LSINC 1101

Informatique 1
Introduction à la programmation

Modalités du cours

Kim Mens
Siegfried Nijssen

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 – vendredi	*auto-apprentissage*	**Introduction**	Regarder les capsules vidéo d'introduction à la matière liée à une nouvelle mission
	auto-apprentissage	**Apprentissage**	Lecture du syllabus Répondre aux questions
Lundi (assistant)	*séance tutorée*	***Bilan final***	*Question de bilan final de la mission précédente*
		Bilan intermédiaire	Réponses aux questions pour la nouvelle mission
Lundi – mercredi	*Travail en binôme*	**Réalisation**	Ecriture du programme Soumission au tuteur
Jeudi (professeur)	*cours magistral*	**Restructuration**	Restructuration de la matière vue durant la semaine
	séance tutorée	**Bilan final**	Feedback sur la mission + évt. Introduction nouvelle mission

Les titulaires

Pr. Kim Mens

- Kim.Mens@uclouvain.be
- Début et troisième partie du cours

Pr. Siegfried Nijssen

- Siegfried.Nijssen@uclouvain.be
- Deuxième partie du cours

(Pr. Charles Pecheur

- Co-titulaire du cours à LLN)

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 **fortement recommandé** !

Les références du cours s'y rapportent

Complémentaire aux transparents/capsules du cours

Disponible (en ligne) à l'examen

(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					
14:00 16:00	APP			APP Cours	
16:00			remise travail 16:00		

La matière

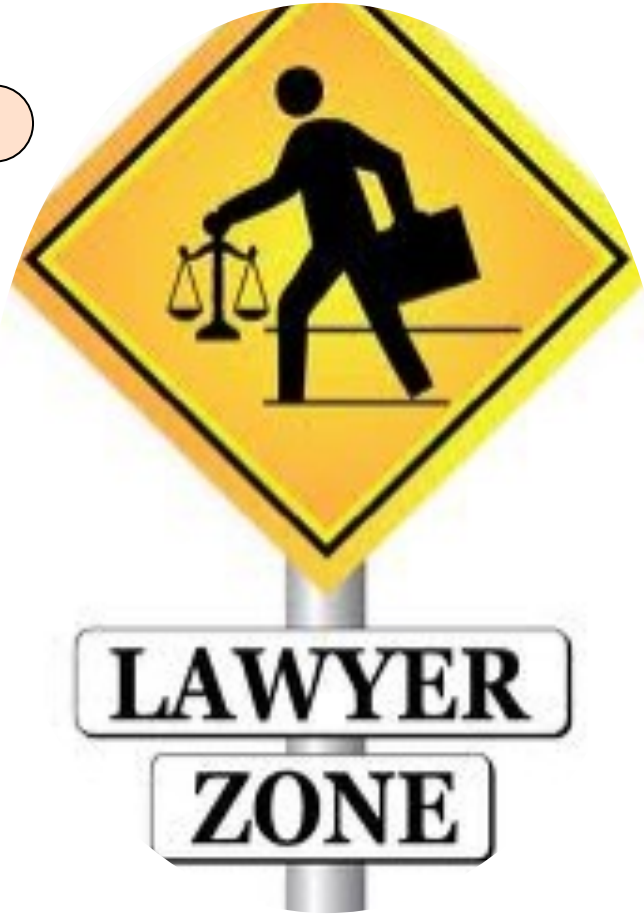
La matière faisant l'objet de l'examen comprend

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)

- (probablement) écrit sur papier

Examen en janvier / juin / août

- sur ordinateur (INGInious)

