*i* Python est un langage de programmation puissant et facile à apprendre.
Il dispose de structures de données de haut niveau et permet une approche simple mais efficace de la programmation orientée objet. Python permet l'écriture de scripts et le développement rapide d'applications dans de nombreux domaines et sur la plupart des plateformes.

*https ://docs.python.org/*

## A propos de Python : Recherche d'information

1. Qui a inventé le langage `Python` ? En quelle année ?

   .....

   .....

2. Donner la définition de langage interprété et de langage compilé.

   .....

   .....

   .....

   .....

   .....

   .....

   .....

   .....

   .....

   .....

3. Python est un langage interpreté ou compilé.

   .....

# La syntaxe

## L'affichage

*i* Faire le parcours Affichage sur eXoMorphisme
exomorphisme.fr - 1 NSI - Thème 6 : Langage et programmation - Les bases de Python -Parcours : Affichage

1. Quelle instructions permet d'afficher du texte à l'écran.

.....

2. En python, une chaine de caractères doit être encadrer par quels caractères ?

.....

3. L'instruction ci-dessous génére des erreurs. Apporter les corrections nécessaires.

```
1 print(Bonjour
```

.....

4. Comment coder un retour à la ligne dans une chaine de caractères en python ?

.....

## Le module Turtle

*i* Faire le parcours `Turtle` sur eXoMorphisme
exomorphisme.fr - 1 NSI - Thème 6 : Langage et programmation - Les bases de Python -Parcours : Turtle

1. De quel langage de programmation est inspiré le module `turtle`

.....

2. Quelle intruction est nécessaire en début de programme pour pouvoir utiliser le module `turtle`

.....

3. Quelle figure géométrique est dessiner par le code ci-dessous.

```
1  import turtle as tl
2
3  tl.forward(100)
4  tl.left(60)
5  tl.forward(100)
6  tl.left(60)
7  tl.forward(100)
8  tl.left(60)
9  tl.forward(100)
10 tl.left(60)
11 tl.forward(100)
12 tl.left(60)
13 tl.forward(100)
```

.....

.....

# Les variables

*i* En python une variable est un symbole (*nom*) qui renvoie a une valeur.
L'affectation de variable est l'action de nommer une valeur.
Elle se fait avec le symbole `=`

## Règle de nommage

**Obligatoire**

— Ne pas contenir un signe opératoire + - * / %/ ni le caractères # (*Il est réserver aux commentaires*)

— Doit commencer par une lettre

— Ne doit pas être un mot clé Python (`if`, `in`, `as` ;), ...)

**Préconisation :**

— le nom est entièrement écrit en minuscules et les mots sont séparés par des espaces soulignés (underscores _). Exemple : ma_variable

— le nom doit permettre de comprendre le rôle de la variable, éviter les caractères spéciaux et accentuer

## Affectation

Comme dis précédemment, l'affectation se fait à l'aide du symbole `=`.
Dans un premier temps l'interpréteur va interpréter ce qui se trouve à droite du signe `=` puis va l'affacter au nom qui se situe à gauche du signe `=`

## Typage

Il existe plusieurs type de variables. Parmi les plus courants, on retrouve :

| Variable | Type | Remarques |
|---|---|---|
| a = 2 | int : Les nombres entiers relatifs | |
| a = 3.5 | float : Les nombres réels | Le séparateur est le . et non la , |
| a = "Bonjour" | str : Les chaines de caractères | Entourer par des " ou des ' ou des triples """ ou ''' |
| a = True | bool : Les booléens True ou Fasle | |

On peut dans certain cas changer le type des variables avec les instructions `int`, `float` ou encore `str`
Par exemple, dans ce code a est défini comme une chaine de caractères puis transtyper en entier.

```
1 a = "12"
2 a = int(a)
```

### Exercice 1 Prédire

Déterminer la valeur de la variable `a` à la fin de ces instructions.

```
1 a = 13
2 b = 12
3 a = a + b + 2
```

.....

```
1 b = 12
2 a = b == 12
```

.....

# Les instructions conditionnelles

*i* Un instruction conditionnelle est composée d'un test puis d'un bloc d'instructions qui sera executé, ou non, en fonction de la validité du test.

