

# LSINC 1101

## Informatique 1 Introduction à la programmation

Modalités du cours



Kim Mens  
Siegfried Nijssen

# Les titulaires

Pr. Siegfried Nijssen

- [Siegfried.Nijssen@uclouvain.be](mailto:Siegfried.Nijssen@uclouvain.be)
- Première partie du cours

Pr. Kim Mens

- [Kim.Mens@uclouvain.be](mailto:Kim.Mens@uclouvain.be)
- Deuxième partie du cours

(Pr. Charles Pecheur

- Co-titulaire du cours à LLN)



A neon sign on a brick wall. The sign consists of two lines of text: 'QUIZ' on the top line and 'TIME' on the bottom line. The word 'QUIZ' is rendered in a pink neon color, while 'TIME' is rendered in a cyan neon color. The text is enclosed within a neon border that is also split into pink and cyan sections. The background is a dark brick wall.

QUIZ  
TIME

**J'ai déjà utilisé un ordinateur**



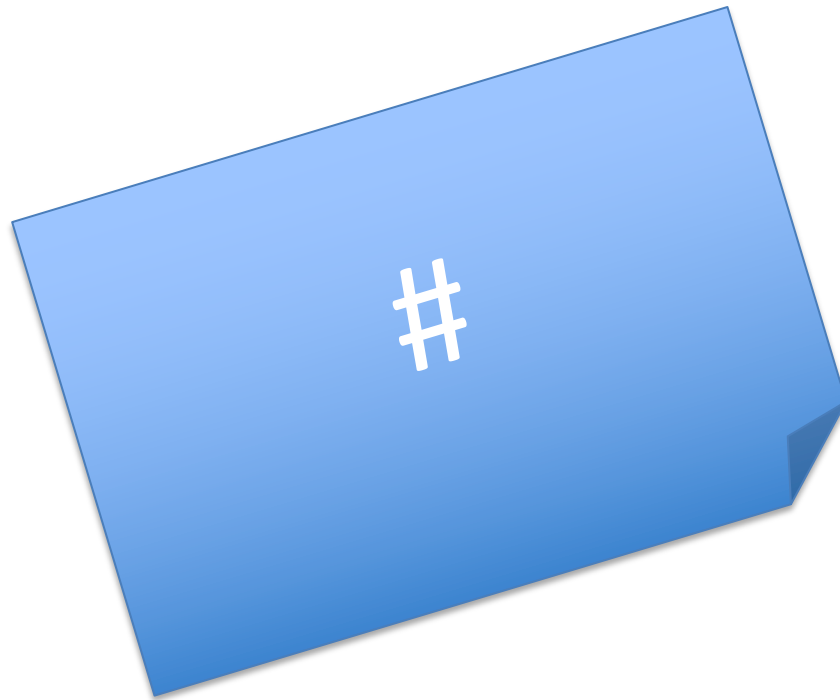
**J'ai déjà utilisé l'internet  
(email, web, chat, ...)**



**J'ai déjà utilisé un logiciel bureautique  
(traitement de texte, tableur, dessin, ...)**



**J'ai déjà suivi un cours d'informatique  
(combien?)**



# Selon moi, l'informatique c'est ...

0. "La programmation"	1. "La science des ordinateurs et des logiciels"
2. "Un sujet lié aux mathématiques et à la logique"	3. "Un domaine mystérieux que je veux mieux comprendre"
4. "Un ensemble de compétences techniques nécessaires pour le monde modern"	5. "Un moyen de résoudre des problèmes à l'aide de la technologie"
6. "Un domaine complexe qui semble difficile, mais fascinant"	7. "Un outil pour créer des jeux, des applications et des sites web"
8. "L'intelligence artificielle"	9. "Je ne suis pas sûr, c'est pourquoi je suis ici."



Combien d'heures de mathématique  
avez-vous eu?