Note =

1/3 Interro + 2/3 Examen

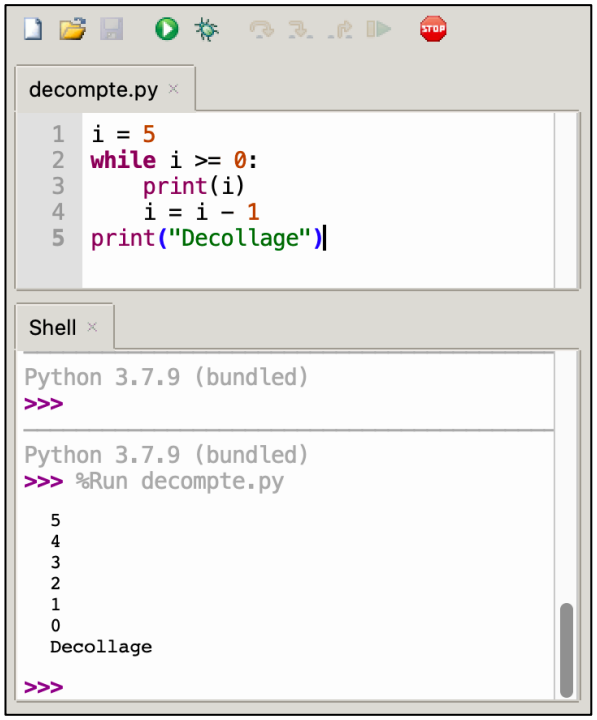
si Interro > Examen

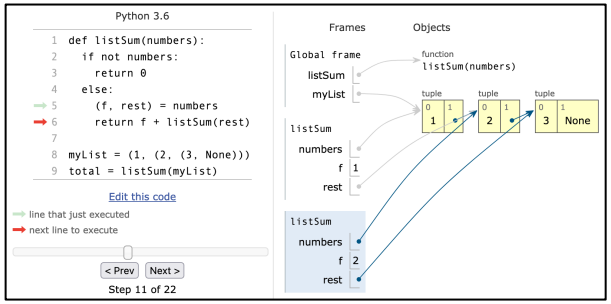
3/3 Examen

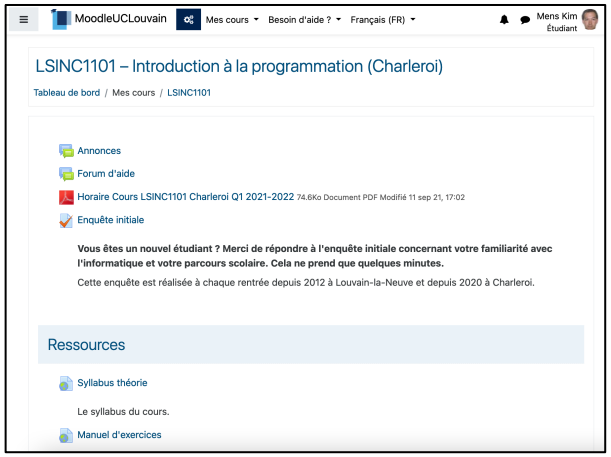
si Examen >= Interro

