

Cours 11:

Quiz 3 et suite des fonctions

Présentation par Tommy Gagnon Joyal



Table des matières

01

Rappel et Quiz 3

Prise des présences via le Quiz

02

Rappel des fonctions

Explication sur les fonctions et exemples.

03

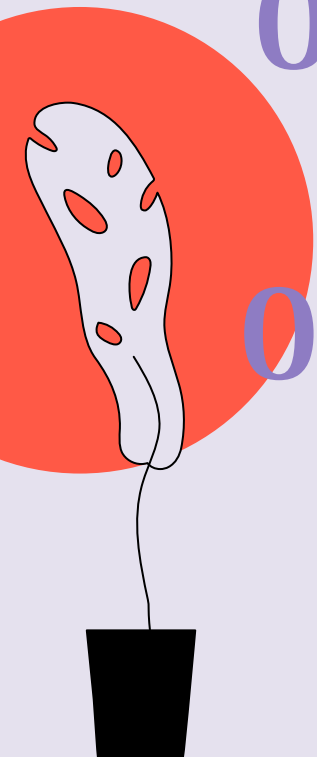
Pause

Café 🥤 😊

04

Exercices en classe

Correction des exercices sur les fonctions du cours précédent





01

Rappel



Prochaines dates

- **Quiz #3: cours #11 (Aujourd'hui)**
- Quiz #4: cours #13
- Quiz Bonus: cours #15



Prochaines dates

- TP2: À donner au cours#12(prochain cours)

Si vous avez des idées/recommandations pour le TP2, n'hésitez pas venir m'en parler.



Quiz time!



Les fonctions

- Une fonction est un bloc de code qui s'exécute uniquement lorsqu'elle est appelée.
- Vous pouvez transmettre des données, appelées paramètres, dans une fonction.
- Une fonction peut renvoyer des données en tant que résultat.

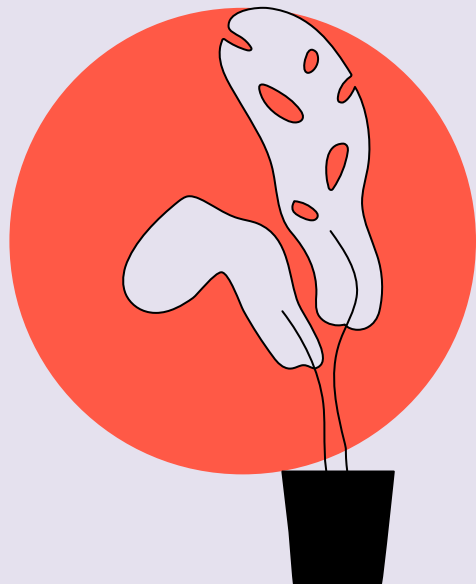


Les fonctions

Création d'une fonction:

- En Python, une fonction est définie en utilisant le mot-clé def :

```
def my_function():  
    print("Hello from a function")
```



Les fonctions

Appeler une fonction:

- Pour appeler une fonction, utilisez le nom de la fonction suivi de parenthèses :

```
def my_function():  
    print("Hello from a function")  
  
my_function()
```



Les fonctions

Arguments:

- Des informations peuvent être transmises aux fonctions sous forme d'arguments.
- Les arguments sont spécifiés après le nom de la fonction, à l'intérieur des parenthèses. Vous pouvez ajouter autant d'arguments que vous le souhaitez, il suffit de les séparer par une virgule.



Les fonctions

Arguments:

- L'exemple suivant comporte une fonction avec un argument (fname). Lorsque la fonction est appelée, nous transmettons un prénom, qui est utilisé à l'intérieur de la fonction pour afficher le nom complet :

```
def my_function(fname):  
    print(fname + " Refsnes")  
  
my_function("Emil")  
my_function("Tobias")  
my_function("Linus")
```



Les fonctions

Valeur par défaut des paramètres:

- L'exemple suivant montre comment utiliser une valeur par défaut pour un paramètre.
- Si nous appelons la fonction sans argument, elle utilise la valeur par défaut :

```
def my_function(country = "Norway"):  
    print("I am from " + country)
```

```
my_function("Sweden")  
my_function("India")  
my_function()  
my_function("Brazil")
```

Les fonctions

Valeurs de retour:

- Pour permettre à une fonction de renvoyer une valeur, utilisez l'instruction `return` :

