



académie
Versailles

RÉGION ACADÉMIQUE
ÎLE-DE-FRANCE

MINISTÈRE
DE L'ÉDUCATION NATIONALE,
DE L'ENSEIGNEMENT SUPÉRIEUR
ET DE LA RECHERCHE



Égalité - Éducation - Excellence
République Française

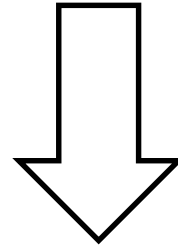
Sciences Numériques & Technologie (SNT)

Mise en
œuvre des
programmes
de SNT

Coder

Algorithme

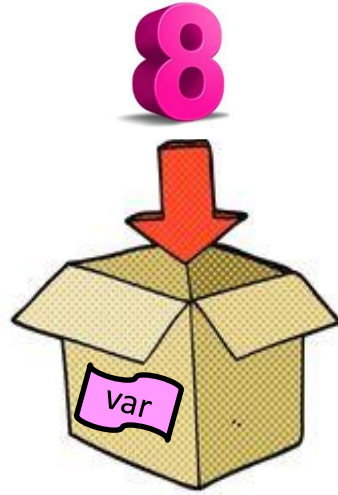
« Un **algorithme** est une suite finie et non ambiguë d'opérations ou d'instructions permettant de résoudre une classe de problèmes. »



C'est la recette de cuisine à donner à l'ordinateur pour qu'il mijote un bon petit plat.

Variables

var = 8



Valeur

Nombre entier

ex: 42

Nombre flottant

ex: 42.5

Caractères

ex: "coucou"

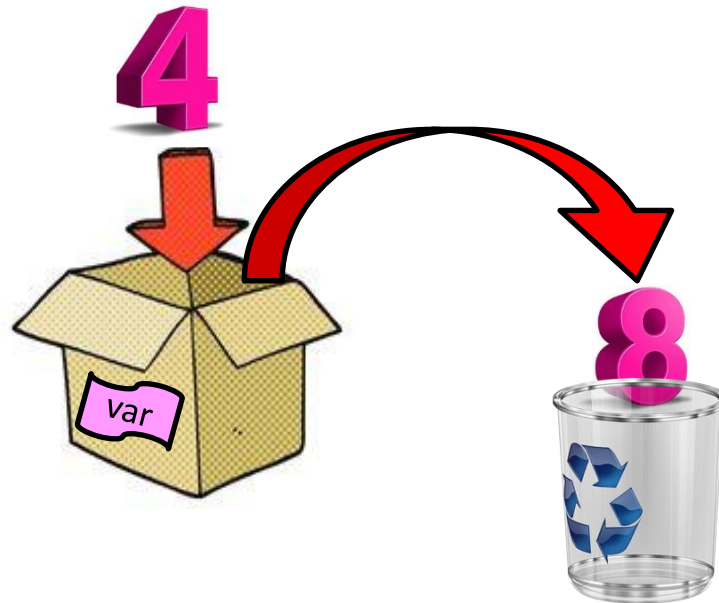
Liste

ex: [1, "a", 2.8]

...

Variable

var = 4



Conditions - IF

Python est capable de faire des comparaisons entre variables.

| | | | | | |
|------------|----|-------------------|----|-------------------|----|
| Egalité | == | Supérieur strict | > | Inférieur strict | < |
| Différence | != | Supérieur ou égal | >= | Inférieur ou égal | <= |

if Test :

 Bloc d'instructions

elif Test2 :

 Bloc d'instructions

else:

 Bloc d'instructions

Exemple:

```
if noteBac >= 10 :
```

```
    print("Tu as le bac!")
```

```
elif noteBac < 10 and noteBac >= 8 :
```

```
    print("Rattrapage...")
```

```
else:
```

