# Introduction rapide à Python

Objectif :

- Découvrir les bases de Python nécessaires pour réaliser le TP1.

# Plan

1) Introduction
2) Premier programme
3) Variable
4) Interaction utilisateur
5) Affichage
6) Structure conditionnelle
7) Indentation

# Qu'est-ce que Python ?

- Un langage de programmation généraliste, interprété, et de haut niveau
- Conçu pour être simple à apprendre et facile à lire grâce à une syntaxe claire
- Utilisé dans de nombreux domaines :
  - Intelligence artificielle (très utilisé)
  - Analyse de données et Data Science
  - Développement d'applications desktop
  - Développement web (ex. Django, Flask)
  - Automatisation des tâches

- Créé en 1989 par Guido van Rossum, aux Pays-Bas
- Première version : publiée en 1991
- Dernière version: Version 3 (Version stable : Python 3.14 (oct. 2025))

- https://www.python.org/psf/

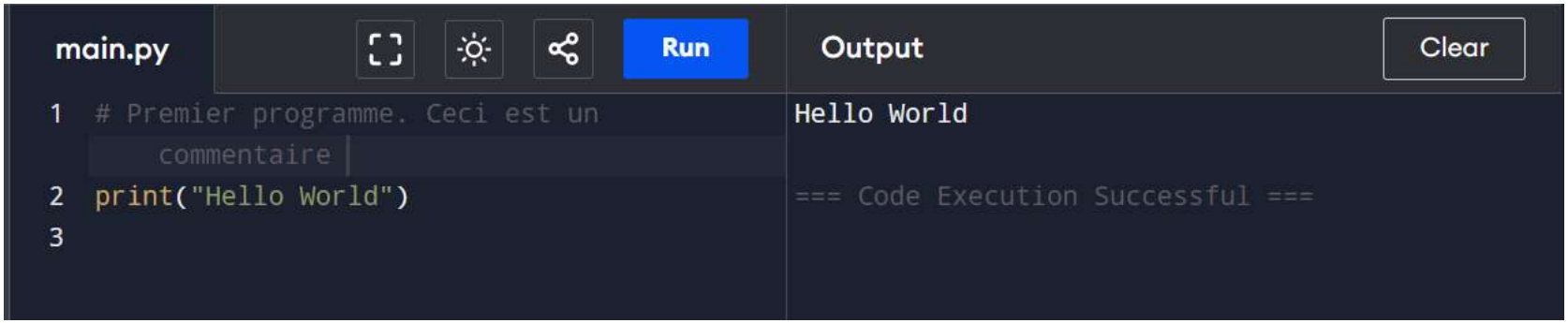
# Caractéristiques intéressantes

- Multiplateforme: fonctionne sur de nombreux systèmes d'exploitation : Windows, Mac OS, Linux, Android, iOS, etc
- Gratuit: Vous pouvez l'installer sur autant d'ordinateurs que vous voulez (même sur votre téléphone !)
- De haut niveau: Il demande relativement peu de connaissance sur le fonctionnement d'un ordinateur pour être utilisé
- Un langage interprété: Un script Python n'a pas besoin d'être compilé pour être exécuté, contrairement à des langages comme le C ou le C++
- Orienté objet
- Relativement simple à prendre en main
- Langage de programmation très utilisé (voir les classements TIOBE [1] et IEEE Spectrum [2])
- Grande communauté : Support et ressources abondantes pour résoudre des problèmes.

1. https://www.tiobe.com/tiobe-index/
2. https://spectrum.ieee.org/the-top-programming-languages-2023

# Premier programme

```
# Premier programme. Ceci est un commentaire
print("Hello World")
```




- Pas de main()
- Pas de compilation
- Exécution directe
- Pont virgule « ; » n'est pas nécessaire

- Interpréteur en ligne: https://www.programiz.com/python-programming/online-compiler/

# Variable

- La déclaration et l'initialisation se font en même temps
- Pas de déclaration de type
- Types dynamiques :
  - Le type d'une variable est déterminé automatiquement au moment de l'exécution
  - En C : typage statique

```
x = 10          # Variable entière
y = 3.14        # Variable flottante
z = "Python"    # Chaîne de caractères
```

- Le symbole = est l'opérateur d'affectation (== est l'opérateur de comparaison)
- Nommage
  - Règles habituelles
  - Python est sensible à la casse, ce qui signifie que les variables TesT, test et TEST sont différentes

# Interaction utilisateur

- La fonction **input()**
  - Permet de lire une entrée saisie par l'utilisateur au clavier
  - Retourne une chaîne de caractères (string)


- Si on veut un entier :