#

$$\lim_{n \rightarrow \infty} \left(1 + \frac{1}{n}\right)^n$$

# Pourquoi avez-vous choisi de faire des études en sciences informatiques?

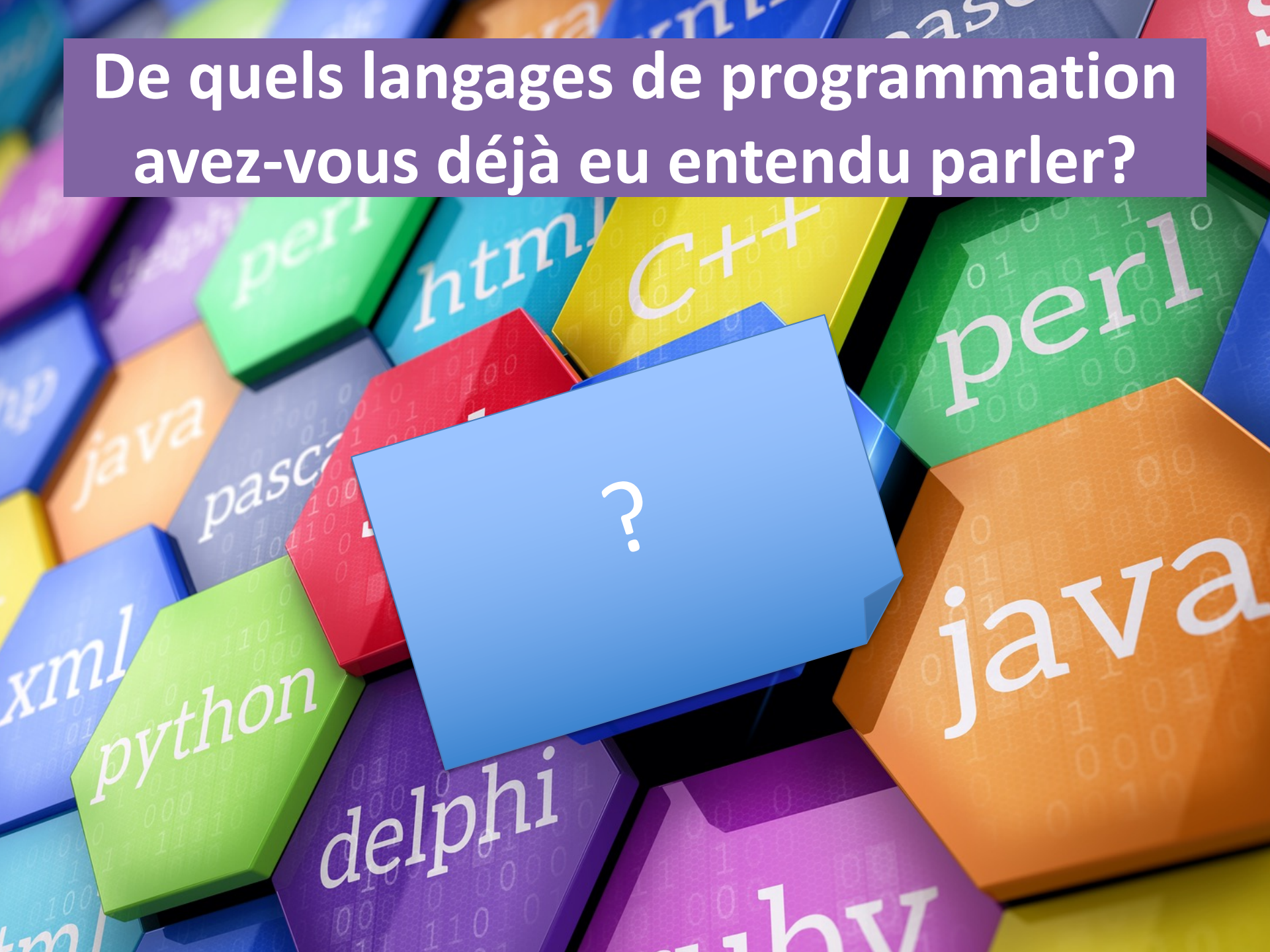
0. Opportunités de carrière	1. Impact et innovation
2. Diversité des domaines d'application	3. Résolution de problèmes
4. Rémunération	5. Évolution rapide du secteur
6. A cause d'un ami / membre de famille	7. Créativité
8. Autre	9. Je ne sais pas

# Pourquoi LSINC à Charleroi?

0. Bonne réputation	1. Proximité géographique de mon domicile
2. Moins d'étudiants	3. Charleroi, c'est le top.
4. Recommandation d'un ami / membre de famille	5. Publicité
6. Autre	6. Pourquoi pas?



