

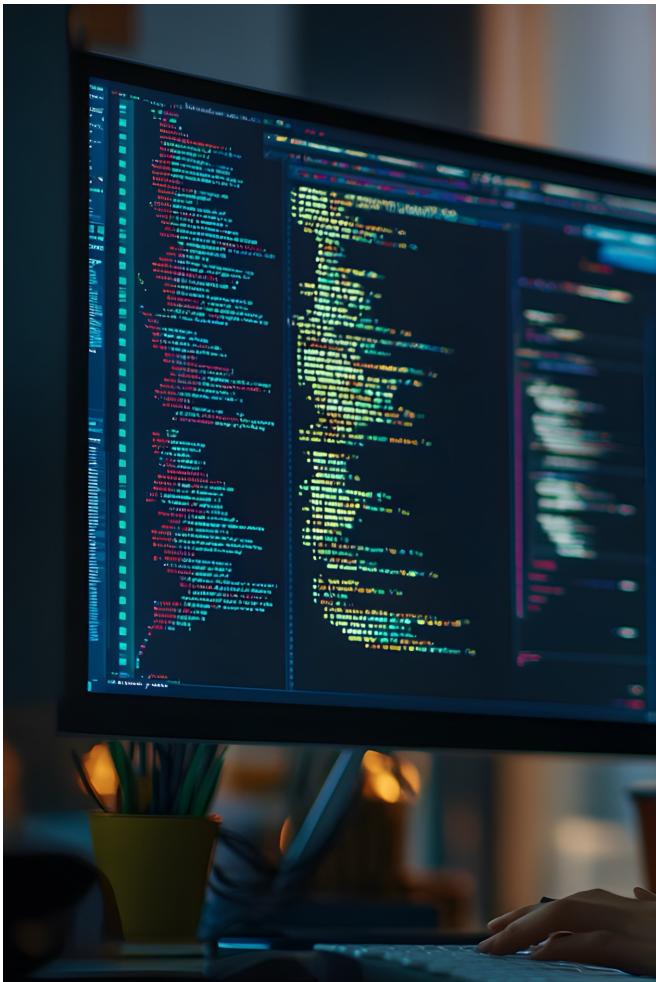
## **Information, Calcul et Communication**

### **CS-119(k) ICC – Programmation Semaine 1**

Rafael Pires

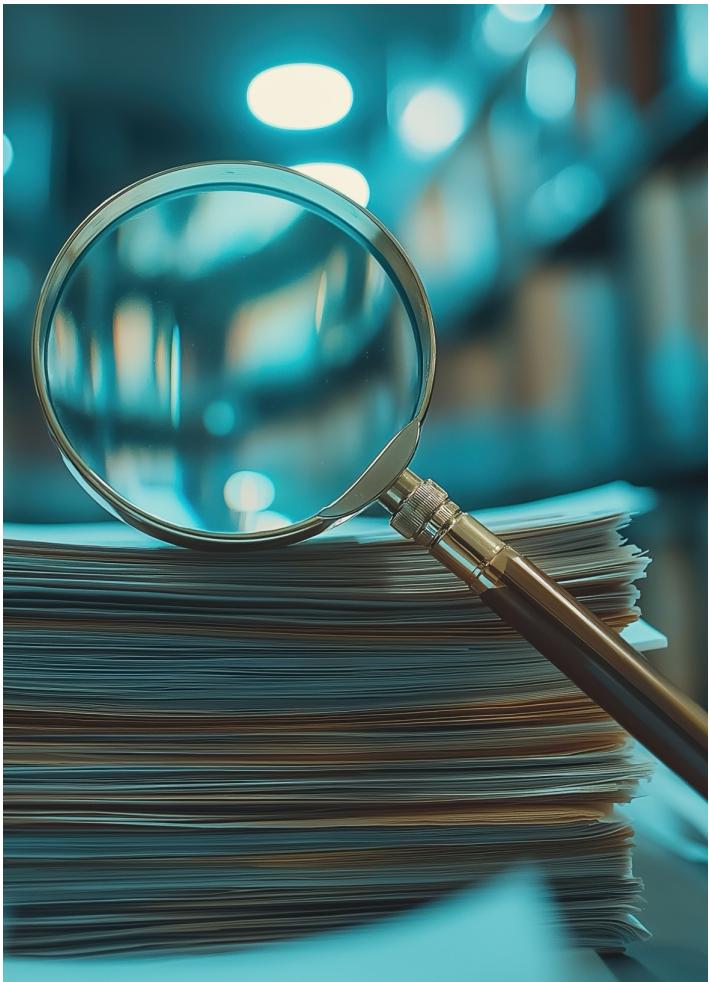
[rafael.pires@epfl.ch](mailto:rafael.pires@epfl.ch)