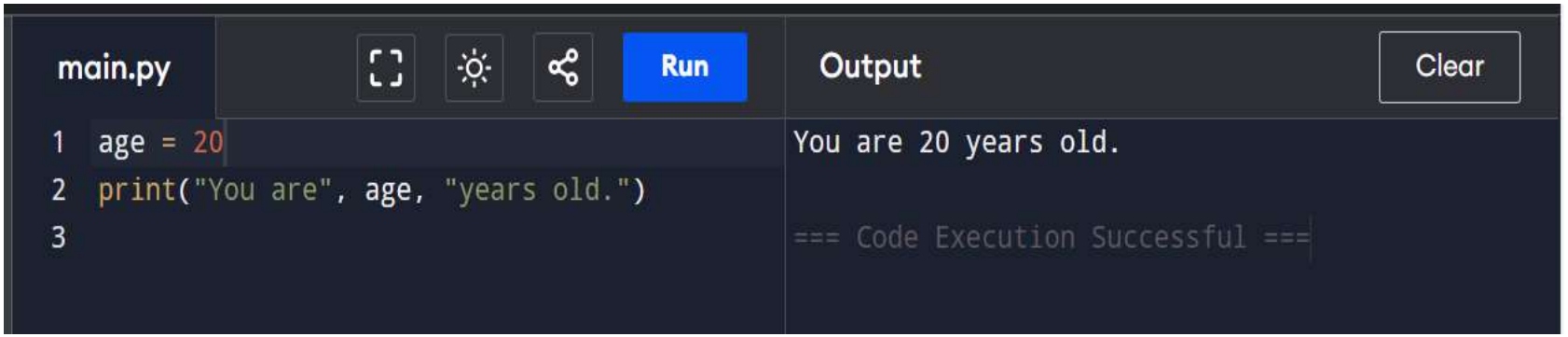
```
age = int(input("Enter your age: "))
```

# Affichage

- La fonction print()

```
print("Hello", name)
```

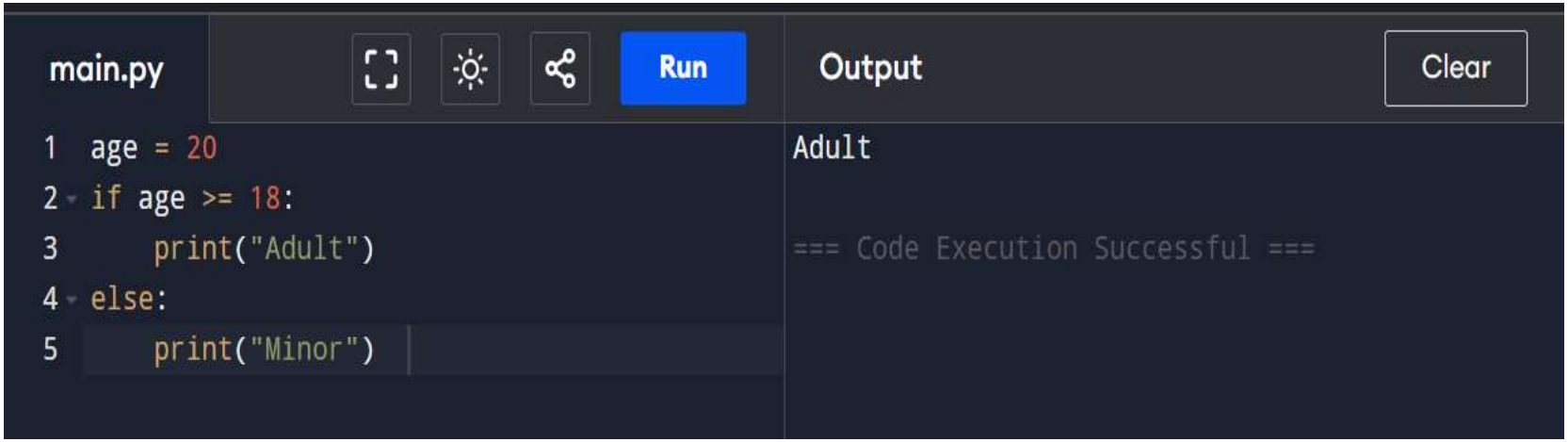
- On peut afficher plusieurs éléments :



# Structure conditionnelle

- Syntaxe

```
if condition:
    instruction
elif autre_condition:
    instruction
else:
    instruction
```



# bloc d’instructions et Indentation

- Indentation
  - Point essentiel
  - utilisé obligatoirement pour délimiter les blocs de code
  - Pas d’accolades

```python
# Python
age = 20
if age > 18:
    print("Adult")
    print("You can vote.")
print("End of the program.")
```

```c
//C
if (age > 18) {
    printf("Adult\n");
    printf("You can vote.\n");
}
printf("End of the program.\n");
```

# bloc d’instructions et Indentation (2)

- Erreur courante
  - Ne pas indenter
  - ou utiliser une mauvaise indentation génère une erreur

```
if True:
print("Erreur car pas d'indentation")
```

- Pratiquement,
  - l’indentation doit être homogène (soit des espaces, soit des tabulations, mais pas un mélange des deux).
  - Une indentation avec 4 espaces est le style d’indentation recommandé

- Une mauvaise indentation change le comportement du programme.