LSINC 1101

Informatique 1
Introduction à la programmation

Modalités du cours

Charles Pecheur – Cristel Pelsser – Kim Mens

Les titulaires

Pr. Charles Pecheur

- Charles.Pecheur@uclouvain.be
- Première partie du cours

Pr. Cristel Pelsser

- Cristel.Pelsser@uclouvain.be
- Deuxième partie du cours

Pr. Kim Mens

- Kim.Mens@uclouvain.be
- Troisième partie du cours


QUIZ
TIME

Objectifs du cours d'informatique

A l'issue de ce cours, chaque étudiant sera en mesure de :

- **utiliser des outils** informatiques
- **utiliser** à bon escient les éléments d'un **langage de programmation** tel que Python,
- **concevoir et réaliser des programmes** corrects
  - de complexité moyenne
  - avec méthode et rigueur
- démontrer une bonne compréhension des concepts et de la méthodologie de la **programmation**
- et de la programmation **orientée-objet**
- utiliser la programmation pour le **projet de quadrimestre**

Comment atteindre ces objectifs ?

- Apprentissage actif
  - Typiquement par **groupes** de 2
  - A partir de **problèmes concrets** (1 par semaine)
- Permet d'acquérir
  - le **savoir** : par l'**étude** et la **pratique**
  - le **savoir-faire** : par la **pratique** uniquement

Organisation d'une semaine du cours

Une mission par semaine			
Jeudi			
	cours magistral	**Introduction**	Introduction à la matière liée au nouveau problème
Vendredi-weekend	*auto-apprentissage*	**Introduction**	Lecture du syllabus Faire les exercices préparatoires
Lundi	*séance tutorée*	**Bilan intermédiaire**	Réponses aux questions
Mercredi	*Travail en binôme*	**Réalisation**	Ecriture d'un programme Soumission au tuteur

Organisation d'une semaine du cours

Une mission par semaine			
Jeudi	*cours magistral*	**Restructuration**	*Restructuration de la matière vue durant la semaine*
	séance tutorée	**Bilan final**	Feedback sur la mission
			Question de bilan final
	cours magistral	**Introduction**	Introduction à la matière liée au nouveau problème
Vendredi-weekend	*auto-apprentissage*	**Introduction**	Lecture du syllabus Faire les exercices préparatoires
Lundi	*séance tutorée*	**Bilan intermédiaire**	Réponses aux questions
Mercredi	*Travail en binôme*	**Réalisation**	Ecriture d'un programme Soumission au tuteur

Syllabus du cours

Disponible en ligne

https://syllabus-interactif.info.ucl.ac.be/index/info1-theory

En anglais 

Version adaptée du livre open source :

How to Think Like a Computer Scientist – Learning with Python 3 (RLE)

by Peter Wentworth, Jeffrey Elkner, Allen B. Downey, and Chris Meyers

open-source, disponible gratuitement

A propos du syllabus

A **maîtriser** (et non mémoriser)

= **étudier** intelligemment

Lecture **obligatoire** !

Les références du cours s'y rapportent

Complémentaire aux transparents/capsules du cours

Disponible (en ligne) à l'examen (sur ordinateur)

(les exercices ne seront pas disponible en ligne à l’examen)

Manuel d'exercices

Disponible en ligne

https://syllabus-interactif.info.ucl.ac.be/index/info1-exercises

En français

Enoncés de toutes les missions

- matière à lire
- exercices de démarrage (interactifs)
- réalisation de la mission
- exercices complémentaires

Horaire LSINC1101

	Lu	Ma	Me	Je	Ve
8:00 10:00					
10:00 12:00					
13:30 15:30	APP			13:45 ! 15:45 APP Cours	
16:00			remise travail 16:00		

La matière

La matière faisant l'objet de l'examen comprend

tout ce qui a été dit ou montré au cours

oralement, sur écran ou à l'aide d'autres media,

et ne se limite donc exclusivement

au texte du "syllabus du cours"