# Présentation



- **Rafael Pires**
- Licence et Master en Informatique, Master en Mécatronique (Brésil)
- Doctorat en Informatique à l'Université de Neuchâtel, 2020
- EPFL :
  - Postdoc au laboratoire  
Scalable Computing Systems (SaCS)
  - Enseignant depuis l'Automne 2024

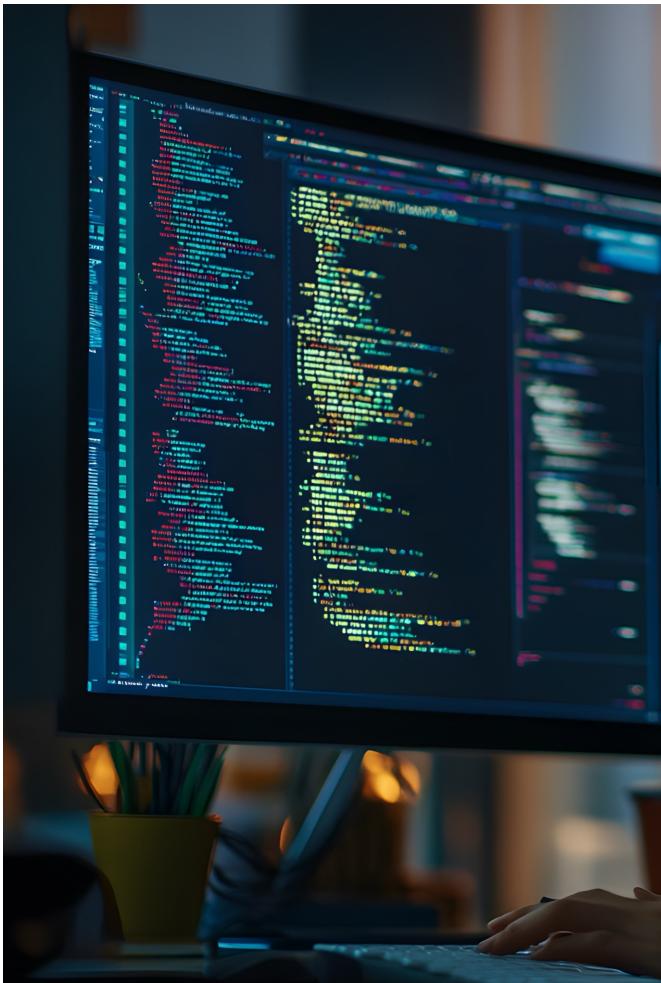
# Documents de cours



- Toutes les informations et liens vers les documents de cours sont sur **Moodle** cours CS-119(k)  
<https://moodle.epfl.ch/course/view.php?id=15817>
- Tous les cours sont **enregistrés**
  - **Enregistrement** à retrouver sur Moodle après
  - Mais... venez au cours autant que possible !
- Pour vos questions : **Ed Discussion**
  - Posez vos **questions** sur le cours, les exercices, le miniprojet
  - **Catégorisez** votre question correctement

