

Exploiter l'API de HAL avec Python



1. Rappel sur les API de HAL

2. Présentation du langage Python

3. Quels outils pour coder en Python?

4. Les API HAL et Python : décrypter pour comprendre



Rappel sur les API de HAL

- API : Application Programming Interface
- Les API permettent une communication de machine à machine
- Une API est constituée de plusieurs parties :

Une base : <http://api.archives-ouvertes.fr/>

Un référentiel sur lequel se baser : `ref/anrproject` (par exemple)

Paramètres de l'URL de l'API : `?q=text:asie&wt=xml` (par exemple)

Un exemple : https://api.archives-ouvertes.fr/search/?q=title_t:neurologie&wt=xml

Présentation du langage Python

- Python est un langage de programmation interprété, qui favorise la programmation avec un typage dynamique et l'utilisation de fonctions et des classes.
- Il a été conçu par Guido Van Rossum à la fin des années 80, et il est aujourd'hui un des langages les plus populaires par sa simplicité d'utilisation et de compréhension.



Comparaison d'un « Hello World » en plusieurs langages

En Python

```
print(« Hello World »)
```

En C

```
#include <stdio.h>
int main() {
    printf("Hello, World!");
    return 0;
}
```

En Java

```
public class HelloWorld {
    public static void main(String[]
        args) {
        System.out.println("Hello,
            World");}
}
```

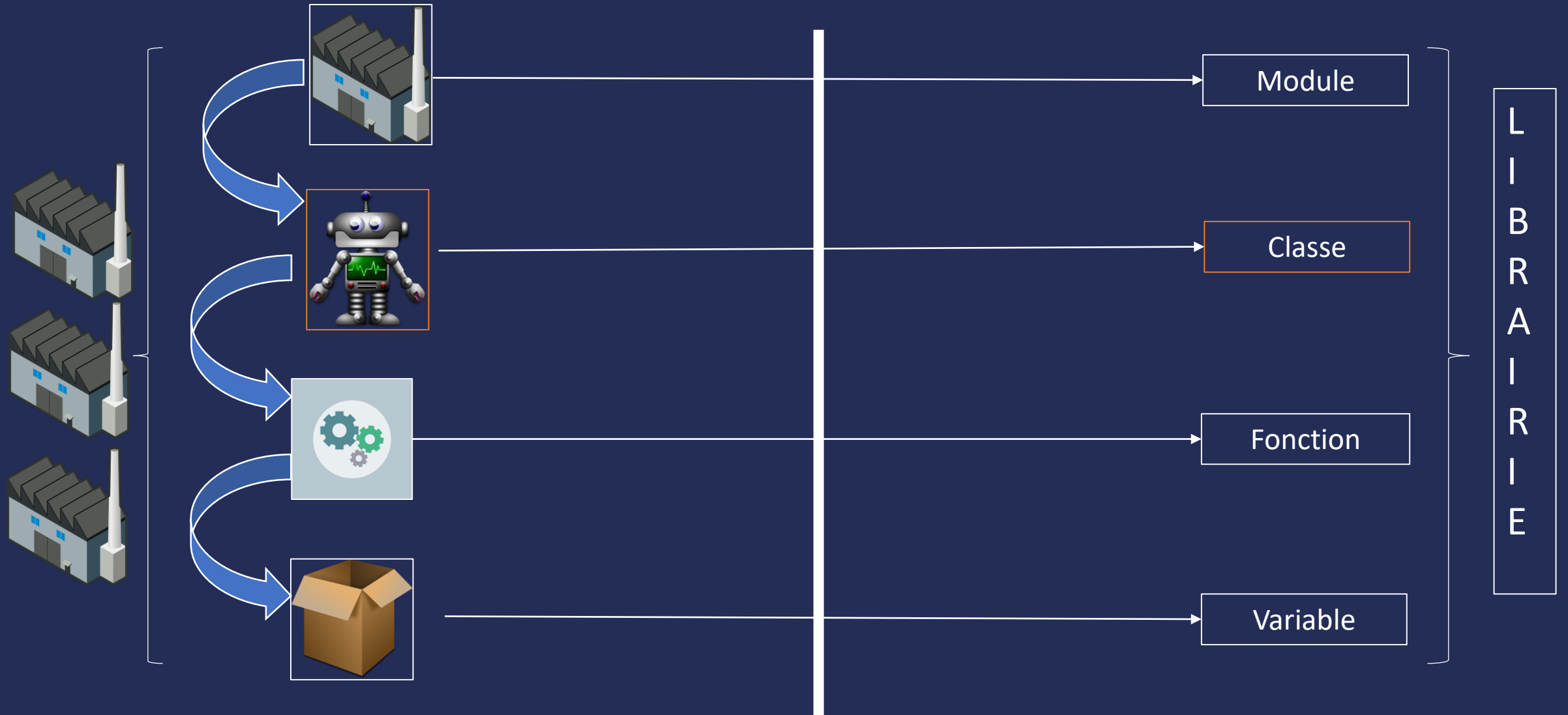
Que faire avec Python?