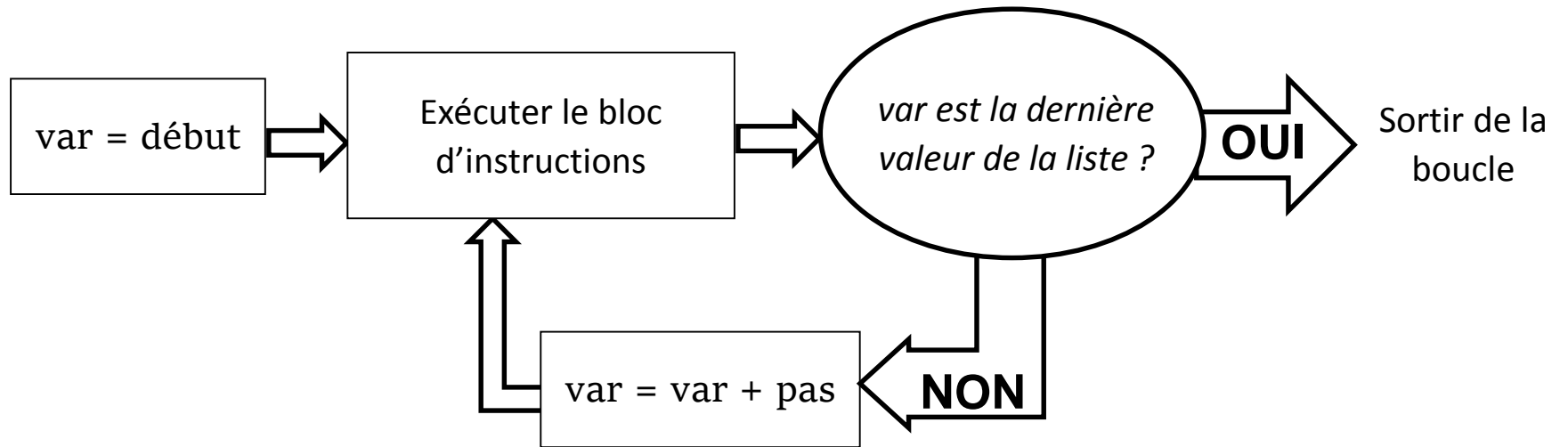
```
    print("Pas de bac cette année")
```

Note: Les indentations (touche Tab ↵) sont essentielles en langage python – elles définissent les différents blocs d'instruction.

Boucle - FOR

Python est capable de faire des tâches répétitives.

```
for var in range(début,fin,pas):  
    Bloc d'instructions
```



Boucle - FOR

Les paramètres sont:

- **début:** Entier d'où on commence à compter.
Par défaut, début = 0

Ex: `for i in range (10)` ⇔ `for i in range (0, 10, 1)`

- **fin:** On s'arrête de compter à fin **non-inclus**.

- **pas:** De combien en combien on compte.
Par défaut, pas = 1

Ex: `for i in range (2, 10)` ⇔ `for i in range (2, 10, 1)`

Exemple:

```
for i in range (5):
    print(i)
```



Affiche dans la console:

```
0
1
2
3
4
```

Fonctions

Une fonction est un bloc de code qui prend en argument un ensemble de variables et qui renvoie un résultat utilisable tel quel par un autre programme.

```
def nom_du_programme(arguments):
    Bloc de code
    return resultat
```

Exemple: Suite et série en mathématique

On définit les termes de la suite:

```
def u(n):
    terme = 2
    for i in range(1, n+1):
        terme = (1/3)*terme
    return terme
```

On calcule la série (somme des termes de la suite):

```
resultat = 0
for k in range(n+1):
    resultat = resultat + u(k)
print(resultat)
```

Bibliothèques

On peut charger sur python des bibliothèques de fonctions pré-écrites pour les utiliser dans les programmes.

On peut:

- Importer toute la bibliothèque
`import Bibliothèque`
- Importer toute la bibliothèque en lui donnant un autre nom
`import Bibliothèque as Nom`
- Importer une fonction particulière
`from Bibliothèque import Fonction`
- Importer une fonction particulière en lui donnant un autre nom
`import Bibliothèque.Fonction as Nom`