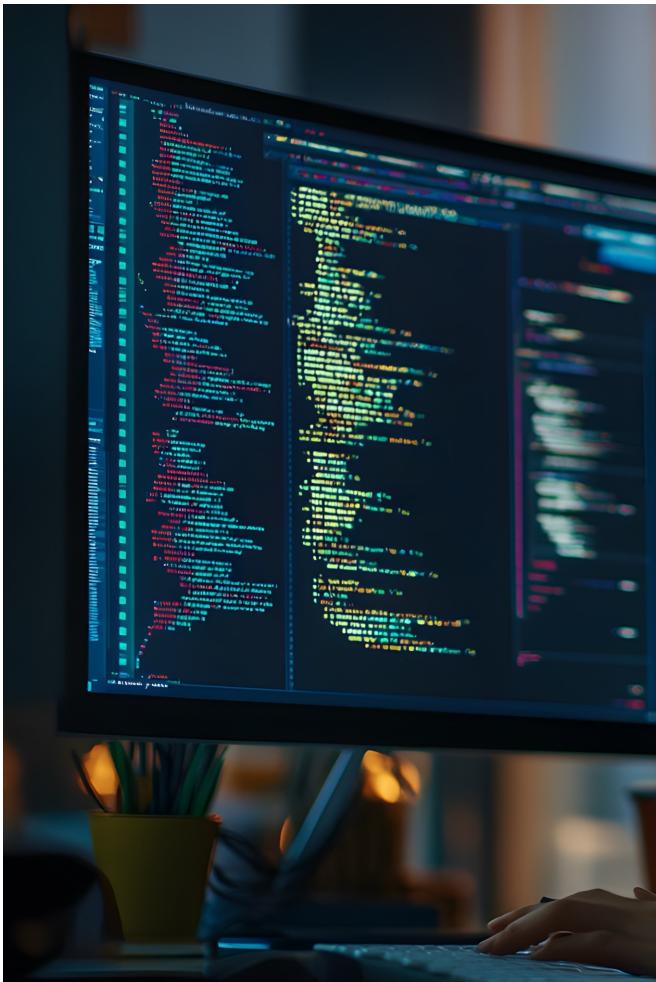
Remerciements : Jean-Philippe Pellet, Olivier Lévêque

# Partie Programmation d'ICC



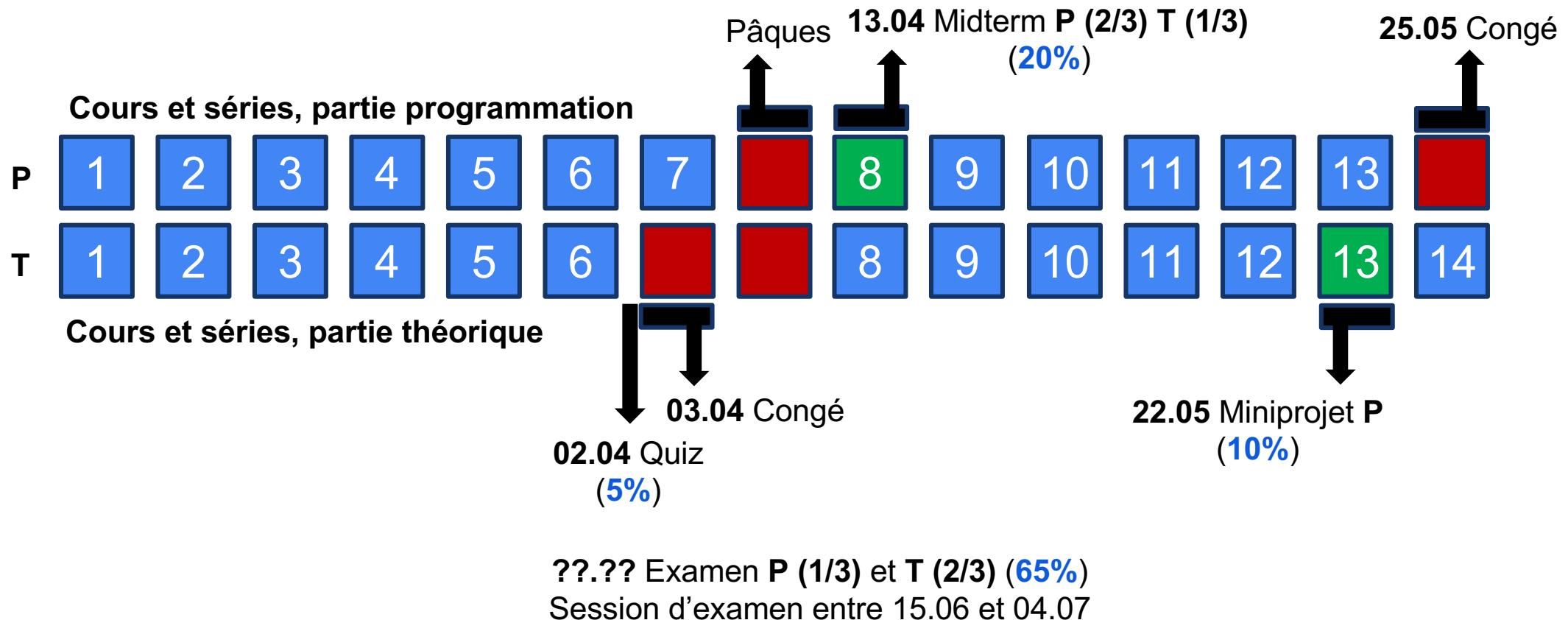
- Approche pragmatique: ***make things work***
- Peu de théorie, davantage de pratique: programmation comme **outil**
- Cible :
  - Concepts les plus importants en **programmation**
  - Bases du langage **Python**
  - Débrouillardise et « savoir-chercher »
- On commence de zéro... mais on avance assez vite

