



PROJET ROBOT

1. Les Variables

Une variable est une manière de stocker une donnée. La variable doit avoir un type pour définir la donnée stockée.

Les types possibles sont:

- int => Chiffre ou nombre
- str => Caractère ou chaîne de caractères
- bool => Boolean True ("Vrai") ou False ("Faux")
- ...

Exemple:

```
un_bool = True
un_int = 42
une_str = "Coucou"

print("Ceci est un boolean:", un_bool)
print("Ceci est un integer:", un_int)
print("Ceci est une chaine de caracteres:", une_str)
```

2. Les fonctions

Les fonctions définissent des actions à faire lors de leur appel.
Une fonction est constituée de son nom, des paramètres pris en
entrée et de son type de retour.

Exemple:

```
# Cette fonction affiche "bonjour"
def afficher_bonjour():
    print("Bonjour")

# Cette fonction affiche le contenu de mot
def afficher_argument(mot="Salut"):
    print(mot)

# Cette fonction affiche "Coucou" et renvoi le même mot
def afficher_et_renvoyer_coucou() -> str:
    mot = "Coucou"
    print(mot)
    return mot

def main():
    afficher_bonjour()
    afficher_argument("Bonjour à tous, je suis un argument")
    afficher_argument()
    retour = afficher_et_renvoyer_coucou()
    print("retour =", retour)
```


3. Les conditions

Les conditions permettent de limiter l'exécution d'une action si et seulement si la condition est valide.

Exemple:

```
def positif_ou_negatif(nombre: int):  
    if nombre > 0:  
        print(nombre, "est positif !")  
    else:  
        print(nombre, "est negatif !")
```

4. Les Boucles

Les boucles sont des éléments qui permettent de répéter une action, un nombre de fois défini par la condition d'arrêt.

Exemple:

```
def dire_bonjour_a_tous_les_etudiants(nombre_etudiants: int):  
    indice = 1  
    while indice <= nombre_etudiants:  
        print("Bonjour etudiant numero", indice, "\n")  
        indice = indice + 1
```

5. Les Class

Les class sont des structures de code qui définissent un objet et les comportements qu'il peut réaliser.

Comme dans l'exemple suivant, vous pouvez importer une class "Robot" depuis la bibliothèque "Robot".

Ensuite, il vous faut créer un objet Robot, avec ici un nom et un âge. Par la suite, vous pouvez donner des ordres à votre robot, en faisant appel aux méthodes que la class Robot contient.

Example:

```
from pybot import Robot

def main():
    robot = Robot("Nono", 11)
    robot.dire_bonjour()
    robot.rendre_muet()
    robot.dire_bonjour()
    robot.donner_la_parole()
    robot.dire_bonjour()
```

6. La bibliothèque Pybot

Pybot est une bibliothèque avec une Class Robot que l'on peut utiliser pour faire un programme graphique.

Voici présenté ici les méthodes et des exemples d'utilisation:

**Pour démarrer nous avons besoin
d'importer la bibliothèque et de
démarrer le robot:**

```
from pybot import Robot
```

```
robot = Robot()
```

Pour allumer l'écran avec une longueur et une hauteur :

```
robot.allumer_ecran(800, 600)
```

Pour éteindre l'écran :

```
robot.eteindre_ecran()
```

Pour changer le titre :

```
robot.changer_titre("Exemple 1")
```


**Pour terminer le programme nous
avons besoin de lancer la boucle:**

```
robot.lancer_boucle()
```

Un exemple de programme complet :

```
from pybot import Robot

robot = Robot()

longueur = 800
hauteur = 600

robot.allumer_ecran(longueur, hauteur)

robot.changer_titre("Exemple 1")

robot.lancer_boucle()
```

Pour ajouter un bouton, avec le nom du bouton et la fonction exécuté au click de ce bouton :

```
robot.ajouter_bouton("Quitter", robot.eteindre_ecran)
```

Pour afficher la caméra :

```
robot.afficher_camera()
```

Pour éteindre la caméra :

```
robot.éteindre_camera()
```

Pour enregistrer une photo :

```
robot.enregistrer_photo()
```


Pour afficher une photo :

```
robot.afficher_photo()
```


Pour afficher un visage :

```
robot.afficher_visage_triste()  
robot.afficher_visage_content()  
robot.afficher_visage_colere()  
robot.afficher_visage_fier()
```