## Syntaxe

En Python, le test commence par le mot clef **if** suivi d'une condition à valeur booléenne ( True ou False ) et se termine par le symbole **:**
Le bloc d'instructions qui suit s'exécute si et seulement si le test a pour valeur True

## 3 cas de figure

S'il n'y a qu'un seul cas à distinguer, on utilisera :

```
1 if condition:
2     bloc_d_instructions
```

S'il n'y a que deux cas à distinguer, on utilisera :

```
1 if condition:
2     bloc_d_instructions_1
3 else:
4     bloc_d_instructions_2
```

S'il y a plus de deux cas, on utilisera elif pour ajouter des conditions.

```
1 if condition_1:
2     instruction_1
3 elif condition_2:
4     instruction_2
5 elif condition_3:
6     instruction_3
```

## Opérateurs booléens

Il existe plusieurs type de variables. Parmi les plus courants, on retrouve :

| Test | Sens | Remarques |
|---|---|---|
| a == 2 | Test d'égalité | Le signe = étant réservé à l'affectation on utilise ici le double égal == |
| a > 3.5 | supérieur | Le séparateur est le . et non la , |
| a >= 3.5 | supérieur ou égal | |
| a < | inférieur | |
| a <= | inférieur ou égal | |
| "B" in "Bonjour" | Test d'appartenance | Nécéssite un itérable |

## Exercices

**Exercice 2** Quel résultat

Quel sera l'affichage du programme ci-dessous.

```
1 nb = 15
2 if nb % 2 == 0:
3     print("Le nombre est pair")
4 else :
5     print("Le nombre est impair")
6 print("Fin du programme)
```

. . . . . .

. . . . . .

## Exercice 3 Completer

Complète le code ci-dessous pour qu'il affiche `Fizz` si le nombre nb est un multiple de 3, `Buzz` si nb est un multiple de 5 et `FizzBuzz` si c'est un multiple de 5 et 3

```
nb = int(input("Entrer un nombre entier :"))

if ....................................
    print(".............................")
elif ....................................
    print(".............................")
elif ....................................
    ....................................
```

## Exercice 4 Débuguer

Chacun des ces programmes ci-dessous contient une erreur. Détermine cette erreur

```
nb1 = 13
nb2 = 14
if a = b:
    print("Les nombres nb1 et nb2 sont égaux")
else:
    print("Les nombres ne sont pas égaux")
```

.....

.....

.....

```
nb1 = 13
nb2 = 14
if a > b
    print("nb1 est supérieur à nb2")
```

.....

.....

.....

.....

# Les boucles itératives

*i* la boucle fait partie des grandes structures de base de l'algorithmique, et donc de la programmation.
Une itération est une séquence d'instructions destinée à être exécutée plusieurs fois.

## Répétition d'un bloc d'instruction

En Python, la répétition d'un bloc d'instructions se fait par `for var in range(nb_de_repetition):` Le bloc d'instructions à répéter est délimité par l'indentation.

**Exemple :**

L'exécution du code suivant

```
1 for i in range(4):
2     print("Hello")
```

Affichera :

```
Hello
Hello
Hello
Hello
```

*i* L'instruction `range(4)` crée un objet de type `range` qui contient les valeurs 0, 1, 2 et 3.

Quand on écrit l'instruction : `for i in range(4):`
A chaque répétition de la boucle, la valeur de la variable `i` change et prend successivement les valeurs de l'objet range

### Exercice 5 ★

Quelle sera l'affichage, après l'execution du code ci-dessous

```
1 for i in range(4):
2     print("Hello")
3 print("Word")
```

.....

.....

.....

.....

.....

.....

### Exercice 6 ★

Retrouver les erreurs dans les codes ci-dessous.

```
1 for i in range(4)
2     print("Bla")
```

.....

.....

.....

```
1 for i range(4):
2     print("Hello")
```

.....

.....

.....

```
1 for i in range(4):
2 print("Bonjour")
```

.....

.....

.....

### Exercice 7 ★★

Après éxécution du code ci-dessous, quelle sera la valeur de la variable `res`.

```
res = 1
for i in range(5):
    res += i*3
```

.....

.....

.....

### Exercice 8 ★

Ecrire un programme qui calcule et affiche la somme des nombres de 1 à 99.

.....

.....

.....

.....

.....

.....

## Itérer sur une liste ou une chaine de caractères

*i* Les objets itérables sont les objets que l'on peut parcourir à l'aide d'un `for`. Il possède généralement la méthode `__iter__`.

Les types `str, list, dict, tuple` sont les itérables que nous allons voir cette année.

**Exemples :**

Code

```
for mot in ["Hello","Word"]:
    print(mot)
```

Sortie

```
Hello
Word
```

Code

```
for lettre in "Hello":
    print(mot)
```

Sortie

```
H
e
l
l
o
```

### Exercice 9 ★★

Complete le programme ci-dessous pour qu'il affiche le nombre de caractères de la chaine ch.

```
ch = "Vive les NSI"
counter = ..........
for ..........
.....
.....
```

# Les boucles conditionnelles