Cours =

cours magistral +

séances tutorées

Evaluation

*Un **travail de programmation** est effectué chaque semaine.*

*Une **interrogation intermédiaire** a lieu en milieu de quadrimestre,*

*un **examen** en fin quadrimestre.*

*La **note finale** du cours prend en compte l'interrogation intermédiaire et les travaux durant le quadrimestre, en plus de la note de l'examen.*

Evaluation

Interrogation à mi-quadrimestre (S8)

- écrit sur papier

Examen en janvier / juin / août

- sur ordinateur (INGInious)

Note =

1/3 Interro + 2/3 Examen

si Interro > Examen

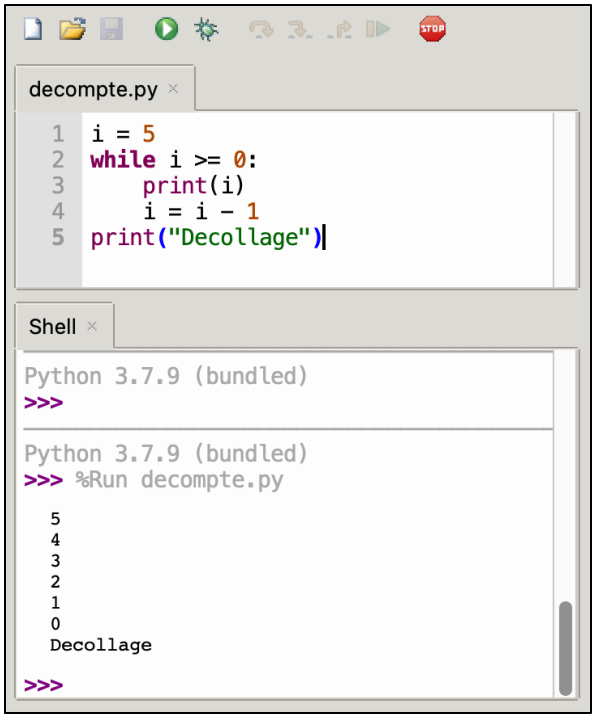
