

# Soutenance de stage

Kevin Voisin

10 juin 2022

## Présentation de l'entreprise



### Le laboratoire

Le Laboratoire Informatique, Image et Interaction (L3i) a été créé en 1993 et compte près d'une centaine de membres.

Il fait partie de plusieurs réseaux de recherches:

- ▶ régionaux: Fédération CNRS MIRES, ERT
- ▶ nationaux: GDR I3, GDR IRIS
- ▶ internationaux: IAPR

## Présentation du projet

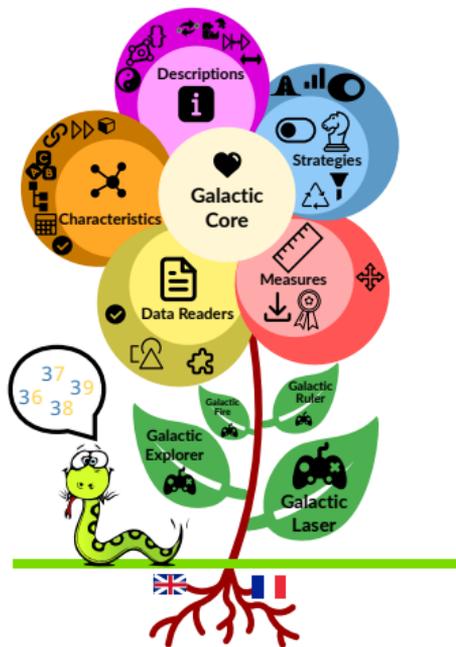
- ▶ Le projet Galactic est mené par le L3i et cherche à définir un framework en Python.
- ▶ Il étudie la théorie des treillis ainsi que l'analyse de concepts formels.



### Signification de GALACTIC

**GA**lois  
**LA**tices  
**C**oncept  
**T**heory  
Implicational systems and  
**C**losures

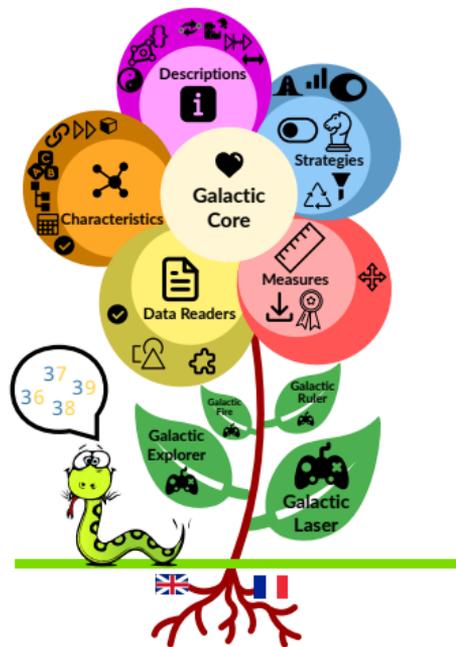
## Structure du projet



### Le framework possède une architecture simple:

- ▶ Un coeur
- ▶ Des applications:
  - ▶ Galactic Laser: construction et lecture de treillis.
  - ▶ Galactic Ruler: extraction de règles d'implication.
  - ▶ ...
- ▶ Différents plugins

## Structure du projet



### Les différents types de plugins:

- ▶ Des plugins de caractéristique
- ▶ Des plugins de description
- ▶ Des plugins de stratégie
- ▶ Des plugins de mesure
- ▶ Des plugins de lecteurs de données
- ▶ Des plugins de localisation

## Présentation du sujet

### Sujet

Ajout d'un plugin permettant l'extraction d'une caractéristique en précisant un chemin d'accès (chaîne de caractères) du type de ceux utilisés dans les sélecteurs CSS ou les chemins Xpath.

- ▶ Définition de la syntaxe
- ▶ Implémentation des fonctionnalités
- ▶ Réalisation des tests

## Python

### Syntaxe liée aux listes:

```
• liste=["a","b","c"]  
liste[0]  
liste[1:2]  
liste[index][index]
```

## Python

### Syntaxe liée aux listes:

```
• liste=["a","b","c"]  
liste[0]  
liste[1:2]  
liste[index][index]
```

### Syntaxe liée aux dictionnaires

```
dictionnaire = {"a":1,"b":2}  
dictionnaire[cle]  
dictionnaire[cle][cle]
```

## Sélecteurs CSS

Il existe des sélecteurs CSS qui permettent de choisir certains éléments en particuliers dans l'arborescence:

- ▶ Le sélecteur "+"
- ▶ Le sélecteur "~"
- ▶ Le sélecteur ">"
- ▶ Le sélecteur " "
- ▶ Le sélecteur "\*"

## Xpath

### Définition

Xpath permet la définition de chemins XML. Le but étant d'accéder à des données dans l'arborescence de façon simple.

Exemple de chemin :

```
| /order/items/book[title="Harry Potter"]/comment/text()
```

## Xpath

### Syntaxe liée aux listes:

```
["a", "b", "c"]
```

```
[1, [2, 3, 4]]
```

```
["a", "b", "c"] (2) /*Récupère "b"*/
```

```
[1, [2, 3, 4]] (2) (3) /*Récupère 4*/
```

```
array {1, 2, "galactic"}
```

```
/*Impossibilité d'imbriquer*/
```

## Xpath

### Syntaxe liée aux listes:

```
["a","b","c"]  
[1,[2,3,4]]  
["a","b","c"](2) /*Récupère "b"*/  
[1,[2,3,4]](2)(3) /*Récupère 4*/  
  
array {1,2,"galactic"}  
/*Impossibilité d'imbruquer*/
```

### Syntaxe liée aux dictionnaires (maps)

```
personne{"nom":"Dupont", "age":30}  
  
personne("nom")  
/*Récupère "Dupont"*/
```

