

# Matière à connaître



- Il n'y aura pas de questions QCM sur la théorie
- Possibilité de faire des tests en Python
  - code de test en Python
  - fonction `help()` de Python
- Il ne faut ***pas*** connaître par cœur toutes les fonctions de Python
  - Le syllabus théorie sera disponible  
(PDF et lien via INGIInious)
- Quelle matière à (ne pas) connaître?

# Matière – Partie 1

À connaître	Pas à connaître
Syntaxe de Python	
Expressions et instructions	
Opérateurs arithmétiques et ordre	
Variables et affectation	
Boucles : <b>for</b> , <b>while</b> , <b>range()</b>	Turtle graphics
Types de données : <b>int</b> , <b>float</b> , <b>bool</b> , <b>str</b>	
Conditionnelles et valeurs booléennes	
Fonctions : définition, paramètres, arguments, résultats	Modules et import
Portée et durée de vie des variables	Définition de modules
Spécifications pré/post	Modules particuliers : random, time, os, ...

# Matière – Partie 2

À connaître		Pas à connaître
<b>Strings</b>	notation [i], [:i], [i:j], [i:]	format()
	len(), +, *	
	strip(), split(), upper(), lower()	find(), join ()
	>, <, >=, <=, ==, !=	
	for ... in ...	if ... in, not in
<b>Listes</b>	notation [i], [:i], [i:j], [i:]	listes en compréhension
	len(), +, *	enumerate()
	append()	sort(), sorted(), extend()
	l.remove(3) ≠ del l[3]	insert(), count(), index(), reverse()
	for ... in ...	if ... in, not in
String → liste	list()	
... → String	str()	
<b>Tests</b>	assert	if __name__ == '__main__'

# Matière – Partie 2

À connaître		Pas à connaître
<b>Tuples</b>	notation () et [i]	
<b>Dictionnaires</b>	notation { : } et [ ]	keys(), values()
	notation [], get(), in	setdefault()
	for ... in, items()	
<b>Références</b>		copy ≠ deepcopy
Structures de données imbriquées et hétérogènes		
Structures muables vs. immuables		
<b>Fichiers</b>	open, close, with ... as	
	read, readline, readlines	
	write, writelines	
	for ... in file	
<b>Exceptions</b>	try... except, try ... except as	les différents types d'exceptions : IndexError, FileNotFoundError, ...
	raise	

# Matière – Partie 3

À connaître	Pas à connaître
is versus ==	
classes et objets	
attributs et méthodes d'instance	
création d'un objet	
appel d'une méthode	
appel à self ( self.attribut, self.method() )	
références	copy, deepcopy
égalité entre objets	
__init__, __str__, __eq__	les autres méthodes magiques

# Matière – Partie 3

À connaître	Pas à connaître
attributs privés	méthodes privées
méthodes accesseurs et mutateurs	
attributs et méthodes de classe	
composition de classes	classes imbriquées
héritage	héritage multiple
super()	
polymorphisme	tests unitaires avec unittest
la liaison dynamique de self	
portée et visibilité des variables	

# Matière – Structures et algorithmes

À connaître	Pas à connaître
Principes de comment un algorithme de recherche dichotomique fonctionne	Connaître par cœur le code d'un algorithme de recherche dichotomique
Comment utiliser une structure chaînée	Connaître par cœur l'implémentation de la classe LinkedList
Comment implémenter une liste chaînée	