3/3 Examen

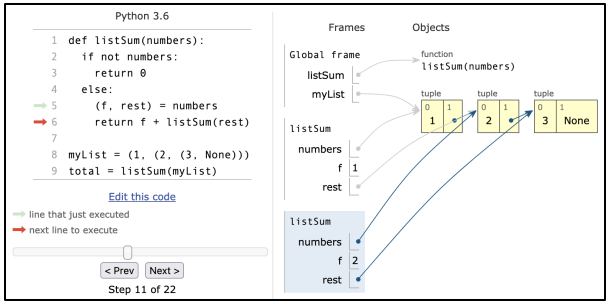
si Examen >= Interro

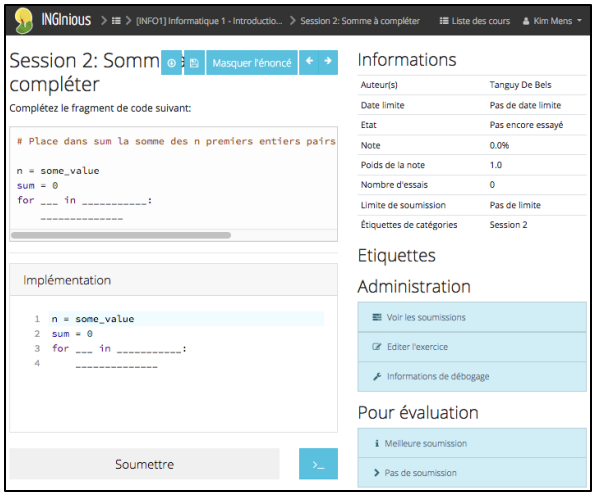
+ bonus éventuel

pour participation

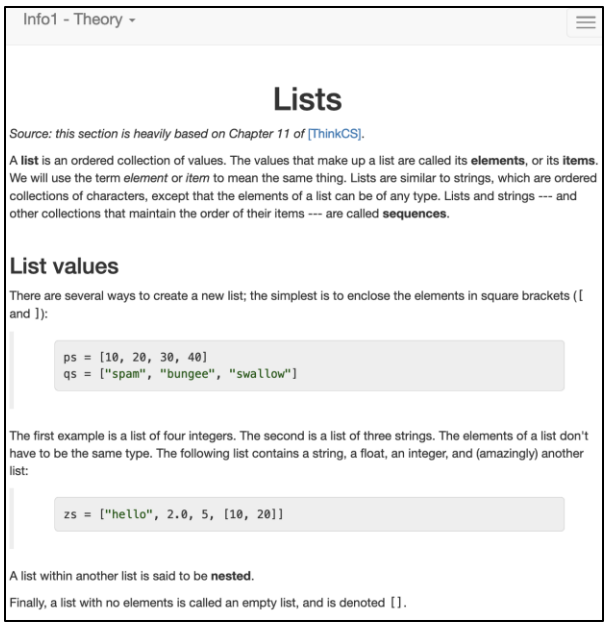
active aux missions

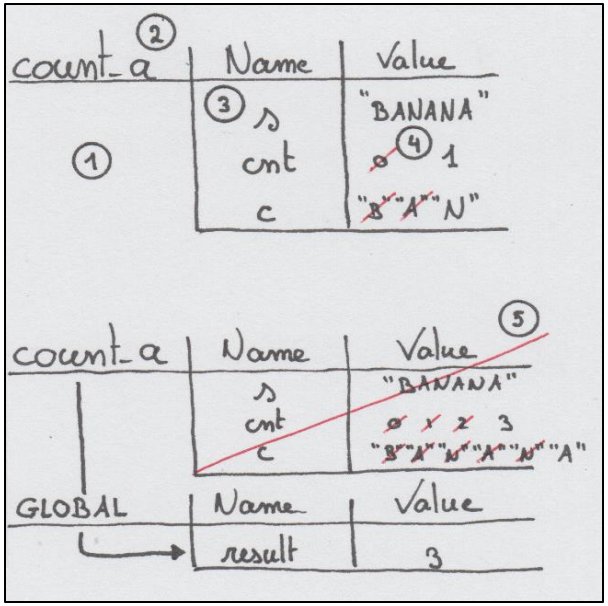




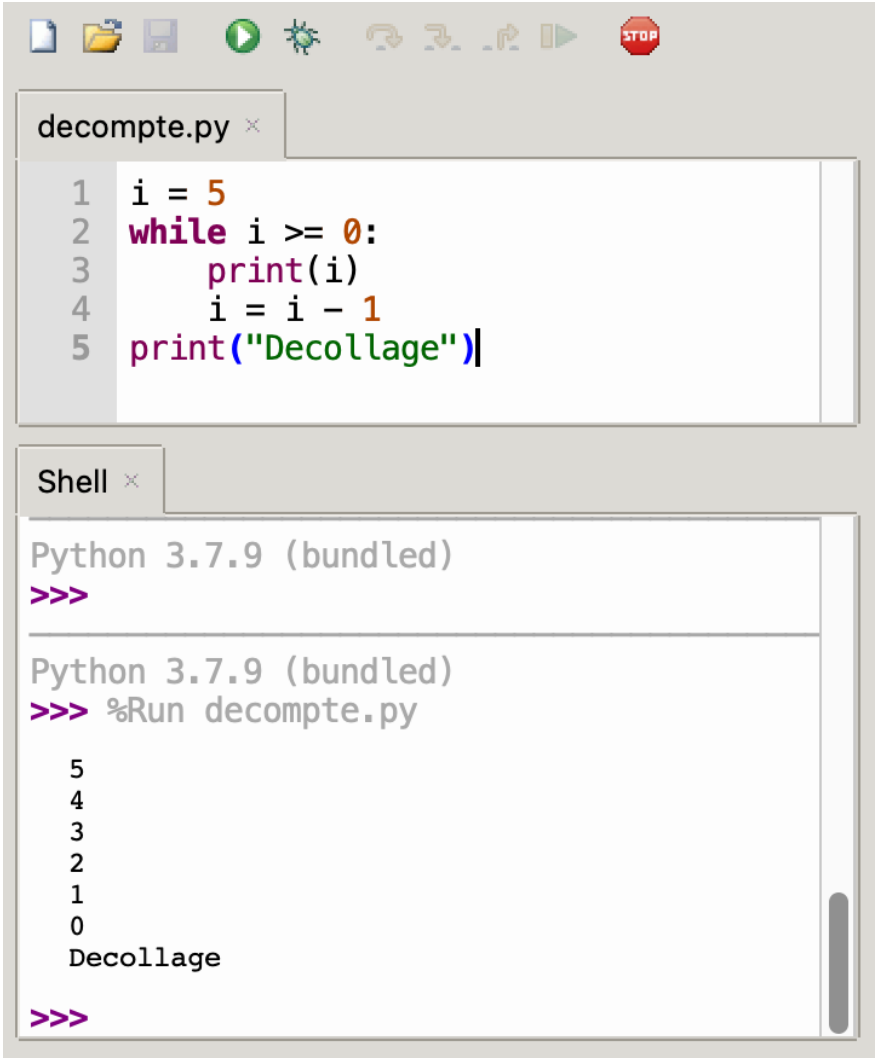


Outils informatiques





Thonny



https://thonny.org/

PythonTutor

Python 3.6

```
1  def listSum(numbers):
2    if not numbers:
3      return 0
4    else:
5      (f, rest) = numbers
6      return f + listSum(rest)
7
8  myList = (1, (2, (3, None)))
9  total = listSum(myList)
```

