

Créez GrandPy Bot, le papy-robot



Github : https://github.com/Zepmanbc/oc_dapthon_pr7

Trello : <https://trello.com/b/Wh1dkH9n/ocdapythonpr7>

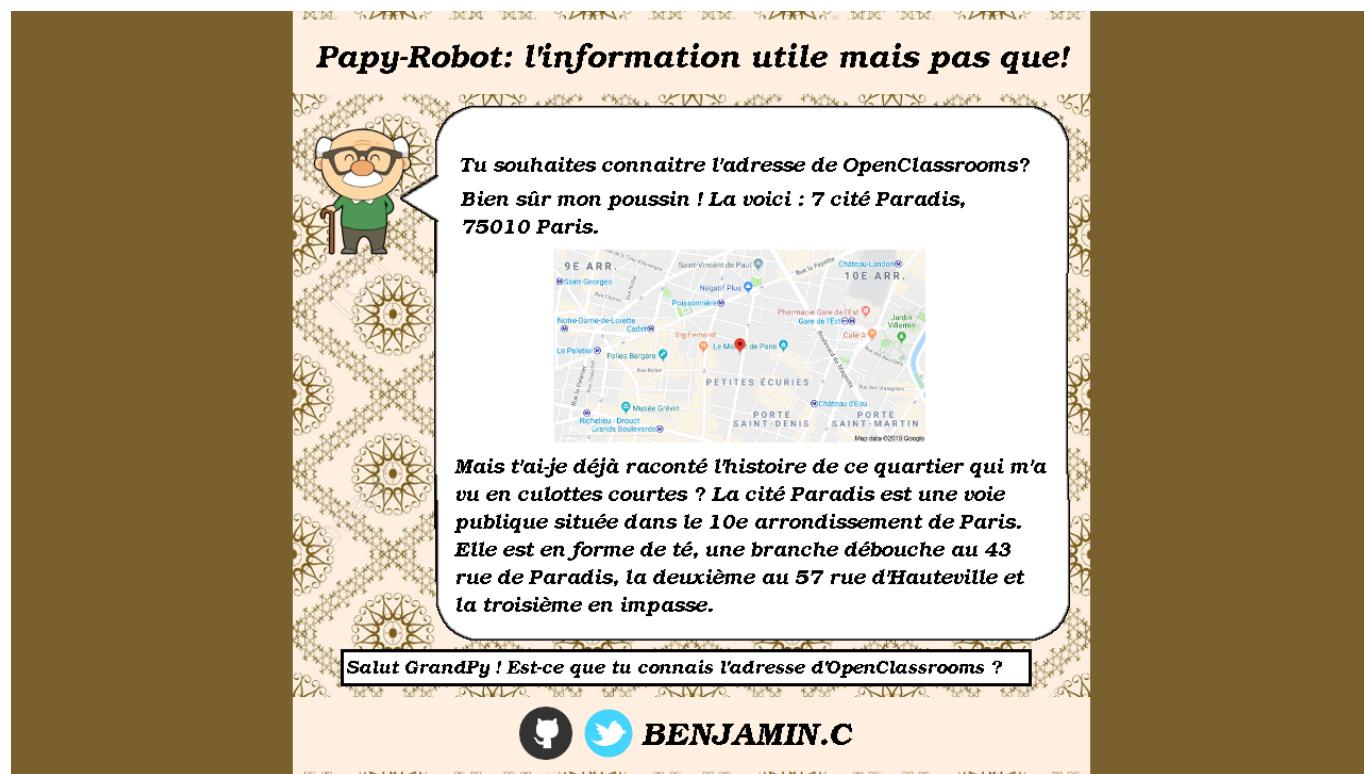
Lien Heroku : <https://bc-ocdapythonpr7.herokuapp.com/>

Analyse Fonctionnelle

Démarche

Front

Réalisation d'une maquette pour la version écran d'ordinateur et de smartphones.





PapyRobot: l'information utile mais pas que !

Salut GrandPy ! Est-ce que tu connais l'adresse d'OpenClassrooms ?