- Création de sites web
- Automatisation
- Web-scrapping
- Analyse et visualisation de données
- Calcul Scientifique
- Machine Learning et Deep Learning

Quels outils pour coder en Python?

- Des IDE (Environnements de Développement) : PyCharm, Visual Studio Code, Atom...
- Jupyter Notebook (avec la suite Anaconda)
- Des compilateurs Python en ligne : Google Colaboratory, CoCalc, Programiz (<https://www.programiz.com/python-programming/online-compiler/>)

Python : fondamentaux



Modèle

Import des librairies

```
Import nom_de_mon_module
```

Création de classes

```
class Nom_de_ma_classe :
```

Création de nos fonctions (ne pas oublier l'indentation – décaler la suite de notre code avec un alinéa)

```
    def fonction_1(valeur1, valeur2) :
```

Code de notre fonction, qui va comprendre d'autres fonctions et des variables

```
        total = valeur1 + valeur2
```

```
        return total
```

On peut ensuite créer d'autres fonctions à notre classe robot

```
    def fonction_2(valeur1, valeur2) :
```

```
        total = valeur1 – valeur2
```

```
        return total
```

Nom_de_mon_module.Nom_de_ma_classe.fonction_1(4,7) Donnera comme résultat 11

En l'absence de classe :

Nom_de_mon_module.fonction_2(9,5) Donnera comme résultat 4

Les variables (ou attributs)

- Une variable est une valeur que l'on va stocker en lui attribuant un nom et un type (de manière explicite ou implicite)
- On dit en Python qu'une variable possède un type, c'est-à-dire que la donnée qu'on va lui attribuer aura une certaine nature. En Python, il existe un grand nombre de type de variables (int, float, string). Certaines contiennent une seule valeur, d'autres peuvent en contenir plusieurs (list par exemple)
- Il est possible cependant de changer le type d'une variable
- Ex :
 - `a = 4`
 - `b = 2,65`
 - `c = « Bonjour »`
 - `liste_course = ['jambon', 'fraise', 'yaourt', 'fromage']`

Interactions entre les variables : conditions et boucles

- Condition « if », « elif » et « else » : inclure une ou plusieurs conditions sur la valeur d'une variable
- On peut cumuler les conditions avec les mots-clés « and » et « or »
- Le terme « while » permet de créer une boucle continue tant que la condition renseignée n'est pas pourvue
- Le terme « for » permet des itérations sur un élément ou une liste d'éléments. On peut spécifier une plage de début et de fin à la boucle « for » à l'aide du mot-clé « range »

Les fonctions (ou méthodes)

- Les fonctions sont des instructions que l'on passe à un programme pour manipuler les variables
- Il y a par défaut plusieurs variables pré-définies :

Ex : `print(« Bonjour »)`

Fera apparaitre à l'écran « Bonjour »

`question = input(« Quel est ton prénom »)`

Demanderà à l'utilisateur de rentrer une information, et attribuera à la variable « question » la valeur renseignée.

Il est possible de créer soi-même ses propres fonctions, en la définissant par le mot-clé « def »

Les classes

- Une classe est une structure qui va regrouper des variables et des fonctions (on parlera dans le cas d'une classe d'attributs et de méthodes) afin de pouvoir créer de nouvelles structures de données
- Pour créer une nouvelle classe, il suffit de la nommer en utilisant le mot clé « class » avant.
- Il n'est cependant pas obligatoire de faire appel à des classes!

Les modules

- Un module est un fichier Python qui va comprendre des classes, des fonctions et des variables auxquelles on va pouvoir faire appel directement dans notre code
- On fait appel à un module au début de notre code, grâce au mot-clé « import », et on peut le renommer à l'aide d'un alias, pour faciliter l'écriture du code
- Il est possible d'importer seulement une classe du module en utilisant le terme « from »
- Plusieurs modules peuvent être regroupés dans un package (ou librairie)

Comment utiliser les API de HAL avec Python (étapes)

1. Importer les librairies pour nous connecter au serveur de HAL générant les API
-> Module Requests
2. Se connecter à l'API HAL recherchée
-> Nécessite de préparer son API avant!
3. Récupérer et lire le résultat
-> Module Json
4. Extraire les informations recherchées dans le résultat de notre requête

Et après?

- Suivre l'atelier de Maxence Larrieu du 09/06 😊
- Suivre l'atelier de Brigitte Bidegaray-Fesquet du 09/06 😊
- Consulter les nombreuses ressources en ligne (La documentation Python -> <https://docs.python.org/fr/3/tutorial/>) ou en imprimé (« Python pour les SHS », d'Emilien Schultz et Matthias Buissonnier, ISBN : 978-2753581067)
- Tester, expérimenter des codes, des API, des librairies, partager son code
- Et si besoin, demander de l'aide -> julien.caugant@univ-amu.fr

MERCI!