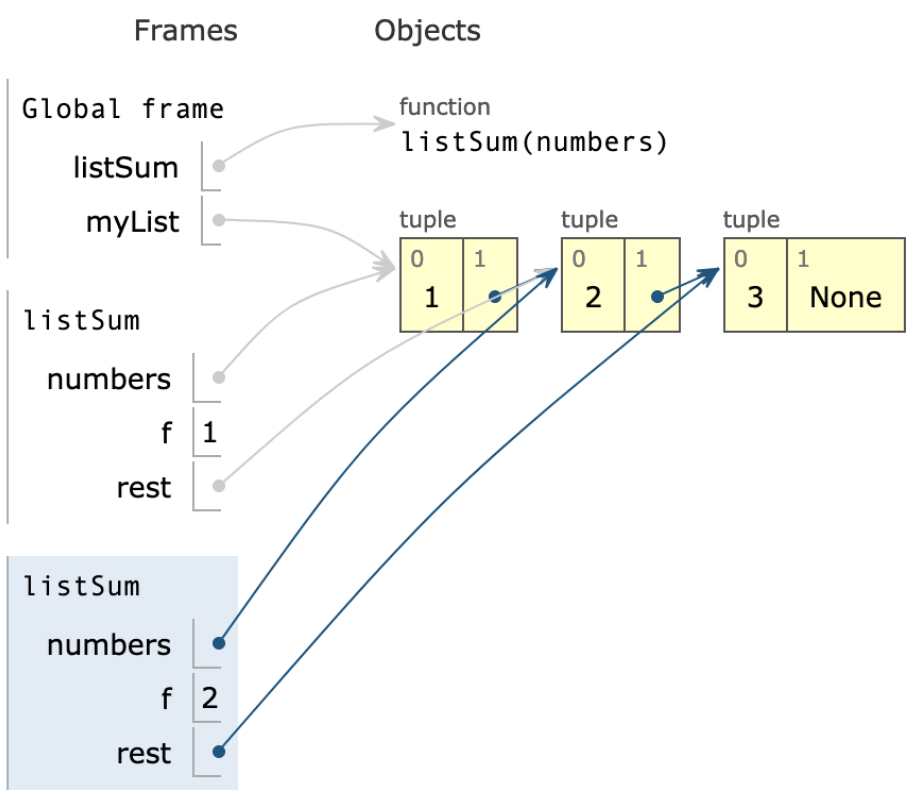
Edit this code

line that just executed

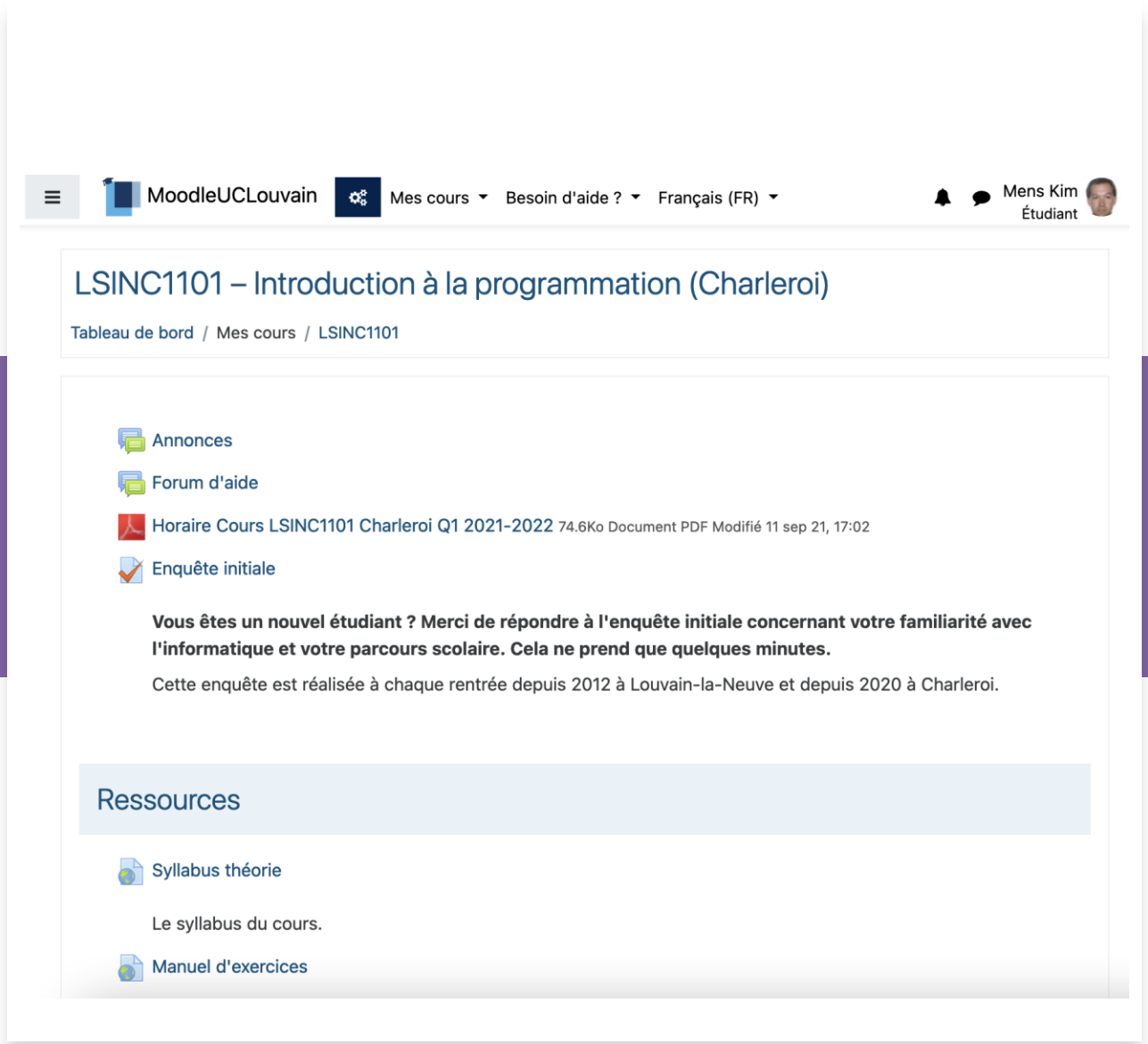
next line to execute

< Prev Next >

Step 11 of 22



https://pythontutor.com/



Moodle

https://moodle.uclouvain.be/course/view.php?id=4265

Lists

Source: this section is heavily based on Chapter 11 of [ThinkCS].

A **list** is an ordered collection of values. The values that make up a list are called