De quels langages de programmation  
avez-vous déjà eu entendu parler?





J'ai déjà écrit un programme informatique  
(Dans quel langage(s)?)



# Quel est votre niveau de français?

Lecture (L)

Ecriture (E)

Orale (O)

0

(aucune connaissance)

**A1**

DÉCOUVERTE

EQUIVALENTS

CAMBRIDGE : YLE

**A2**

ÉLÉMENTAIRE

EQUIVALENTS

CAMBRIDGE : KET  
TOEIC : 245-380

**B1**

INTERMÉDIAIRE

EQUIVALENTS

CAMBRIDGE : PET  
TOEIC : 381-540  
IELTS : 4.5 - 5.0  
TOEFL : IBT 40-50

**B2**

INTERMÉDIAIRE  
SUPÉRIEUR

EQUIVALENTS

CAMBRIDGE : FCE  
TOEIC : 541-700  
IELTS : 5.5 - 6.0  
TOEFL : IBT 60-70

**C1**

AVANCÉ

EQUIVALENTS

CAMBRIDGE : CAE  
TOEIC : 701-890  
IELTS : 6.5 - 7.0  
TOEFL : IBT 80-90

**C2**

MAÎTRISE

EQUIVALENTS

CAMBRIDGE : CPE  
TOEIC : 90+  
IELTS : 7.0-7.5  
TOEFL : IBT 100-110

ÉLÉMENTAIRE

INTERMÉDIAIRE

AVANCÉ

# How is your knowledge of English?

Lecture (L)

Ecriture (E)

Orale (O)

0

(aucune connaissance)

**A1**

DÉCOUVERTE

EQUIVALENTS

CAMBRIDGE : YLE

**A2**

ÉLÉMENTAIRE

EQUIVALENTS

CAMBRIDGE : KET  
TOEIC : 245-380

**B1**

INTERMÉDIAIRE

EQUIVALENTS

CAMBRIDGE : PET  
TOEIC : 381-540  
IELTS : 4.5 - 5.0  
TOEFL : IBT 40-50

**B2**

INTERMÉDIAIRE  
SUPÉRIEUR

EQUIVALENTS

CAMBRIDGE : FCE  
TOEIC : 541-700  
IELTS : 5.5 - 6.0  
TOEFL : IBT 60-70

**C1**

AVANCÉ

EQUIVALENTS

CAMBRIDGE : CAE  
TOEIC : 701-890  
IELTS : 6.5 - 7.0  
TOEFL : IBT 80-90

**C2**

MAÎTRISE

EQUIVALENTS

CAMBRIDGE : CPE  
TOEIC : 90+  
IELTS : 7.0-7.5  
TOEFL : IBT 100-110

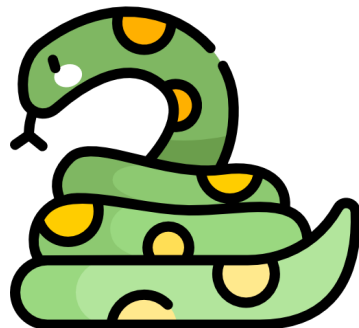
ÉLÉMENTAIRE

INTERMÉDIAIRE

AVANCÉ



# Quel est votre niveau de Python?



0

(aucune connaissance)