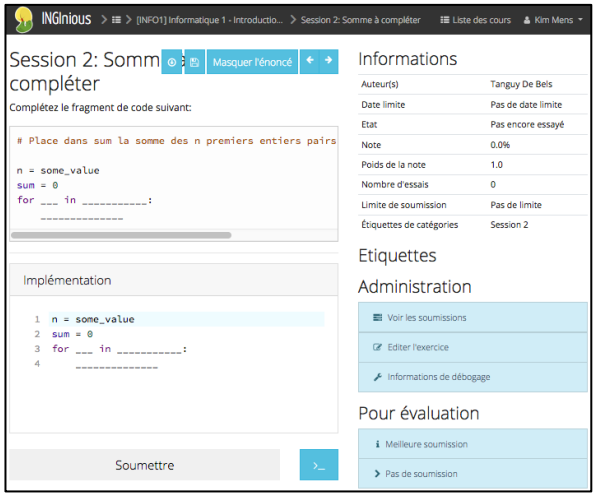
+ bonus éventuel

si participation suffisamment active aux missions

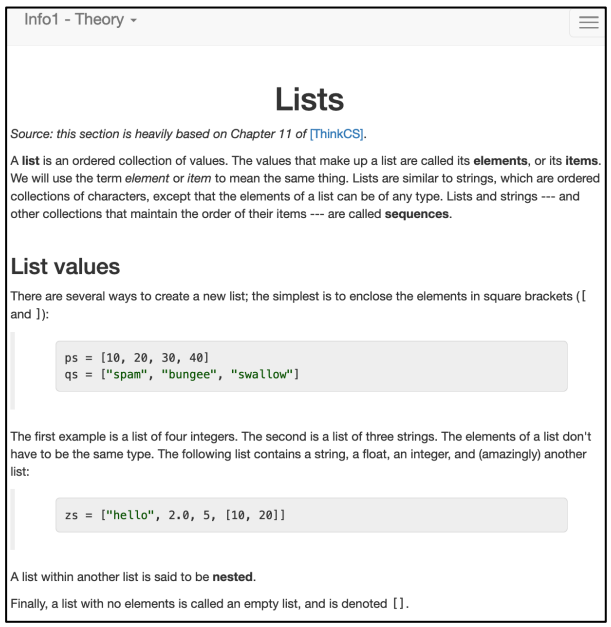


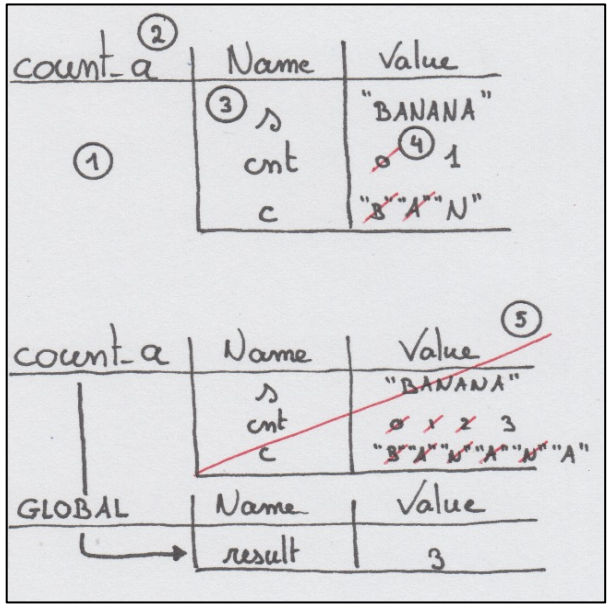




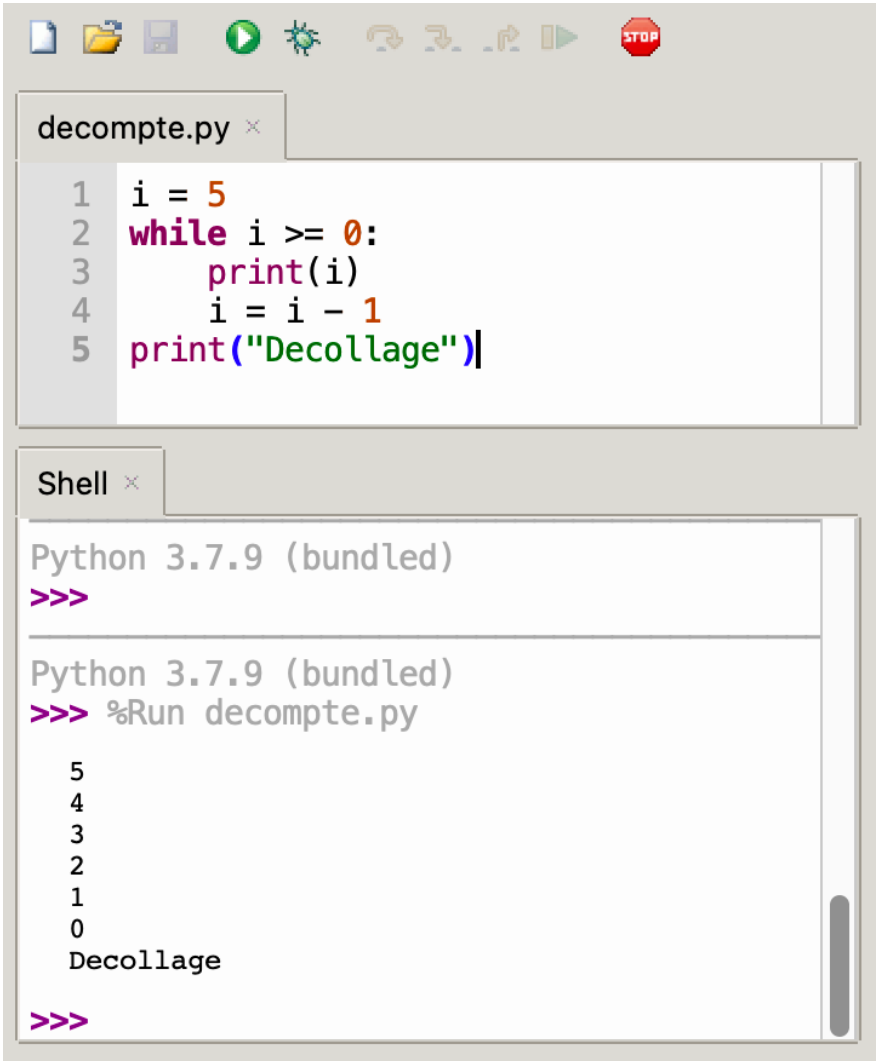


Outils informatiques





Thonny


decompte.py
1 i = 5
2 while i >= 0:
3 print(i)
4 i = i - 1
5 print("Decollage")
Shell
Python 3.7.9 (bundled)
>>>
Python 3.7.9 (bundled)
>>> %Run decompte.py
5
4
3
2
1
0
Decollage
>>>