its **elements**, or its **items**. We will use the term *element* or *item* to mean the same thing. Lists are similar to strings, which are ordered collections of characters, except that the elements of a list can be of any type. Lists and strings --- and other collections that maintain the order of their items --- are called **sequences**.

List values

There are several ways to create a new list; the simplest is to enclose the elements in square brackets ([and]):

```
ps = [10, 20, 30, 40]
qs = ["spam", "bungee", "swallow"]
```

The first example is a list of four integers. The second is a list of three strings. The elements of a list don't have to be the same type. The following list contains a string, a float, an integer, and (amazingly) another list:

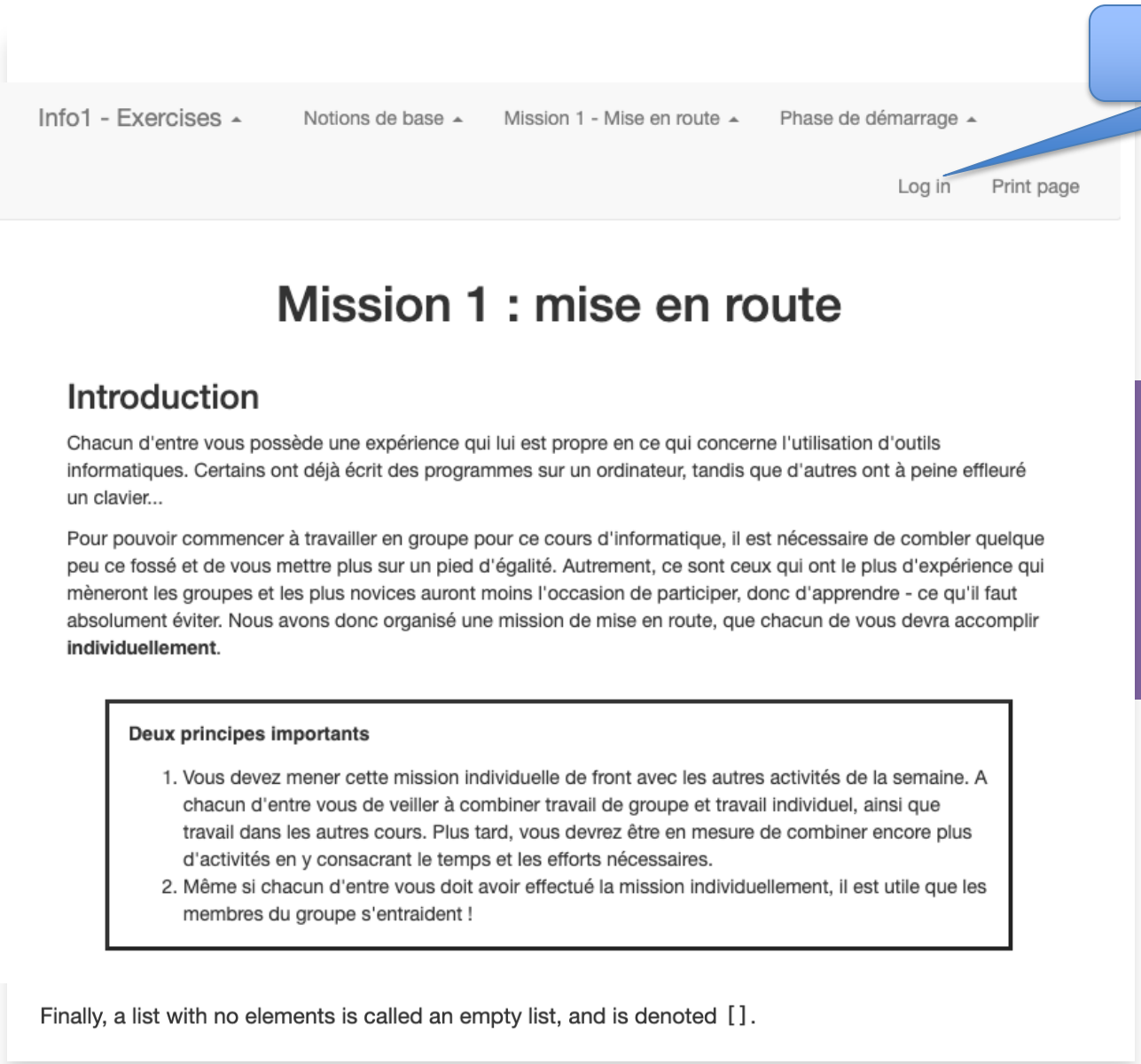
```
zs = ["hello", 2.0, 5, [10, 20]]
```