Logiciels

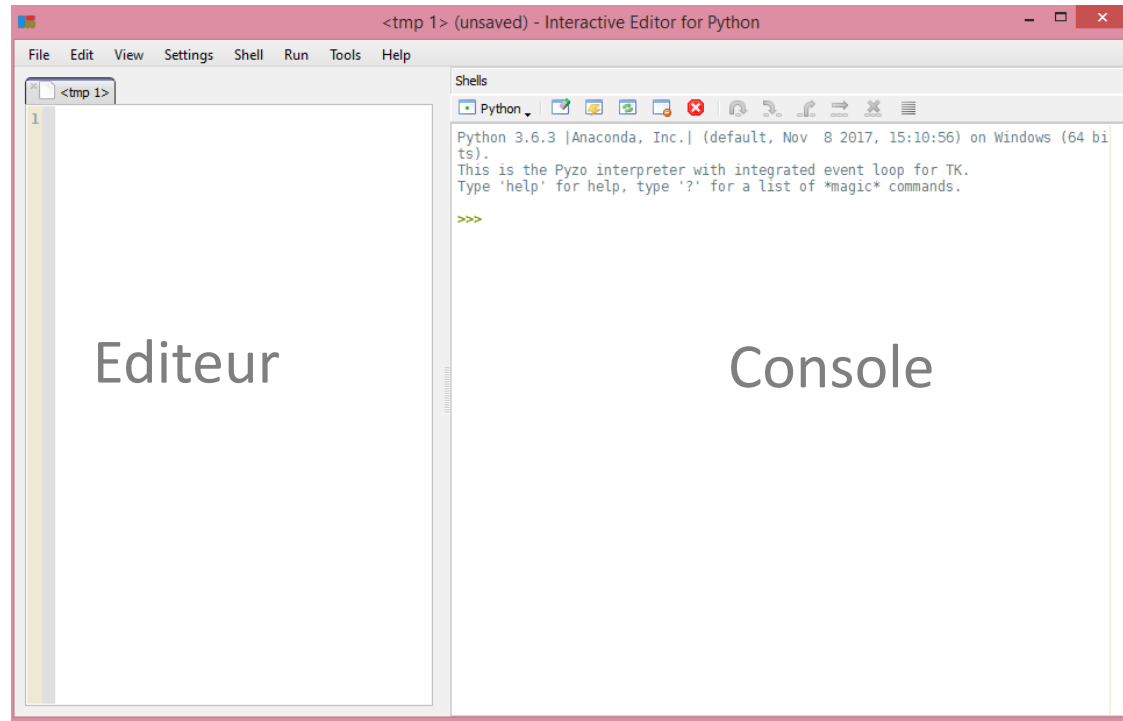
Edupython



Pyzo



Python



HTML

Notepad++



Thimble Mozilla



Messages d'erreur

Quand l'ordinateur n'est pas content avec votre code, il vous dit pourquoi et où!

Quelques exemples:

Print(a)

```
>>> (executing line 1 of "<tmp 1>")
Traceback (most recent call last):
  File "<tmp 1>", line 1, in <module>
    print(a)
NameError: name 'a' is not defined
```

Il faut définir la variable a en lui donnant une valeur

```
notebac = 5
if notebac < 10
    print("pas de bac")
```

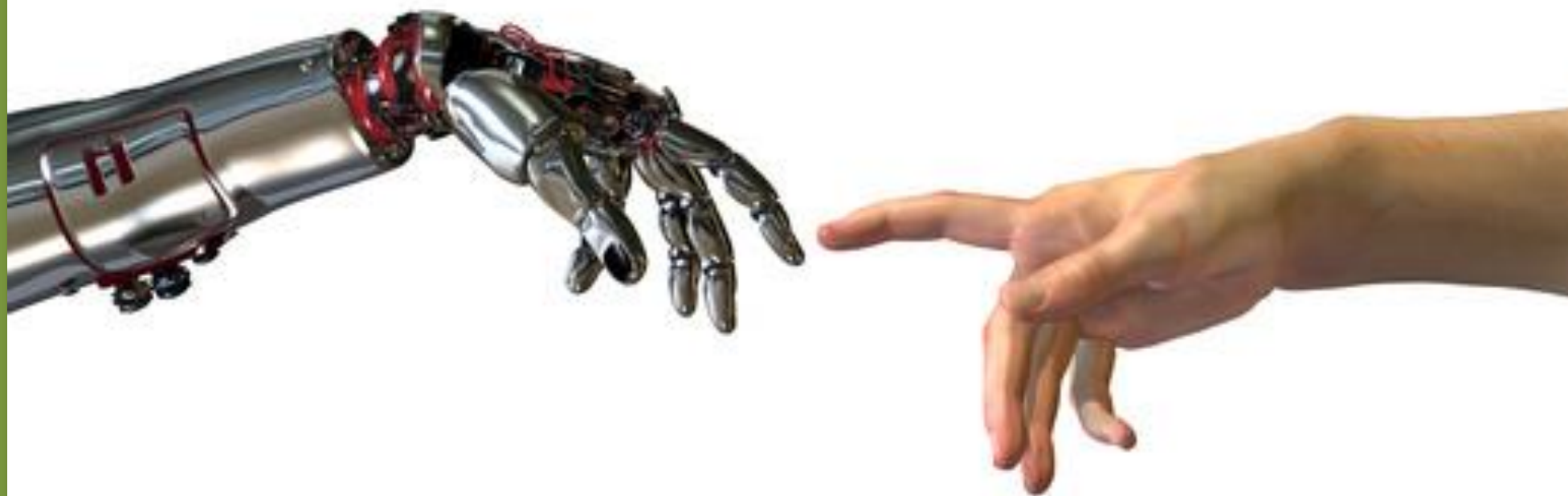
```
>>> (executing lines 1 to 3 of "<tmp 1>")
File "<tmp 1>", line 2
    if notebac < 10
        ^
SyntaxError: invalid syntax
```

On a oublié le : à la fin de la condition if

```
notebac = 5
if notebac == 15,5:
    print(« tu as le bac")
```

```
>>> (executing lines 1 to 3 of "<tmp 1>")
File "<tmp 1>", line 2
    if notebac == 3,5:
        ^
SyntaxError: invalid syntax
```

Un nombre flottant s'écrit avec un point, pas une virgule



Merci pour votre écoute

Temps Questions