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)
```

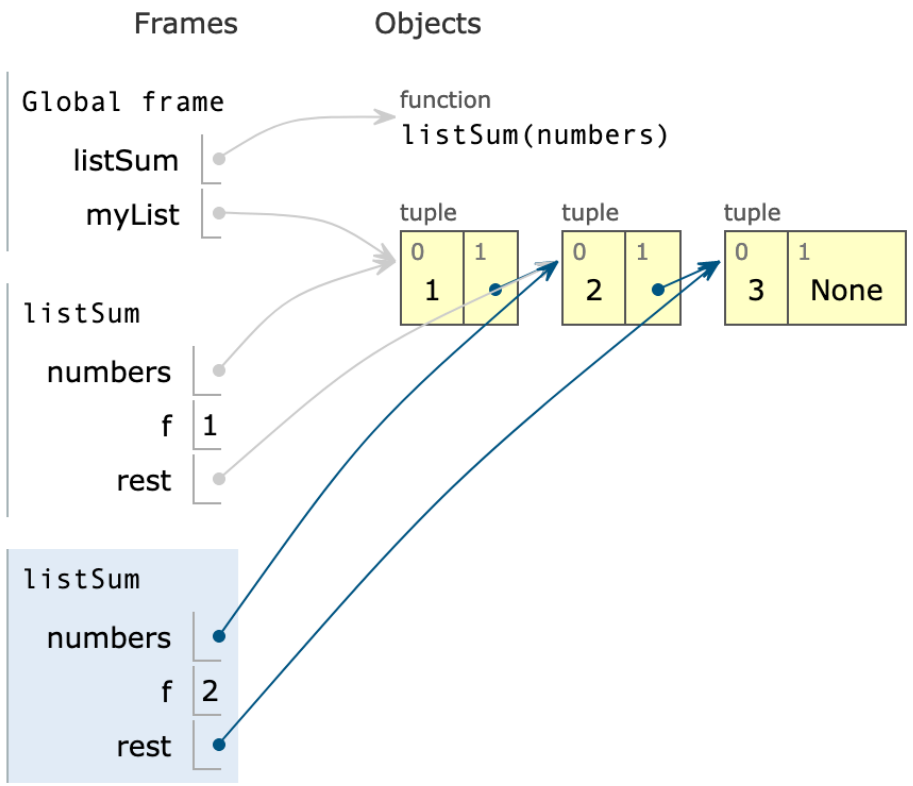
Edit this code

line that just executed

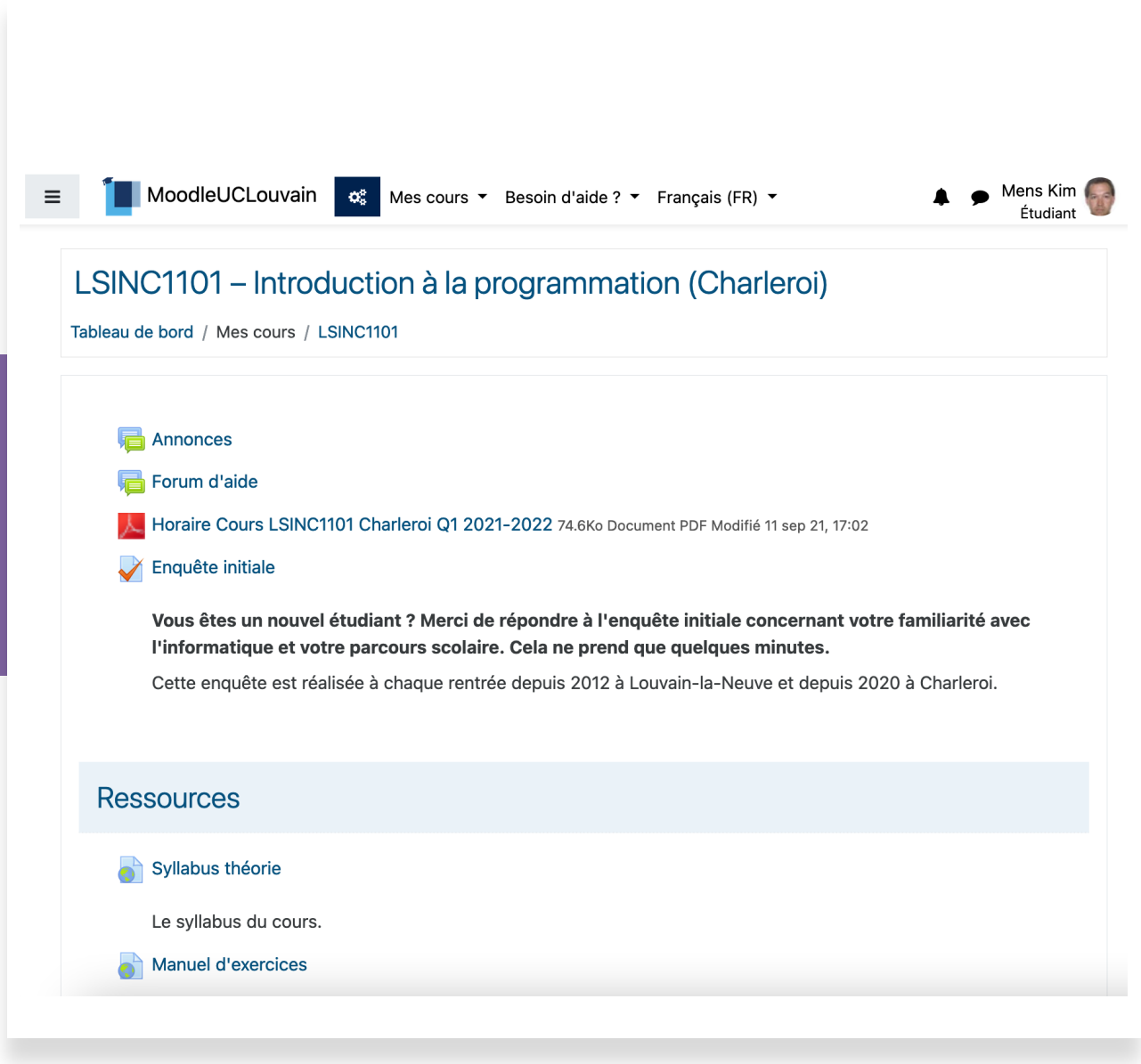
next line to execute

< Prev Next >

Step 11 of 22



https://pythontutor.com/


MoodleUCLouvain
Mes cours
Besoin d'aide ?
Français (FR)
Mens Kim
Étudiant
LSINC1101 – Introduction à la programmation (Charleroi)
Tableau de bord / Mes cours / LSINC1101
Annonces
Forum d'aide
Horaire Cours LSINC1101 Charleroi Q1 2021-2022 74.6Ko Document PDF Modifié 11 sep 21, 17:02
Enquête initiale
Vous êtes un nouvel étudiant ? Merci de répondre à l'enquête initiale concernant votre familiarité avec l'informatique et votre parcours scolaire. Cela ne prend que quelques minutes.