A1

DÉCOUVERTE

EQUIVALENTS

CAMBRIDGE : YLE

A2

ÉLÉMENTAIRE

EQUIVALENTS

CAMBRIDGE : KET  
TOEIC : 245-380

B1

INTERMÉDIAIRE

EQUIVALENTS

CAMBRIDGE : PET  
TOEIC : 381-540  
IELTS : 4.5 - 5.0  
TOEFL : IBT 40-50

B2

INTERMÉDIAIRE  
SUPÉRIEUR

EQUIVALENTS

CAMBRIDGE : FCE  
TOEIC : 541-700  
IELTS : 5.5 - 6.0  
TOEFL : IBT 60-70

C1

AVANCÉ

EQUIVALENTS

CAMBRIDGE : CAE  
TOEIC : 701-890  
IELTS : 6.5 - 7.0  
TOEFL : IBT 80-90

C2

MAÎTRISE

EQUIVALENTS

CAMBRIDGE : CPE  
TOEIC : 90+  
IELTS : 7.0-7.5  
TOEFL : IBT 100-110

ÉLÉMENTAIRE

INTERMÉDIAIRE

AVANCÉ



# Je dispose actuellement..

0. D'un <b>smartphone</b> à mon usage personnel	1. D'une <b>tablette</b> à mon usage personnel
2. D'un <b>ordinateur portable</b> à mon usage personnel	3. D'un <b>ordinateur fixe</b> à mon usage personnel
4. D'un <b>ordinateur portable</b> partagé	5. D'un <b>ordinateur fixe</b> partagé
6. D'une <b>tablette</b> partagé	7. Aucun

# L'ordinateur que j'utilise le plus souvent fonctionne sous l'environnement

0. Windows	1. MacOSX
2. Android	3. iOS
4. Linux (ou variante)	5. Autre (lequel)
6. Je ne sais pas	7. Je n'utilise jamais d'ordinateur

**J'ai déjà utilisé un interpréteur de commandes (invite de commandes, terminal, shell, ...)**



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

# Planning par semaine

Q1	SINC BAC 1	S1	S2	S3	S4	S5	S6	S7 (SMART)
	LSINC1101	16/09-20/09	23/09-27/09	30/09-04/10	07/10-11/10	14/10-18/10	21/10-25/10	28/10-01/11
LUNDI	12h45-13h45							
	13h45-15h45		LSINC1101 TP Intro M1 Assistant E6K—13	LSINC1101 TP Intro M2 Assistant E6K—13	LSINC1101 TP Intro M3 Assistant E6K—13	LSINC1101 TP Intro M4 Assistant E6K—13	LSINC1101 TP Intro M5 Assistant E6K—13	
MARDI	15h15-16h45	LSINC1101 CM Modalités du cours Professeur E6K—13						
JEUDI	13h45-14h45	LSINC1101 CM Restructuration APPO Professeur E6K-13	LSINC1101 TP Bilan M1 Professeur E6K—13	LSINC1101 TP Bilan M2 Professeur E6K—13	LSINC1101 TP Bilan M3 Professeur E6K—13	LSINC1101 TP Bilan M4 Professeur E6K—13	LSINC1101 TP Bilan M5 Professeur E6K—13	
	14h45-15h45	LSINC1101 CM Introduction M1 Professeur E6K-13	LSINC1101 CM Restructuration M1 Introduction M2 E6K-13	LSINC1101 CM Restructuration M2 Introduction M3 E6K-13	LSINC1101 CM Restructuration M3 Introduction M4 E6K-13	LSINC1101 CM Restructuration M4 Introduction M5 E6K-13	LSINC1101 CM Restructuration M5 - Introduction M6 E6K-13	

Q1	SINC BAC 1	S8	S9	S10	S11	S12	S13	S14
	LSINC1101	04/11-08/11	11/11-15/11	18/11-22/11	25/11-29/11	02/12-05/12	09/12-13/12	18/12-21/12
LUNDI	12h45-13h45	LSINC1101- Évaluation intermédiaire						
	13h45-15h45	LSINC1101 TP Intro M6 Assistant E6K—13	LSINC1101 TP Intro M7 Assistant E6K—13	LSINC1101 TP Intro M8 Assistant E6K—13	LSINC1101 TP Intro M9 Assistant E6K—13	LSINC1101 TP Intro M10 Assistant E6K—13	LSINC1101 TP Intro M11 Assistant E6K—13	LSINC1101 Feedback examen à blanc Assistant E6K—13
MARDI	15h15-16h45							
JEUDI	13h45-14h45	LSINC1101 TP Bilan M6 Professeur E6K—13	LSINC1101 TP Bilan M7 Professeur E6K—13	LSINC1101 TP Bilan M8 Professeur E6K—13	LSINC1101 TP Bilan M9 Professeur E6K—13	LSINC1101 TP Bilan M10 Professeur E6K—13	LSINC1101 TP Bilan M11 Professeur E6K—13	LSINC1101 CM Modalités de l'examen Professeur E6K— 13
	14h45-15h45	LSINC1101 CM Restructuration M6 - Introduction M7 E6K-13	LSINC1101 CM Restructuration M7 - Introduction M8 E6K-13	LSINC1101 CM Restructuration M8 Introduction M9 E6K-13	LSINC1101 CM Restructuration M9 Introduction M10 E6K-13	LSINC1101 CM Restructuration M10 Introduction M11 E6K- 13	LSINC1101 CM Restructuration M11 Professeur E6K—13	Questions- Réponses Professeur E6K— 13