## Comparaison des solutions

	Python	CSS	Xpath
<b>Listes:</b>			
Créer une Liste	✓	×	✓
Sélectionner un élément	✓	×	✓
Sélectionner plusieurs éléments	✓	×	×
Gestion listes imbriquées	✓	×	✓
<b>Maps:</b>			
Créer une Map	✓	×	✓
Sélectionner un élément	✓	×	✓
Gestion maps imbriquées	✓	×	✓
<b>Autres:</b>			
Présence de séparateur(s)	×	✓	✓
Sélectionner un élément	✓	✓	✓
Sélectionner plusieurs	✓	✓	✓

## Définition de la syntaxe

### Exemples de chemins corrects:

- ▶ `~/galactic/items[2]`
- ▶ `~[3][2]/nom\10`
- ▶ `~/universite/noms[1:20]`

## Définition de la syntaxe

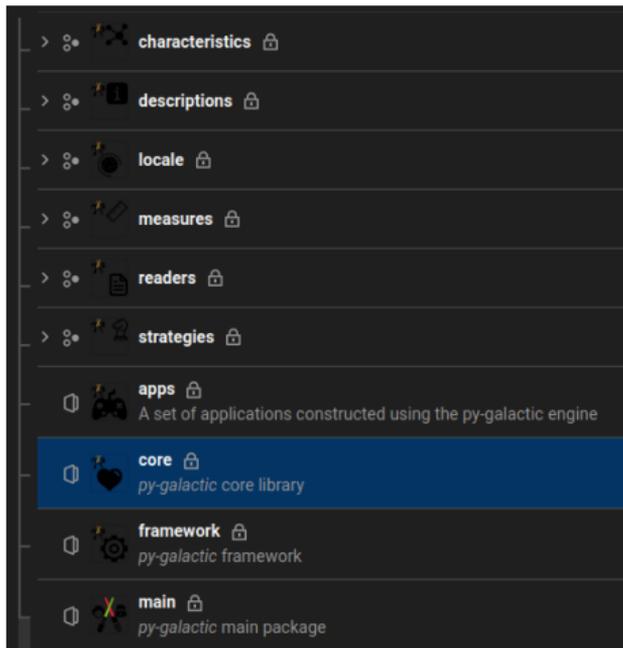
### Exemples de chemins corrects:

- ▶ `~/galactic/items[2]`
- ▶ `~[3][2]/nom\10`
- ▶ `~/universite/noms[1:20]`

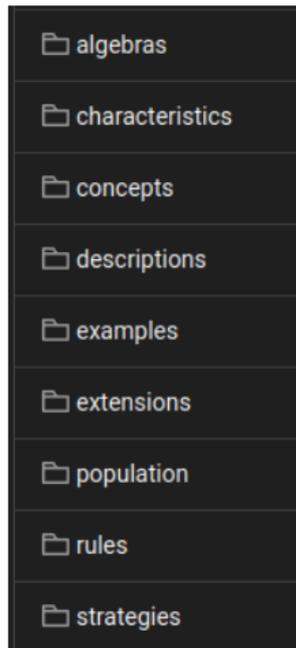
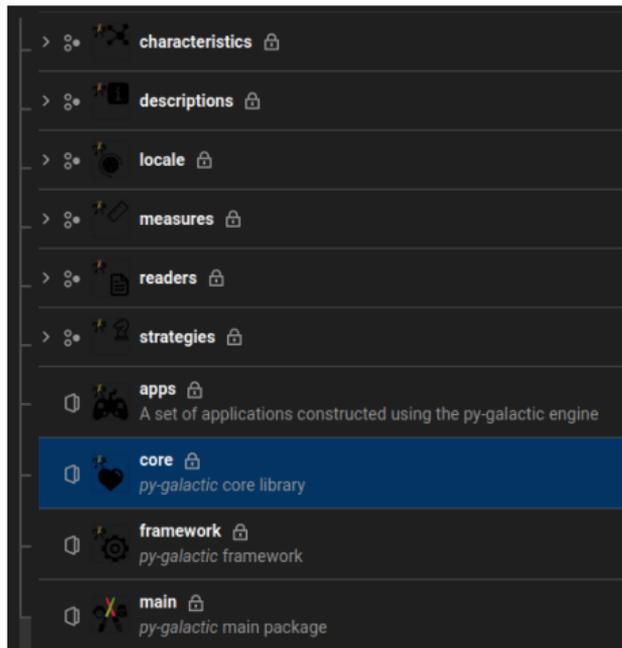
### Exemples de chemins incorrects

- ▶ `/galactic[10]`
- ▶ `~/galactic[1:3][1]`
- ▶ `~galactic[10]/nom`

## Mise en contexte



## Mise en contexte



## Développement du programme

Fonctions principales à définir:

- ▶ `__init__`
- ▶ `__call__`

## Développement du programme

### Fonctions principales à définir:

- ▶ `__init__`
- ▶ `__call__`

```
def __init__(self, path: str = "~",  
             **kwargs: Any, ) -> None:
```

## Développement du programme

### Fonctions principales à définir:

- ▶ `__init__`
- ▶ `__call__`

```
def __init__(self, path: str = "~",  
             **kwargs: Any, ) -> None:
```

```
def __call__(self, individual: Any) -> Optional[Any]:
```

## Conclusion

### Travail restant à réaliser

- ▶ Tests
- ▶ Correction de bugs
- ▶ Commentaires

## Conclusion

### Travail restant à réaliser

- ▶ Tests
- ▶ Correction de bugs
- ▶ Commentaires

### Ce que je retiens du stage

- ▶ Python
- ▶ Expressions régulières
- ▶ Développement logiciel
- ▶ Recherches