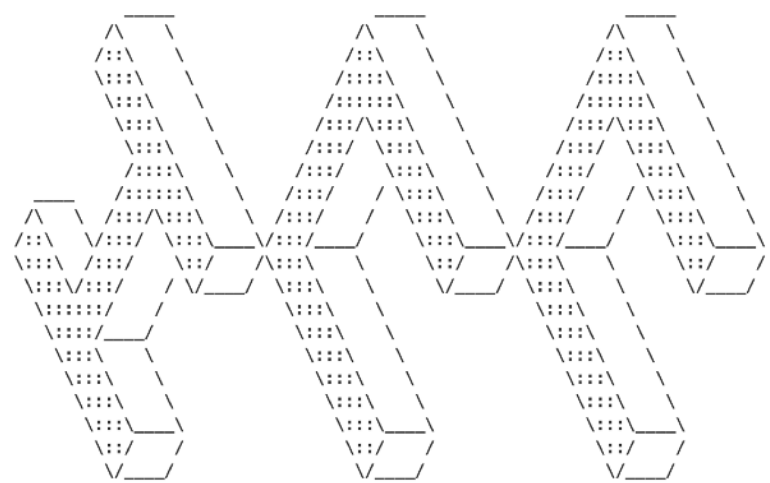
ascii.py ICC_Programmation • ascii.py/...

```
3  with open('ascii.txt') as f:
4      lines = f.read().splitlines()
5      for line in lines:
6          print(line)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER



Information, Calcul et Communication

Partie Programmation

Cours 1: Introduction

22.09.2023

Patrick Wang

Au sommaire cette semaine

1. Déroulement du semestre
2. Les outils pour programmer
3. Premiers pas en Python

Au sommaire cette semaine

1. Déroulement du semestre
2. Les outils pour programmer
3. Premiers pas en Python

1. Déroulement du semestre

Généralités

- Chaque semaine :
  - Cours : 9h15 – 10h00 en CE 1 3 avec retransmission sur Zoom
  - Exercices : 10h15 – 12h00 en BC07-08, CM 1 103, CM 1 110, CM 1 112
- Moodle : https://moodle.epfl.ch/course/view.php?id=15727
  - Supports de cours et d'exercices
  - Forum pour les annonces
  - Ed Discussion pour vos questions

1. Déroulement du semestre

Programme détaillé

- Semaine 1 – 6 : Cours et exercices
- Semaine 7 : Vendredi 3 novembre, 8h15 – 11h15 :
  - Examen intermédiaire (50% de la note finale)
  - Pas de cours de programmation cette semaine
- Semaine 8 – 13 : Cours et exercices
  - Mini-projet de programmation, individuel ou en binôme
  - 15% de la note finale
- Semaine 14 : Vendredi 22 décembre, 8h15 – 11h15 :
  - Examen final (35% de la note finale)

1. Déroulement du semestre

Séances supplémentaires de soutien

- Lundis soirs, de 17h00 à 19h00
- Salle CM 1 103
- Format :
  - Séances optionnelles
  - Soutien pour les deux parties théorique et programmation
  - Moment privilégié pour poser vos questions

1. Déroulement du semestre

Objectifs de la partie programmation

- Mettre en œuvre les concepts élémentaires de la programmation
- Découvrir quelques concepts “avancés”
- Se familiariser avec le langage de programmation Python
- Voir la programmation comme outil

Au sommaire cette semaine

1. Déroulement du semestre
2. Les outils pour programmer
3. Premiers pas en Python

2. Les outils pour programmer

C’est quoi “programmer” ?

- Programmer c’est :
  - Utiliser un langage de programmation
  - Construire un programme en rédigeant des instructions dans ce langage
  - Laisser l’ordinateur exécuter ces instructions pour résoudre (si notre programme est correct) un problème donné

2. Les outils pour programmer

Un langage de programmation

- Langage de programmation utilisé pour ce cours : Python 3.10
- Pourquoi parle-t-on de langage de programmation ?
  - Grammaire de Python 3.10 : https://docs.python.org/3.10/reference/grammar.html
  - Vocabulaire :
    - Mots-clés réservés par le langage : https://docs.python.org/3.10/reference/lexical_analysis.html#keywords
    - Des mots que d'autres personnes ont créés pour nous
    - Plein d'autres mots que nous pourrons nous-même créer !

2. Les outils pour programmer

Un *integrated development environment* (IDE)

- Visual Studio Code :
  - Disponible sur les machines en salle BC07-08
  - Disponible via VMWare Horizon Client en CM1 103, CM1 110, CM1 112
  - Disponible sur les machines virtuelles IC-CO-IN-INJ-2023-Fall depuis https://vdi.epfl.ch
- Possibilité d'utiliser un autre IDE, mais le support sera moins efficace

Au sommaire cette semaine

1. Déroulement du semestre
2. Les outils pour programmer
3. Premiers pas en Python

3. Premiers pas en Python

Quelques *instructions* pour commencer

- Voici quelques exemples d'instructions que l'on peut écrire en Python
- Essayons-les une par une :