A list within another list is said to be **nested**.

Finally, a list with no elements is called an empty list, and is denoted `[]`.

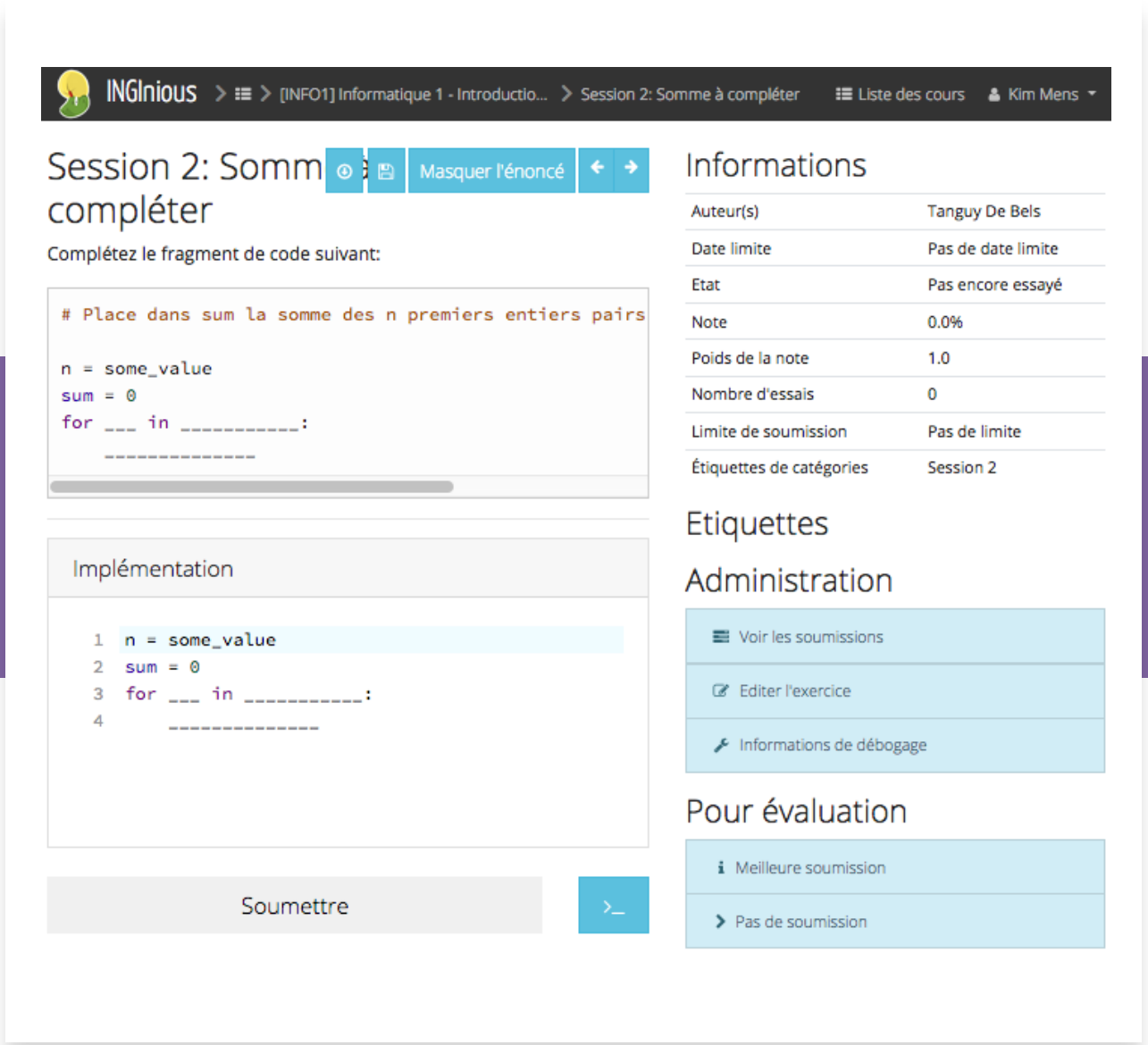
Syllabus en ligne : théorie

https://syllabus-interactif.info.ucl.ac.be/



Syllabus en ligne : exercices

https://syllabus-interactif.info.ucl.ac.be/



INGInious

Accès **à partir du syllabus d'exercices**

https://inginious.info.ucl.ac.be/course/LSINF1101-PYTHON

Exemples résolus avec étapes

Boucle while

1. Définir et **initialiser** les variables ;
2. Déterminez la **condition** d'arrêt (ou de continuation)
3. Écrire le **corps** de la boucle (mettre à jour la variable de contrôle jusqu'à la condition d'arrêt si nécessaire).

Calculez la somme Sum des n premiers entiers positifs

```
i = 1
Sum = 0
```

(1) **Initialiser**

```
While (i <= n)
```

(2) **Condition**

```
Sum += i
i += 1
```

(3) **Corps**

https://syllabus-interactif.info.ucl.ac.be/syllabus/info1-exercises/EXTRA/SLWE

A faire dès que possible

Si ce n'est déjà fait, d'urgence

obtenir son identifiant UCLouvain (inscription)

Ensuite

S'inscrire au cours sur Moodle et INGInious

A faire (avant lundi prochain)

Lire le syllabus du cours

https://syllabus-interactif.info.ucl.ac.be/index/info1-theory

→ chapitres 1 → 5

Faire la mission 1 "mise en route"

https://syllabus-interactif.info.ucl.ac.be/index/info1-exercises

→ Mission 1 - Mise en route

→ Démarrage : QCM + questions ouvertes

→ Réalisation d'un programme

Prochaines échéances

- Avant lundi
  - Avoir lu les chapitres correspondants du syllabus
  - Avoir fait les exercices de démarrage de la mission 1
- Lundi 13h30
  - Séance tutorée intermédiaire Mission 1
- Mercredi 16h00
  - Avoir terminé la phase de réalisation de la mission 1 et soumis le code sur INGInious
- Jeudi 13h45
  - Séance tutorée finale mission 1
  - Cours de restructuration mission 1
  - Introduction à la matière pour la mission 2