# 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

(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

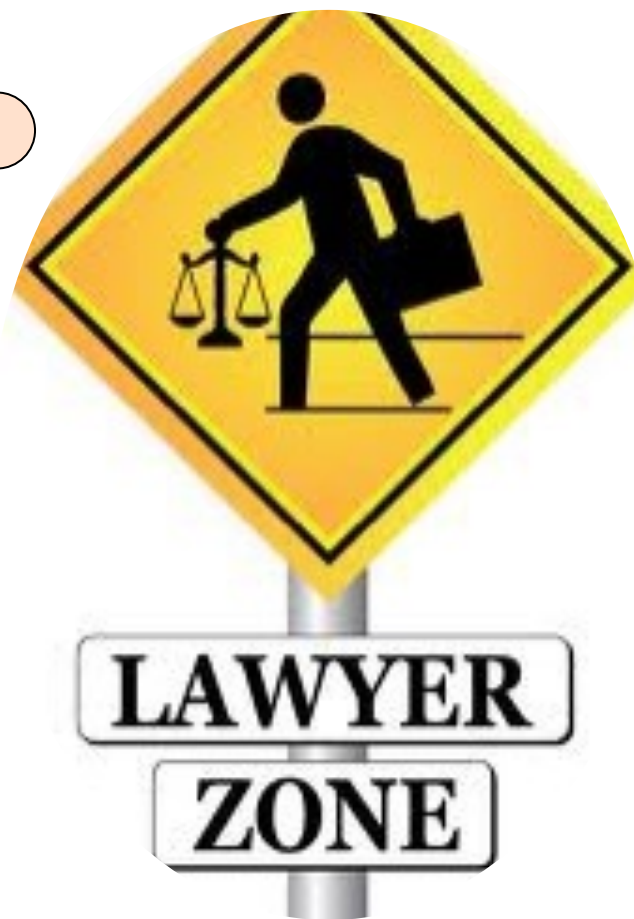
---

	Lu	Ma	Me	Je	Ve
8:00					
10:00					
10:00					
12:00					
13:45	APP			APP	
15:45				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  $\geq$  Interro

+ bonus éventuel

si participation suffisamment  
active aux missions

```

decompte.py x
1 i = 5
2 while i >= 0:
3     print(i)
4     i = i - 1
5 print("Decollage")

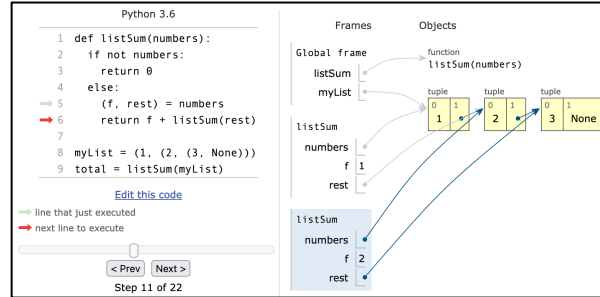
Shell x
Python 3.7.9 (bundled)
>>>

Python 3.7.9 (bundled)
>>> %Run decompte.py

5
4
3
2
1
0
Decollage

>>>

```



MoodleUCLouvain

LSINC1101 – Introduction à la programmation (Charleroi)

Tableau de bord / Mes cours / LSINC1101

- Announces
- Forum d'aide
- Horaires 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