```
22 + 9 + 2023
1.2 * 2
my_variable: str = "bon" + "bon"
print(my_variable)
print(2 * "bon")
```

3. Premiers pas en Python

Des données de plusieurs *types*

- Quelques types courants :
  - `int` (integer) : nombre entier
  - `float` (floating point) : nombre à virgule
  - `str` (string) : chaîne de caractères

```
22 + 9 + 2023
1.2 * 2
my_variable: str = "bon" + "bon"
print(my_variable)
print(2 * "bon")
```

3. Premiers pas en Python

Des *opérateurs* pour manipuler des données

- Opérations arithmétiques sur les nombres : (+, –, *, /, et d'autres...)
- Opérations sur les chaînes de caractères : (+, *)
- Attention ! Selon le type des opérandes, un même symbole peut réaliser des opérations différentes !

```
22 + 9 + 2023
1.2 * 2
my_variable: str = "bon" + "bon"
print(my_variable)
print(2 * "bon")
```

3. Premiers pas en Python

Des *variables* pour manipuler des données

- Une variable est un emplacement mémoire identifiable grâce à un nom qu'on lui attribue (on commence déjà à créer notre propre vocabulaire !)
- Attention à la syntaxe de l'assignation de variable :

```
my_var: type = value
```

- "On assigne à `my_var` la valeur `value`"
- "On affecte la valeur `value` à `my_var`"

```
22 + 9 + 2023
1.2 * 2
my_variable: str = "bon" + "bon"
print(my_variable)
print(2 * "bon")
```

3. Premiers pas en Python

Des *appels de fonctions*

- “Des mots que d’autres personnes ont créés pour nous”
- `print` est une fonction et fait partie de ces mots-là
- Syntaxe d’un appel de fonction : `func_name(arg1, ..., argn)`
- On peut dire : “On appelle la fonction `func_name` avec les arguments `arg1`, `...`, `argn`.”
- (On verra plus tard comment créer des fonctions)

```
22 + 9 + 2023
1.2 * 2
my_variable: str = "bon" + "bon"
print(my_variable)
print(2 * "bon")
```

3. Premiers pas en Python

Exemple d'un autre programme

```python
my_var: str = "un bout de texte"

# len() qui détermine la longueur d'une chaîne de caractères
length: int = len(my_var)

# Un exemple de "méthode" parmi plein d'autres
test: bool = my_var.endswith("xte")
print(test)

# On peut récupérer une lettre à un "indice" donné
first_letter: str = my_var[0]
last_letter: str = my_var[length - 1]
# On peut récupérer une sous-partie grâce au slicing
substring: str = my_var[3:7]
print(substring)
```

3. Premiers pas en Python

Exemple d'un autre programme

```python
my_var: str = "un bout de texte"

# len() qui détermine la longueur d'une chaîne de caractères
length: int = len(my_var)

# Un exemple de "méthode" parmi plein d'autres
test: bool = my_var.endswith("xte")
print(test)

# On peut récupérer une lettre à un "indice" donné
first_letter: str = my_var[0]
last_letter: str = my_var[length - 1]
# On peut récupérer une sous-partie grâce au slicing
substring: str = my_var[3:7]
print(substring)
```

3. Premiers pas en Python

Exemple d'un autre programme

```python
my_var: str = "un bout de texte"

