

A la découverte des objets

Université Virtuelle de Côte d'Ivoire



Table des matières

Objectifs	3
Introduction	4
I - L'Orientée Objet	5
1. La programmation Orientée Objet (POO)	5
2. Description d'un Objet	5
3. La conception de Programme Orientée Objet	6
4. Les Objets en Python	6
5. Exercice : Question1	7
6. Exercice : Question2	8
7. Exercice : Question3	8
8. Exercice : Question4	8
II - Manipulation d'objet : cas des chaînes de caractères	9
1. Appliquer une méthode à un objet	9
2. Les méthodes des chaînes de caractère	10
3. Actions multiples sur un objet	10
4. Exercice : Question1	11
5. Exercice : Question2	11
6. Exercice : Question3	12
7. Exercice : Question4	12
Conclusion	13
Solutions des exercices	14

Objectifs



- Illustrer le concept objet en général et en python
- Décrire le concept objet avec les chaînes de caractère
- Écrire des scripts python pour manipuler des objets

Introduction



Un objet est une structure de données, c'est un conteneur à l'intérieur duquel on peut définir des variables et des fonctions.

En python, tout est objet ; les variables sont des objets, les fonctions sont des objets, etc.

Ce cours permettra à l'apprenant de comprendre la notion d'objet en général et de s'initier à la programmation objet avec le langage Python.

L'Orientée Objet



Objectifs

A la fin de cette activité, l'apprenant sera capable de :

- Illustrer le concept Objet en général et en Python en particulier
- Décrire le schéma d'un objet
- Catégoriser les composantes d'un objet

Les objets constituent le cœur de la programmation en langage Python. En python tout est objet. Dans cette partie de la leçon seront abordées les notions des principes d'un objet en général et une corrélation sera faite avec l'usage des objets avec le langage Python

1. La programmation Orientée Objet (POO)

Définition : POO

La Programmation Orientée Objet est un modèle de langage de programmation qui s'articule autour de la manipulation des objets. Par le passé, un programme était une procédure logique qui récupérait des données en entrée, les traitait puis produisait des données en sortie. Ainsi le codeur se focalisait sur la logique, le fonctionnement de l'algorithme. Avec la POO, le codeur se focalise sur la manipulation des objets, allant de leur définition jusqu'à leur utilisation.

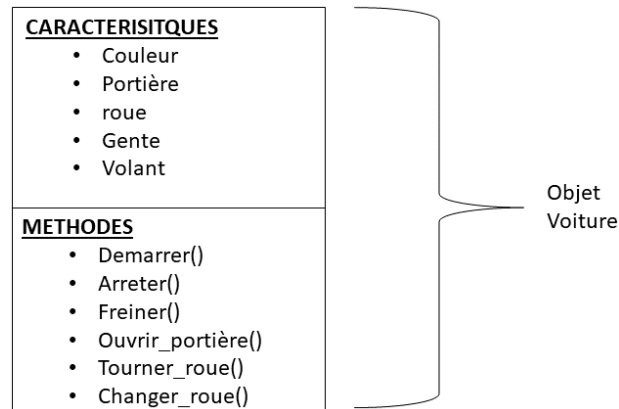
C'est quoi un Objet ?

Un objet est une sorte de variable spéciale qui est constitué de données et qui permet la manipulation de ces données.

En POO, un objet est créé à partir d'une classe (les classes seront l'objet d'apprentissage de la leçon 5).

2. Description d'un Objet

Un objet est constitué d'*attributs* qui sont des variables qui le décrivent (spécifique à lui) et de *méthodes* qui sont des fonctions permettant de manipuler les attributs de l'objet.



description d'un objet de type voiture



Exemple : Manipulation de voiture

Considérons un programme qui permet de rouler une voiture. En analysant, nous constatons que l'objectif du programme est de manipuler un objet qui est une voiture.