Cette enquête est réalisée à chaque rentrée depuis 2012 à Louvain-la-Neuve et depuis 2020 à Charleroi.
Ressources
Syllabus théorie
Le syllabus du cours.
Manuel d'exercices

Moodle

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:

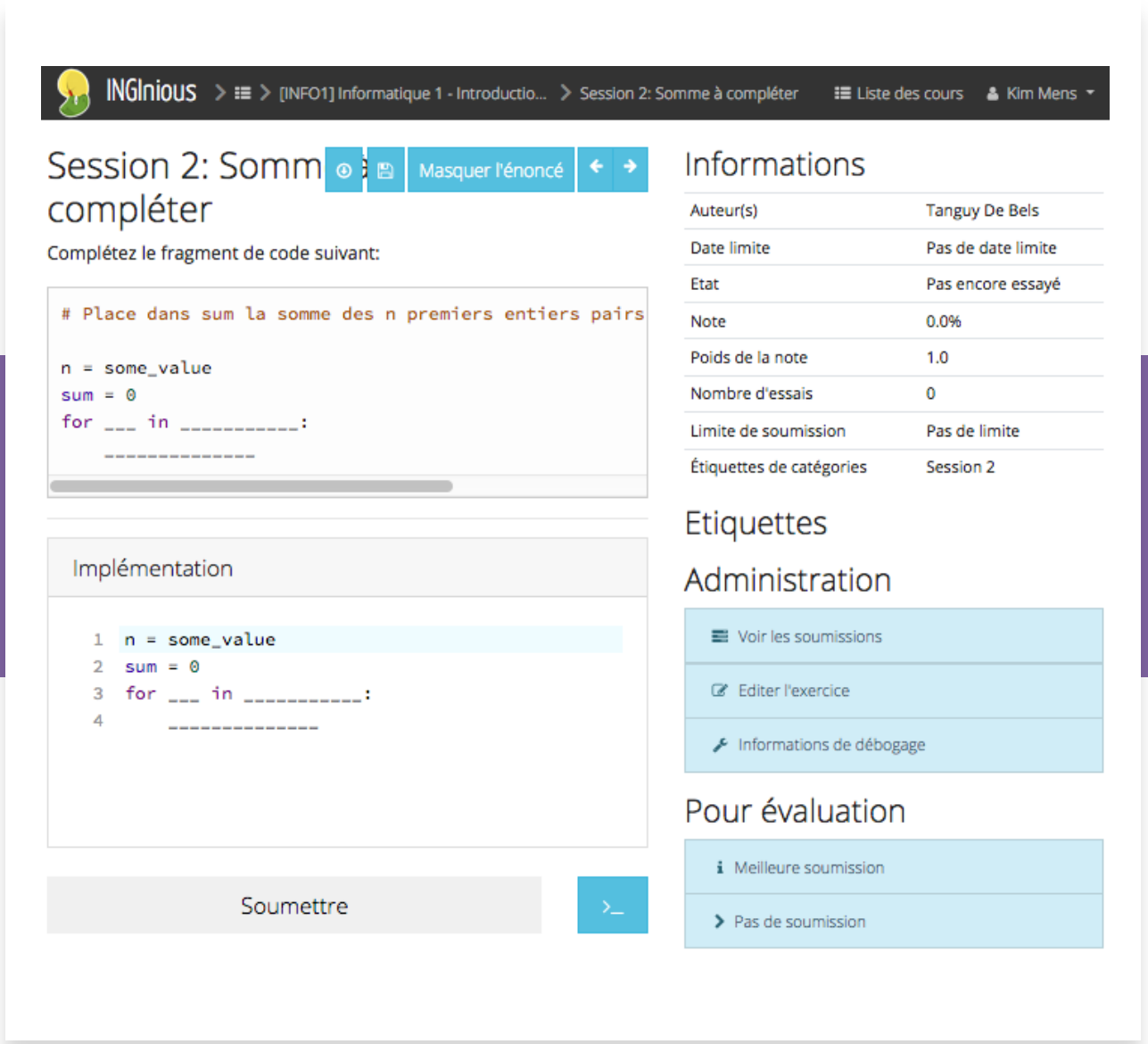
```
zs = ["hello", 2.0, 5, [10, 20]]
```

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+ exercices)



INGInious

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

Explicit Tracing

Concept name

Mémoire dynamique :

Description

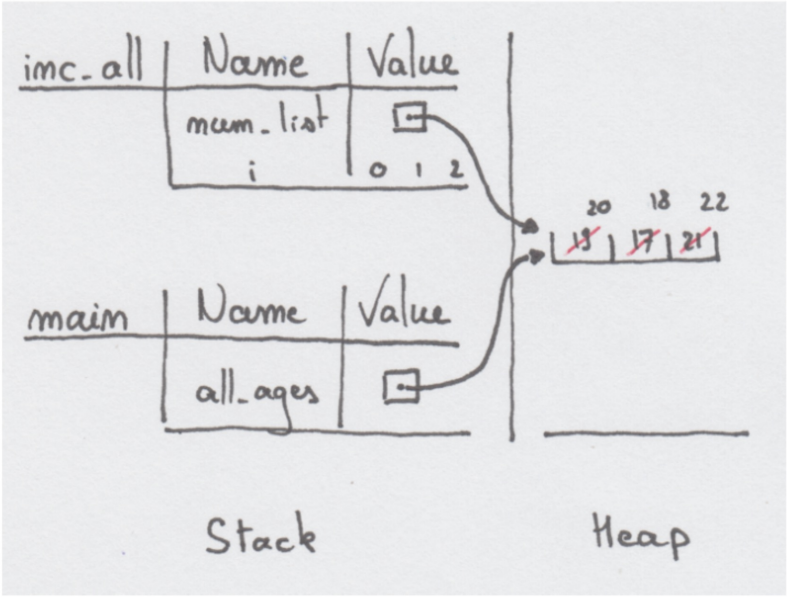
Les flèches représentant des pointeurs vers une adresse mémoire permettent d'expliquer le concept de passer des références comme arguments.

```
def inc_all(num_list):
    for i in range(len(num_list)):
        num_list[i] += 1

all_ages = [19, 17, 21]
inc_all(all_ages)
```

Code example

Tracing example



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 dès que possible

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

Lire le syllabus du cours

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

→ chapitres 1 → 5

Prochaines échéances

- Avant lundi
  - Avoir lu les chapitres du syllabus
  - Avoir fait les exercices de démarrage
- Lundi 14h
  - Séance tutorée intermédiaire Mission 1
- Mercredi 16h00
  - Avoir terminé la phase de réalisation de la mission 1 et soumis le code sur INGInious
- Jeudi 14h
  - Séance tutorée finale Mission 1
  - Cours restructuration Mission 1, introduction Mission 2