# len() qui détermine la longueur d'une chaîne de caractères
length: int = len(my_var)

# Un exemple de "méthode" parmi plein d'autres
test: bool = my_var.endswith("xte")
print(test)

# On peut récupérer une lettre à un "indice" donné
first_letter: str = my_var[0]
last_letter: str = my_var[length - 1]
# On peut récupérer une sous-partie grâce au slicing
substring: str = my_var[3:7]
print(substring)
```

3. Premiers pas en Python

Exemple d'un autre programme

```python
my_var: str = "un bout de texte"

# len() qui détermine la longueur d'une chaîne de caractères
length: int = len(my_var)

# Un exemple de "méthode" parmi plein d'autres
test: bool = my_var.endswith("xte")
print(test)

# On peut récupérer une lettre à un "indice" donné
first_letter: str = my_var[0]
last_letter: str = my_var[length - 1]
# On peut récupérer une sous-partie grâce au slicing
substring: str = my_var[3:7]
print(substring)
```

3. Premiers pas en Python

Exemple d'un autre programme

```python
my_var: str = "un bout de texte"

# len() qui détermine la longueur d'une chaîne de caractères
length: int = len(my_var)

# Un exemple de "méthode" parmi plein d'autres
test: bool = my_var.endswith("xte")
print(test)

# On peut récupérer une lettre à un "indice" donné
first_letter: str = my_var[0]
last_letter: str = my_var[length - 1]
# On peut récupérer une sous-partie grâce au slicing
substring: str = my_var[3:7]
print(substring)
```

3. Premiers pas en Python

Exemple d'un autre programme

```python
my_var: str = "un bout de texte"

# len() qui détermine la longueur d'une chaîne de caractères
length: int = len(my_var)

# Un exemple de "méthode" parmi plein d'autres
test: bool = my_var.endswith("xte")
print(test)

# On peut récupérer une lettre à un "indice" donné
first_letter: str = my_var[0]
last_letter: str = my_var[length - 1]
# On peut récupérer une sous-partie grâce au slicing
substring: str = my_var[3:7]
print(substring)
```

3. Premiers pas en Python

Une séquence d'instructions

- Malgré l'exemple précédent, les instructions s'exécutent dans un ordre bien défini qui n'est pas toujours linéaire
- Un des enjeux de ce cours sera d'être capable de suivre mentalement l'exécution d'un programme (on parle de code tracing)

```python
my_var: str = "un bout de texte"

# len() qui détermine la longueur d'une chaîne de caractères
length: int = len(my_var)

# Un exemple de "méthode" parmi plein d'autres
test: bool = my_var.endswith("xte")
print(test)
```

3. Premiers pas en Python

Comment chercher de l'aide ?

- Questions sur le langage, sa syntaxe, et ses instructions ?
  - Chercher dans la documentation officielle de Python
  - https://docs.python.org/3.10/contents.html
- Questions sur un problème, un algorithme ?
  - Forum Ed Discussion
  - Séances de soutien
  - StackOverflow, Google, ...
  - Copilot ? ChatGPT ?

Bilan

Ce que l'on vient de voir en cours

- Découverte de l'environnement de travail pour ce semestre :
  - Langage de programmation : Python
  - IDE : Visual Studio Code (présent sur les machines virtuelles)
- Concepts :
  - Variables et types
  - Opérateurs
  - Appels de fonctions
  - Séquences d'instructions

Bilan

Ce que vous allez faire en séance d'exercices

- Mettre en place l'environnement de travail
- Pratiquer la lecture de petits programmes et le code tracing
- Identifier les types de certaines variables
- Construire des petits programmes utilisant des variables et des appels de fonctions

Séances d'exercices

L'équipe d'assistant.e.s et les salles

Salle	Capacité max	Section	Assistants
BC07	35	MX	Salomé ABOU JAOUDÉ
			Alexandre CARLHAMMAR
			Joshua COHEN-DUMANI
BC08	39	MX	Youssef CHORFI
			Daniel COLLINS
			François DUMONCEL-KESSEL
			Alaa EL OUAHABI
CM1 103	40	GC	Albert FARES
			Yang GAO
			Théophine GURLIE
			Elie HOUEIS
CM1 110	21	GC	Jan JANCZEK
			Marc TOURÉ
CM1 112	36	GC	Julien DELEZ
			Mathilde JANOTTIN
			Ludovic PUJOL