La voiture a ses caractéristiques qui sont : sa couleur, le nombre de pneu, un volant, le nombre de portières, les pédales, etc. Ces caractéristiques sont appelées *attributs*

Sur cette voiture peuvent être appliquée un ensemble d'actions qui sont : démarrer, freiner, accélérer, tourner, etc. Ces actions sont appelées *méthodes*



Remarque

les attributs et les méthodes d'un objet sont spécifiques à l'objet, ils ne fonctionnent pas s'ils sont appelés sur un autre objet

3. La conception de Programme Orientée Objet

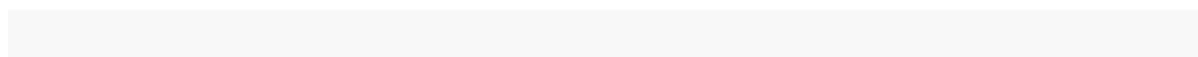
La POO introduit une nouvelle manière de concevoir un programme, Il s'agira maintenant de répondre aux questions suivantes :

- Quel est l'objectif du programme ?
- Quels sont les objets qui sont manipulés dans le programme ?
- Quelles sont les caractéristiques de ces objets ?
- Quelles sont les actions à appliquer à ces objets ?
- Quelles sont les interactions entre les différents objets du programmes ?

Les réponses à ces questions passent souvent par l'analyse et le design Orientée Objet comme par exemple UML

4. Les Objets en Python

En python tout est objet, même les variables sont des objets. l'exemple suivant montre ce qui se passe lors de la manipulation d'une variable



```

1 age=24 #Python crée l'objet <24> puis crée une référence sur l'objet <24>
  nommée [age]
2 a=age #Python crée encore une autre référence nommée [a] sur le même objet
  <24>
3
4 nom="Benjamin" #Python crée l'objet <"Benjamin"> puis crée une référence
  nommée [nom] sur l'objet <"Benjamin">.
5 nom="Albert" #Python crée l'objet <"Albert"> puis re-affecte la référence
  [nom] à l'objet <"Albert">.

```

En somme, le nom de la variable n'est juste qu'une référence sur l'objet qui lui est affecté. Grâce à la référence, il est possible de manipuler l'objet durant le temps de l'exécution d'un programme.

Cycle de vie d'un objet en Python

Un objet en Python *est en vie* (utilisable) tant qu'il possède au moins une référence. Si l'objet ne possède *pas ou plus* de référence alors il est supprimé automatiquement par un gestionnaire de la mémoire appelé *garbage collector* (Collecteur d'ordure en Français). Cela permet une meilleure gestion de la mémoire de l'appareil utilisé pour réaliser le script Python.

Ainsi dans le code précédent, l'objet <"Benjamin"> sera supprimé par la garbage collector puisqu'il ne possède plus de référence.



Exemple

```

1 a=5
2 b=a
3 a=7
4 b=8

```

Jusqu'à la ligne 3, l'objet <5> est toujours en vie car il a toujours pour référence [b].

A la ligne 4, l'objet <5> est supprimé par le *garbage collector* car il ne possède plus de référence.



Remarque

Le programmeur a la possibilité de déclencher la suppression d'un objet en supprimant toutes les références de l'objet grâce à l'instruction python suivante :

```

1 del nom_reference #permet de supprimer la référence [nom_reference]

```

5. Exercice : Question1

[solution n°1 p.14]

Que va retourner cette instruction ?

```

1 genre = "M"
2 type(genre)

```

- ☐ <class, str>
- ☐ <class 'str'>
- ☐ <class 'char'>

6. Exercice : Question2

[solution n°2 p.14]

Écrivez la commande qui permet de supprimer une référence sur un objet

7. Exercice : Question3

[solution n°3 p.14]

Quel est le résultat d'exécution de ce programme ?

```
1 a = 17
2 b = a
3 del a
4 print(a)
```

- ☐ aucun resultat
- ☐ message d'erreur
- ☐ 17

8. Exercice : Question4

[solution n°4 p.14]

Classer les éléments suivant selon leur appartenance. Considérons un objet de type Personne

danser manger regarder marcher poid taille age s'asseoir rire nom

Attributs	Méthodes

Manipulation d'objet : cas des chaînes de caractères



Objectifs

A la fin de cette activité, l'apprenant sera capable de :

- Écrire des scripts python pour manipuler l'objet chaîne de caractère
- Utiliser les méthodes prédéfinies pour modifier un objet de type chaîne de caractère.