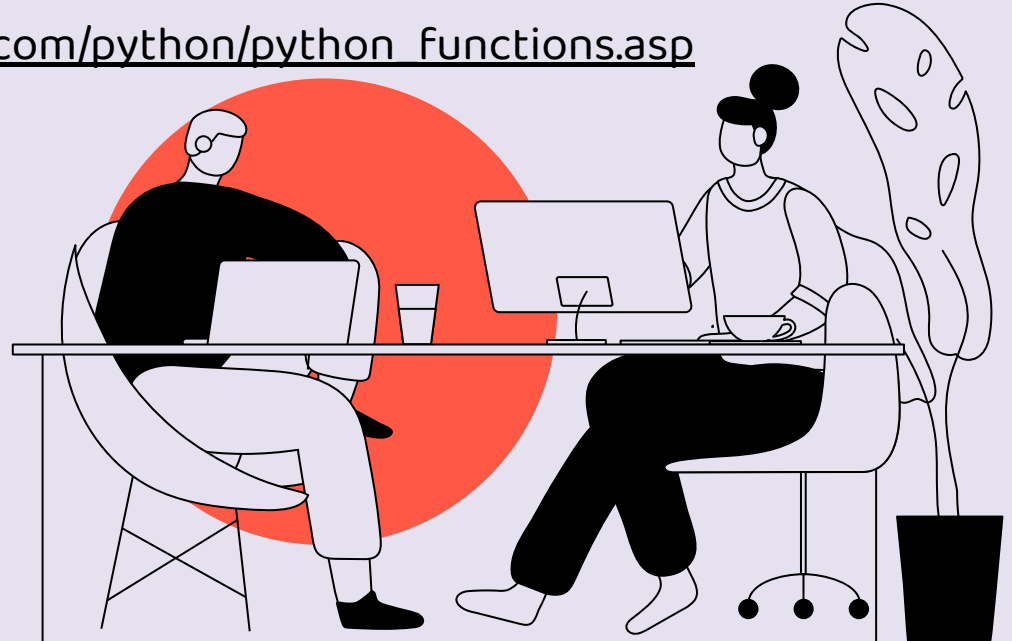
```
def my_function(x):  
    return 5 * x  
  
print(my_function(3))  
print(my_function(5))  
print(my_function(9))
```



Les fonctions

Et plus encore !

- https://www.w3schools.com/python/python_functions.asp



Resources

Langage:

- Interprétée (Comme Python): Exécuté ligne par ligne.
- Compilée: Le code est complètement transformé en Assembleur pour être exécuté.

Programmation:

- Procédurale: Une suite d'instructions (souvent réunies en fonctions) exécutées par une machine.

- Orientée objet (POO) (Comme Python): Chaque programme est constitué d'entités appelées objets, qui ne sont pas facilement accessibles et modifiables.

Type de variable:

- Variable: Symboles qui associent un nom (l'identifiant) à une valeur.
 - CONSTANTE: Une variable qui ne devrait pas changer pendant l'exécution d'un programme.
-

Resources

- Variable Booléenne: Relatif à l'algèbre de Boole. Qui ne peut prendre que deux valeurs distinctes.
- Commentaire: Permet de documenter du code dans un fichier. Celui-ci ne sera pas lu par le langage en question.
- Faiblement typé: Se préoccupe peu des types. Cela permet de chaîner des valeurs de différents types.
- Typé dynamiquement (Comme Python): Il va définir lui-même le type selon la nature de la valeur .
- Orientée objet (POO) (Comme Python): Chaque programme est constitué d'entités appelées objets, qui ne sont pas facilement accessibles et modifiables.

Typage:

- Fortement typé : est un langage dans lequel les types utilisés dans le code source (fonction, variable, etc.) sont vérifiés au moment de la compilation.
-

Resources

Énoncés conditionnels:

- L'énoncé 'if' et ses variantes: Exécute un bloc de code si une condition évalue à true.

- La boucle "while" est également une structure de répétition. Elle permet l'exécution répétée des actions définies à l'intérieur, en utilisant l'indentation pour délimiter le bloc de code à répéter.

Énoncés conditionnels:

- La boucle "for" est une structure de répétition. Elle permet l'exécution répétée des actions définies à l'intérieur, en utilisant l'indentation pour délimiter le bloc de code à répéter.
-