

Python

- Python
 - Livrable attendu
 - Spécifications générales
 - Spécifications détaillées
 - 1. Créer une classe Python `Persona`
 - 2. Dans la classe `Persona`, créer la méthode magique `__str__`
 - 3. Dans la classe `Persona`, créer la méthode `set_address()`
 - 4. Dans la classe `Persona`, créer la méthode `show_address`
 - 5. En utilisant les données fournies ci-dessous :

Persona :

En UX (étude de l'expérience utilisateur), un persona est un personnage fictif représentant un archétype utilisateur d'un service (site web, application mobile, etc.).

Les personas peuvent être très utiles pour créer des utilisateurs factices afin de tester une application/un service.

Livrable attendu

- Un dossier projet contenant un fichier `persona.py` :
 - Déclarant la classe `Persona`
 - Implémentant un bloc d'instruction qui teste la classe en instanciant un objet, en définissant son adresse et en affichant sa représentation humaine et son adresse (vous pouvez utiliser n'importe quelles données factices)
 - Implémentant un bloc d'instructions qui produit le rendu illustré au point n°5

Spécifications générales

Dans notre projet, un persona sera défini par les attributs protégés suivants :

- `first_name`
- `last_name`
- `address_street`
- `address_number`
- `city`
- `postcode`

Spécifications détaillées

1. Créer une classe Python `Persona`

- Son constructeur (méthode `__init__`) est appelé avec les attributs `first_name` et `last_name` (pas de valeur par défaut)
- Pour les attributs restants, le constructeur instancie les valeurs suivantes par défaut :

```
address_street # sera une chaîne de caractères vide
address_number # sera un integer égal à 0
city # sera une chaîne de caractères vide
postcode # sera une chaîne de caractères vide ('')
```

2. Dans la classe `Persona`, créer la méthode magique `__str__`

- Elle affiche le message suivant : `Hi ! I'm first_name last_name` (où `first_name` et `last_name` sont évidemment remplacés par la valeur de ses attributs)
- Exemple : `Hi ! I'm Emilien Duval`

3. Dans la classe `Persona`, créer la méthode `set_address()`

- Elle prend en paramètre `address_street`, `address_number`, `city` et `postcode` (pas de valeur par défaut)
- Elle affecte les valeurs passées en paramètres aux attributs correspondants de la classe

4. Dans la classe `Persona`, créer la méthode `show_address`

- Elle ne prend aucun paramètre en entrée
- Elle affiche `My full address is : address_number address_street, city (postcode)` (où `address_number`, `address_street`, `city` et `postcode` sont évidemment remplacés par la valeur de ses attributs)
- Exemple : `My full address is : 45 avenue du Général de Gaulle, Bordeaux (33000)`

5. En utilisant les données fournies ci-dessous :

```
[{
  "first_name": "Ilyès",
  "last_name": "Fleury",
  "address_street": "Rue Paul Bert",
  "address_number": 73,
  "city": "Dunkerque",
  "postcode": 12681
},{
  "first_name": "Lia",
  "last_name": "Dumont",
  "address_street": "Rue Louis-Blanqui",
  "address_number": 30,
  "city": "Lille",
  "postcode": 63996
},{
  "first_name": "Eléonore",
  "last_name": "Caron",
  "address_street": "Avenue du Château",
  "address_number": 87,
  "city": "Rennes",
  "postcode": 78482
},{
  "first_name": "Eva",
  "last_name": "Girard",
  "address_street": "Rue du Bon-Pasteur",
  "address_number": 9,
  "city": "Rueil-Malmaison",
  "postcode": 23879
}]
```

Exploiter ces données pour implémenter dans une boucle les instructions suivantes :

- Instancier le nombre correspondant d'objets de type `Persona`
- Attribuer leur adresse complète pour chaque objet
- Afficher pour chaque objet :
 - Sa représentation humaine (méthode `__str__`)
 - Son adresse

Sortie terminal attendue :

```
Hi ! I'm Ilyès Fleury
My full address is : 73 Rue Paul Bert, Dunkerque (12681)
--
Hi ! I'm Lia Dumont
My full address is : 30 Rue Louis-Blanqui, Lille (63996)
--
Hi ! I'm Eléonore Caron
My full address is : 87 Avenue du Château, Rennes (78482)
--
Hi ! I'm Eva Girard
My full address is : 9 Rue du Bon-Pasteur, Rueil-Malmaison (23879)
--
```

Bon courage !
Geoffroy Ladrat - 2021