En python tout est un *objet*, les objets sont *créés à partir des classes*. L'image suivante montre en vérifiant les types des variables `a` et `b` qu'elles sont toutes deux issues respectivement de *classes de type entier* `<class 'int'>` et de *type chaîne de caractère* `<class 'str'>`.

Les chaînes de caractères étant une classe, il est possible d'appliquer des actions sur l'objet qui est créé à partir de la *classe str*. dans cette partie, nous verrons comment se fait la manipulation d'objet en prenant pour exemple les chaînes de caractère.

```
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:59:51) [MSC v.1914 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> a = 24
>>> type(a)
<class 'int'>
>>> car = "ma_chaine"
>>> type(car)
<class 'str'>
```

console Python 3.7.0

1. Appliquer une méthode à un objet

Exécuter une méthode d'un objet consiste à réaliser une opération de lecture ou de modification sur cet objet. La syntaxe qui permet de réaliser cette opération est :

```
1 objet.methode()
2 [reference_sur_l'objet].methode()
```

A gauche, nous avons l'objet sur lequel l'action doit être appliquée, cela se fait en utilisant directement l'objet ou une référence sur l'objet. Ensuite un "." puis le nom de la méthode suivi de parenthèse ouvrante et fermante "()"



Exemple : Mettre une chaîne de caractère en majuscule

```
1 nom = "kouame"
2 nom.upper()
3 "kouadio".upper()
```

Ce code affichera comme résultat :

'KOUAME'

'KOUADIO'

2. Les méthodes des chaînes de caractère

Normalement, pour effectuer une action sur un objet, le développeur doit au préalable définir le contenu de la méthode lui-même au sein de la classe à partir de laquelle l'objet sera créé.

Pour les objets de la classe *str*, les programmeurs de Python ont déjà programmé des méthodes de base qui permettent de les manipuler. Le code Python suivant fait illustration de certaines de ces méthodes

```
>>> minuscules = "une chaîne en minuscules"
>>> minuscules.upper() # Mettre en majuscules
'UNE CHAÎNE EN MINUSCULES'
>>> minuscules.capitalize() # La première lettre en majuscule
'Une chaîne en minuscules'
>>> espaces = "    une chaîne avec des espaces    "
>>> espaces.strip() # On retire les espaces au début et à la fin de la chaîne
'une chaîne avec des espaces'
>>> titre="introduction"
>>> titre.upper().center(20)
'      INTRODUCTION      '
```

des méthodes appliquées aux objets de la classe str



Remarque

Les méthodes appliquées aux chaînes de caractères ne modifient pas directement l'objet d'origine, elles créent plutôt un nouvel objet. Ainsi minuscules.upper() ne modifie pas le contenu de l'objet minuscules mais renvoie plutôt une autre chaîne de caractère.

La dernière instruction du code précédent montre qu'il est possible de faire appel à deux méthodes sur un même objet. la suite de ce cours explique comment cela est possible

3. Actions multiples sur un objet

Valeur de retour d'une méthode

En POO en général, il est possible sur une seule ligne de code d'appliquer plusieurs actions à un objet.

Comment cela se fait-il ?

Réaliser une telle opération dépend de la valeur que retourne la 1^{ère} méthode appliquée. Commentons la dernière instruction du code précédent

```
1 titre.upper().center(20)
```

La 1^{ère} opération réalisée sur l'objet titre est faite par la méthode upper(), cette méthode renvoie une chaîne de caractère en majuscules ("INTRODUCTION"). La chaîne de caractère renvoyée étant aussi un objet, il est alors possible de lui appliquer des modifications d'où l'application de la méthode center().