## 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 [ ].

INGenius

Session 2: Somme à compléter

Informations

Auteurs(s): Tanguy De Beis

Date limite: Pas de date limite

Etat: Pas encore essayé

Note: 0.0%

Poids de la note: 1.0

Nombre d'essais: 0

Limite de soumission: Pas de limite

Étiquettes de catégories: Session 2

Étiquettes

Administration

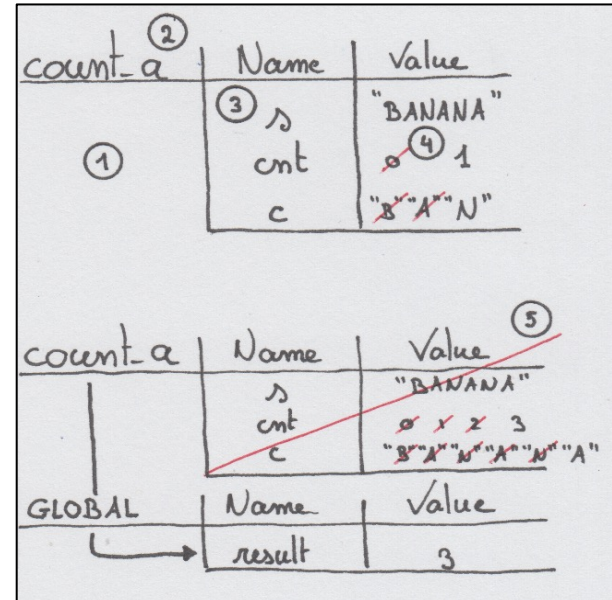
- Voir les soumissions
- Editer l'exercice
- Informations de débogage

Pour évaluation

- Melleure soumission
- Pas de soumission

Soumettre

# Outils informatiques





# Thonny



The screenshot displays the Thonny Python IDE interface. At the top, there is a toolbar with icons for file operations, running, and stopping. Below the toolbar, a window titled 'decompte.py' contains the following Python code:

```
1 i = 5
2 while i >= 0:
3     print(i)
4     i = i - 1
5 print("Decollage")
```

Below the code editor, a 'Shell' window shows the execution of the script. It starts with a prompt 'Python 3.7.9 (bundled) >>>' and then the command '%Run decompte.py'. The output of the script is displayed as follows:

```
Python 3.7.9 (bundled)
>>> %Run decompte.py

5
4
3
2
1
0
Decollage

>>>
```

<https://thonny.org/>

# PythonTutor

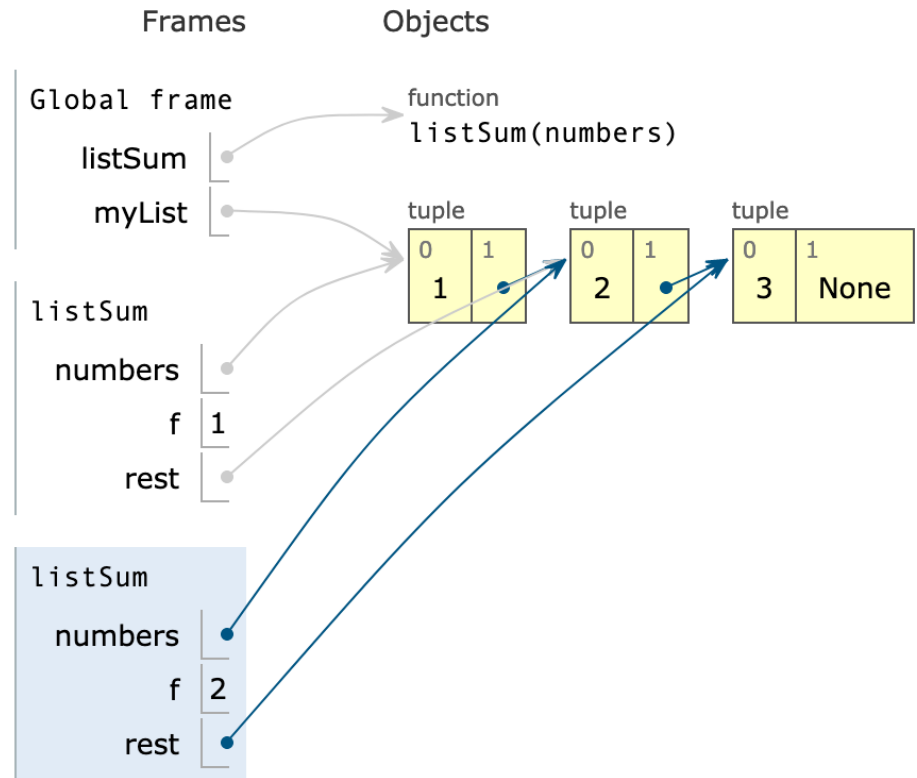
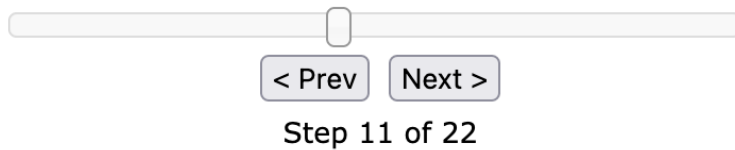
Python 3.6

```
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



<https://pythontutor.com/>

## 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)

# Session 2: Somme à compléter

Complétez le fragment de code suivant:

```
# Place dans sum la somme des n premiers entiers pairs

n = some_value
sum = 0
for ___ in _____:
    _____
```

Implémentation

```
1 n = some_value
2 sum = 0
3 for ___ in _____:
4     _____
```

Soumettre >\_

## Informations

Auteur(s)	Tanguy De Bels
Date limite	Pas de date limite
Etat	Pas encore essayé
Note	0.0%
Poids de la note	1.0
Nombre d'essais	0
Limite de soumission	Pas de limite
Étiquettes de catégories	Session 2

## Etiquettes

## Administration

- Voir les soumissions
- Editer l'exercice
- Informations de débogage

## Pour évaluation

- Meilleure soumission
- Pas de soumission

# INGInious

Accès à partir du syllabus d'exercices

# Explicit Tracing

Concept name

Mémoire dynamique :

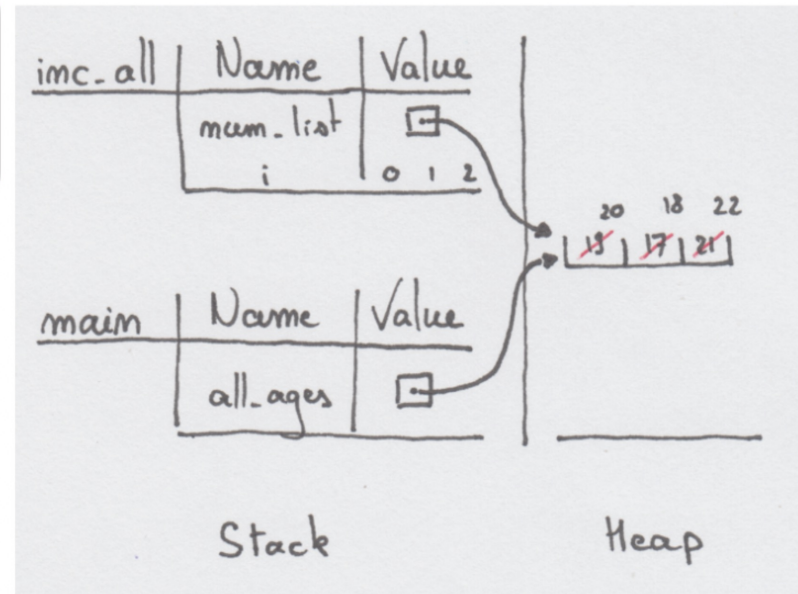
Description

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



# 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 ( $i \leq n$ )

(2) Condition

$Sum += i$   
 $i += 1$

(3) Corps

<https://syllabus-interactif.info.ucl.ac.be/syllabus/info1-exercices/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

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

A neon sign on a dark brick wall. The sign features the words "QUIZ" and "TIME" in a stylized, bubbly font. "QUIZ" is rendered in pink neon, and "TIME" is rendered in cyan neon. The text is enclosed within a neon border that is pink on the top and bottom, and cyan on the left and right. The border has a jagged, speech-bubble-like shape at the bottom right corner.

QUIZ  
TIME

# Avez-vous des questions pour mieux me connaître?

Qu'est-ce qui vous a amené à devenir professeur d'informatique ?	Quelle est votre partie préférée de l'enseignement de la programmation ?
Qu'est-ce que vous aimez faire en dehors de l'enseignement et de l'informatique ?	Avez-vous un film, un livre ou un jeu vidéo préféré ?
Y a-t-il des conseils que vous donneriez à un étudiant qui commence en programmation ?	Avez-vous un "fun fact" à partager sur vous-même ?
Qu'est-ce que vous trouvez le plus important pour réussir dans votre cours ?	Quelle est votre devise ou citation préférée pour motiver vos étudiants ?