Tu souhaitez connaître l'adresse de OpenClassrooms ?

Bien sûr mon poussin ! La voici : 7 cité Paradis, 75010 Paris.



Mais t'ai-je déjà raconté l'histoire de ce quartier qui m'a vu en culottes courtes ? La cité Paradis est une voie publique située dans le 10e arrondissement de Paris. Elle est en forme de té, une branche débouche



BENJAMIN.C

[version ordinateur format GIMP](#)

[version smartphone format GIMP](#)

Déploiement sur Heroku

Création du fichier [Procfile](#)

Clonage du repo dans heroku

```
heroku git:clone -a bc-ocdaphytonpr7
```

pour pousser le projet vers heroku

```
git push heroku master
```

Mise en place du déploiement automatique depuis github après [validation de Travis \(configuration\)](#) et de [l'analyse de couverture](#)

Flask et le TDD

Définition d'une liste de phrases à tester:

- Salut GrandPy ! Est-ce que tu connais l'adresse d'OpenClassrooms ?
- où se trouve l'Arc de Triomphe?
- Quelle est l'adresse de la Tour Eiffel?
- Dis Papy, c'est quoi l'adresse de l'Elysée?
- Tu connais l'adresse de l'Opéra Garnier?

Rédaction des pages de tests puis du code des modules pour obtenir les tests vert. (J'ai réussi à faire du TDD sur toutes les parties du module utils mais j'ai rédigé les tests de views.py après. Il a fallut que je recommence les tests avec les Mocks donc j'ai dû réécrire les tests après le code pour le module information)

J'ai passé énormément de temps sur les Mock avec l'utilisation de monkeypatch et MagicMock.

API Google Map et Wikimedia

Google Map

Création de l'identifiant sur <https://console.cloud.google.com>

Création de la variable d'environnement dans l'environnement virtuel

```
pipenv shell
echo "GMAPKEY=[PRIVATE_KEY]" > .env
echo $GMAPKEY
```

Création de la variable d'environnement dans Heroku

```
heroku config:set GMAPKEY=[PRIVATE_KEY]
```

il faut faire une clé pour le front avec une restriction sur l'adresse https et une autre clé pour le back sans restriction car Heroku peut changer l'adresse IP

Utilisation du paquet googlemaps : <https://github.com/googlemaps/google-maps-services-python>

Utilisation de l'API geocode pour récupérer:

- l'adresse complète => afficher dans la réponse
 - les coordonnées GPS => pour générer la carte
 - une combinaison nom de rue + ville => requête wikipedia
-

Wikipedia

Utilisation du paquet MediaWiki <https://github.com/zikzakmedia/python-mediawiki>

Récupération du contenu de la page (*content*) et découpage de la partie souhaitée => afficher dans la réponse

J'ai également fait une version avec requests directement sur l'adresse pour tester.

Arborescence du projet

```
.
├── app.py
├── .env
└── .gitignore
├── papyrobot
│   ├── __init__.py
│   └── static
│       ├── css
│       │   ├── bootstrap*
│       │   └── style.css
│       ├── fonts
│       │   └── URW Bookman L Bold Italic.ttf
│       ├── img
│       │   ├── favicon.ico
│       │   ├── github5.png
│       │   ├── grandpere.png
│       │   ├── loading.gif
│       │   ├── logo-twitter-rond.png
│       │   ├── motif.jpg
│       │   └── virgule-bulle.png
│       ├── js
│       │   ├── ajax.js
│       │   ├── bootstrap*
│       │   └── front.js
│       └── json
│           ├── dialog.json
│           ├── stopwords_custom.json
│           └── stopwords_fr.json
└── templates
    └── index.html
└── tests
    ├── __init__.py
    └── test_answer.py
```

```
    └── test_gmap.py
    └── test_question.py
    └── test_views.py
    └── test_wiki.py
    └── utils
        ├── __init__.py
        ├── answer.py
        ├── information.py
        └── question.py
    └── views.py
└── Pipfile
└── Pipfile.lock
└── Procfile
└── README.md
└── .travis.yml
```