
| 7F16 | 0111 11112 | 12710 |
| FF16 | 1111 11112 | 25510 |
| FFFF16 | 1111 1111 1111 11112 | 6553510 |

1. Effectuez les opérations ci-dessous et donnez le résultat en hexadécimal sur 32bits.

|  |  |
| --- | --- |
|  | Hexadécimal |
| 1 568 21010 - 471010 | 0017 DB8C16 |
| 471010 - 1 568 21010 | FFE8 249416 |

|  |  |
| --- | --- |
| Avertissement | Que remarquez-vous ?  La calculatrice étend le bit de signe si le résultat est négatif (il n’y a pas d’extension s’il est positif). |

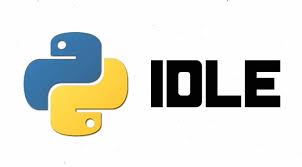
1. Les valeurs ci-dessous étant codées en complément à 2, effectuez les opérations et donnez le résultat en binaire et en hexadécimale.

**Calcul d’adresses**

|  |  |  |
| --- | --- | --- |
|  | Binaire | Hexadécimal |
| EA16 + 1C16 | ~~0001~~0000 01102 | ~~1~~0616 |
| FFEA16 + 001C16 | ~~0001~~ 0000 0000 01102 | ~~1~~000616 |

|  |  |
| --- | --- |
| Avertissement | Que remarquez-vous ?  La calculatrice affiche le débordement. L’interprétation du résultat revient à l’utilisateur. |

# Partie 2 – Les nombres en Python



*Ressources*

*- Numbers in Python (*[*https://realpython.com/python-numbers/*](https://realpython.com/python-numbers/)*)*

*- Python – Installation – Démarrage (https://bit.ly/3LIj3LR)*

## Préparation du bureau du PC

* Entrez l’URL [*https://realpython.com/python-numbers/*](https://realpython.com/python-numbers/)dans Chrome et traduisez la page (clic droit, traduire en Français).
* Lancez l’éditeur **idle** *(https://bit.ly/3LIj3LR)***.**
* Placez les deux fenêtres côte à côte comme dans l’exemple ci-dessous.

|  |  |
| --- | --- |
|  |  |

## Nombres entiers et nombres à virgule flottante

**Entiers**

1. Quelle fonction native (built-in) de Python permet de connaître (et tester) le type d’une donnée ? type()
2. Avec quel opérateur peut-on affecter une valeur numérique à une variable ? =
3. Quelle fonction native de Python permet de convertir une chaîne contenant un entier en un nombre ? int()

**Virgule flottante**

1. Comment s’affiche la valeur 6000000000000.0 dans idle ? 6e12
2. Que signifie **inf** et **-inf** ?

Les flottants maxi (positifs et négatifs) ont été dépassés sur la machine.

|  |  |
| --- | --- |
| Informations | Pour connaître la valeur maximale d’un flottant sur votre ordinateur, entrez :  # Le ; permet de mettre plusieurs commandes sur une ligne !  import sys; sys.float\_info; |

1. Quelles sont les valeurs maximum et minimum des flottants sur votre machine (arrondir à 2 décimales) ?

Ressource : <https://docs.python.org/3/library/sys.html#sys.float_info>

max = ±1,8\*10308 min = ±2,23\*10-308

## Opérateurs et expressions arithmétiques

1. Quel type de donnée renvoie les opérations 125 + 12.4, 12 - 6.1, 5 \* 3.4 et 4.2 / 2 ? Un flottant
2. Quel est le rôle des opérateurs ci-dessous. Donnez un exemple.

/ : division Exemple : 6/3 == 3.0 (La division entière renvoie toujours un flottant)

// : division entière Exemple : 6//4 == 1

\*\* : Puissance Exemple : 12\*\*2 == 144

% : modulo Exemple : 6%4 == 2

1. Quelle équation Python utilise-t-il pour calculer 8 % -2 ? Donnez l’ordre des opérations réalisées et le résultat.

Pour calculer le reste **r** de la division d'un nombre **x** par un nombre **y**, Python utilise l'équation **r = x - (y \* (x // y)).**

Ordre des opérations pour obtenir le résultat de 5%-3 c’est-à-dire **-1**

* (5//-3) = -2
* -2 \* -3 = 6
* 6 – 5 = **-1**

## Faire « mentir » Python

1. Pourquoi le résultat de l’opération 0.1 + 0.2 réalisée en Python n’est-il pas égal à 0.3 ?

C'est une **erreur de représentation en virgule flottante** qui n'a rien à voir avec Python. Elle est liée à la façon dont les nombres à virgule flottante sont stockés dans la mémoire d'un ordinateur.

## Fonctions mathématiques et méthodes numériques

1. Donnez le rôle des fonctions suivantes :

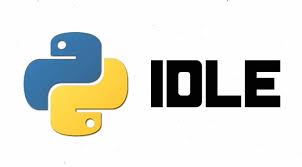
**round()** : arrondit un nombre à l'entier le plus proche .

**abs()** : obtient la valeur absolue d'un nombre.

1. Donnez le rôle de la méthode suivante :

**Is\_integer()** : renvoie vrai si l’objet est un entier.

## Programmation dans idle



Programme 1 : affichage du type d’une variable

Complétez le programme ci-dessous pour obtenir les résultats suivants :

|  |  |
| --- | --- |
| x | testnb(x) affiche |
| Un entier | C'est un entier ! |
| Un flottant | C'est un flottant ! |
| "Hello" | C'est un (suivi du type de x) ! |

Remarque : les tests se font avec le mot-clé **is**.

def testnb(x):

    # x = input('Entrer un nombre : ')

    if type(x) is int:

        print("C'est un entier")

    elif type(x) is float:

        print("C'est un flottant")

    else:

        print("C'est un ", type(x))

testnb(1) # Renvoie : « C’est un entier »

testnb(3.14) # Renvoie : « C’est un flottant »

testnb(« hello ») # Renvoie : « C’est un <class ‘str’> »

Programme 2 :

Complétez le programme ci-dessous pour qu’il pallie au problème de la représentation approximative d’un nombre flottant

(voir le cours pour la méthode) a : première valeur à tester, b : deuxième valeur à tester, p : précision

Exemple :

a= sin(2\*pi) b = 0.0 p= 1e-5

Remarque

sin(2\*pi) = -2.4492935982947064e-16 (Windows 64bits)

**Version 1**

def testegal(x, y, p):

    '''

    x : flottant, nombre à tester à y

    y : flottant

    p : précision

    '''

    if abs(x - y) <= p:

        return True

    else:

        return False

r1 = testegal(math.sin(2\*math.pi), 0.0, 1e-15)

print(r1) # Renvoie **True**

r1 = testegal(math.sin(2\*math.pi), 0.0, 1e-16)

print(r1) # Renvoie **False**

**Version 2**

Ressource isclose() de la bibliothèque math: <https://www.w3schools.com/python/ref_math_isclose.asp>

def testegalv2(a, b):

    if math.isclose(a, b):

        return True

    else:

        return False

r2 = testegalv2(math.sin(2\*math.pi), 0.0)

print('r2 = ', r2) # Renvoie **False** (voir la ressource)

Modification : on introduit la précision souhaitée

math.isclose(a, b, abs\_tol = 10e-15)

r2 = testegalv2(math.sin(2\*math.pi), 0.0)

print('r2 = ', r2) # Renvoie **True**