Si nous décomposons notre instruction, nous avons par ordre d'exécution :

```
1 titre.upper() # renvoie la valeur 'INTRODUCTION' qui est de type <class
'>
2 'INTRODUCTION'.center(20) #renvoie la valeur '      INTRODUCTION      '
```

Que fait la méthode center(20) ?

La méthode center() permet de centrer une chaîne de caractère. On lui passe en paramètre la taille de la chaîne que l'on souhaite obtenir. La méthode va ajouter alternativement un espace au début et à la fin de la chaîne, jusqu'à obtenir la longueur demandée. Dans cet exemple, *l'objet titre* contient la chaîne 'introduction', chaîne qui (en minuscules ou en majuscules) mesure 12 caractères. On demande à center de centrer cette chaîne dans un espace de 20 caractères. La méthode center va donc placer 4 espaces avant le titre et 4 espaces après, pour faire 20 caractères en tout.

4. Exercice : Question1

[solution n°5 p.15]

Laquelle de ces instructions permet de déclencher une action sur un objet ?

- ☐ methode(objet)
- ☐ objet(methode)
- ☐ objet_methode()
- ☐ objet.methode()

5. Exercice : Question2

[solution n°6 p.15]

Écrire le code Python qui permet de déclarer une chaîne de caractère et de mettre tous les caractères de cette chaîne en minuscules en utilisant la méthode lower()

lieu_naissance = "Abidjan"

```
_____.
```

Écrire le code python qui permet d'afficher la chaîne suivante "Bonjour les amis !"

```
sal = "bonjour les amis !"
```

[solution n°8 p.15]

Écrivez le nom de la méthode qui permet de centrer l'objet <chaine> sur un total de 19 caractères.

Conclusion



Ce cours d'introduction à la POO a permis à l'apprenant de découvrir le monde de la conception et de la programmation Objet en général et en particulier avec le langage Python. La manipulation des chaînes de caractères qui sont aussi des objets en Python a permis à l'apprenant de s'initier à la programmation objet.

La prochaine leçon portera sur le formatage des chaînes de caractère, les listes et les tuples.

Solutions des exercices



> Solution n°1

Exercice p. 7

Que va retourner cette instruction ?

```
1 genre = "M"
2 type(genre)
```

- ☐ <class, str>
- ☒ <class 'str'>
- ☐ <class 'char'>

> Solution n°2

Exercice p. 8

Écrivez la commande qui permet de supprimer une référence sur un objet
del

> Solution n°3

Exercice p. 8

Quel est le résultat d'exécution de ce programme ?

```
1 a = 17
2 b = a
3 del a
4 print(a)
```

- ☐ aucun resultat
- ☒ message d'erreur
- ☐ 17

> Solution n°4

Exercice p. 8

Classer les éléments suivant selon leur appartenance. Considérons un objet de type Personne

--	--



Attributs	Méthodes
taille	marcher
poid	danser
age	rire
nom	s'asseoir
	manger
	regarder

> **Solution n°5**

Exercice p. 11

Laquelle de ces instructions permet de déclencher une action sur un objet ?

- ☐ methode(objet)
- ☐ objet(methode)
- ☐ objet_methode()
- ☒ objet.methode()

> **Solution n°6**

Exercice p. 11

Écrire le code Python qui permet de déclarer une chaîne de caractère et de mettre tous les caractères de cette chaîne en minuscules en utilisant la méthode lower()

```
lieu_naissance = "Abidjan"
lieu_naissance.lower()
```

> **Solution n°7**

Exercice p. 12

Écrire le code python qui permet d'afficher la chaîne suivante "Bonjour les amis !"

```
sal = "bonjour les amis !"
sal.capitalize()
```

> **Solution n°8**

Exercice p. 12

Écrivez le nom de la méthode qui permet de centrer l'objet <chaine> sur un total de 19 caractères.

`center(19)`