# Exercices



- **Postes de travail virtuels :**
  - Directement en salles **CO020** et **CO021**
  - Sur votre machine via [\*\*https://vdi.epfl.ch\*\*](https://vdi.epfl.ch)
  - Machine virtuelle : **IC-CO-IN-SC-INJ-2026-Spring** (Linux)
- **Autre possibilité : installation personnelle** de Python et Visual Studio Code (IDE) sur votre propre machine
  - Nous vous **aidons** pour l'installation.
  - Batterie et état de marche de votre machine :  
**votre responsabilité.**
  - Nous n'avons pas testé toutes les configurations.

# Programme du cours



# Equipe

**Assistants  
doctorants :**



**Palak**



**Abdellah  
El Mrini**

**Assistant.e.s  
étudiant.e.s :**

Malak Boulahfa  
Majd El Blidi  
Andrew Hajj Assaf  
Hedi Kharouf  
Luc-Faustin Munongo Kazembe

Corentin Sébastien Rouault  
Lina Sadgal  
Omar Zakariya

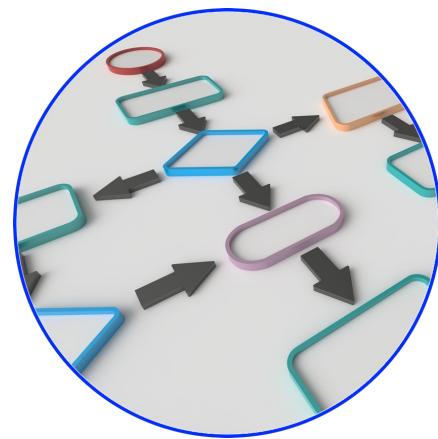
# La programmation



L'ordinateur



Le langage



Les algorithmes

# L'ordinateur



le sèche-cheveux



l'ordinateur

*machine universelle*

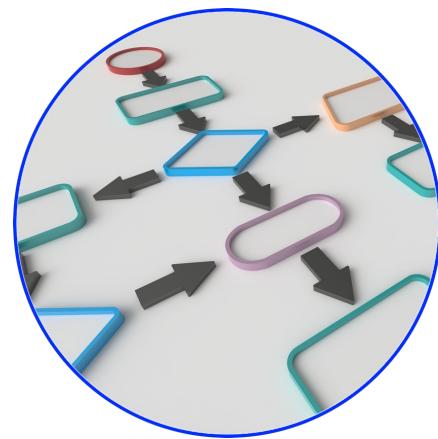
# La programmation



L'ordinateur



Le langage



Les algorithmes

# Python – un language de programmation

«Grammaire»  
**Syntaxe**

Comment mettre  
les mots ensemble  
de manière correcte

**Ensemble de  
règles à savoir**

«Vocabulaire»  
**API/Bibliothèques**

Quels sont les mots qui ont un sens particulier  
dans ce langage, et que signifient-ils exactement?

**En Python: énorme volume de  
bibliothèques**, à apprendre petit à  
petit en fonction des besoins

*Exemples de bibliothèques:* communication réseau,  
manipulation d'images, cryptographie, machine  
learning, manipulation de code, etc., etc.

**Ce cours:** syntaxe de base de Python; exploration de la bibliothèque standard de  
Python et outils pour rechercher et utiliser des bibliothèques existantes



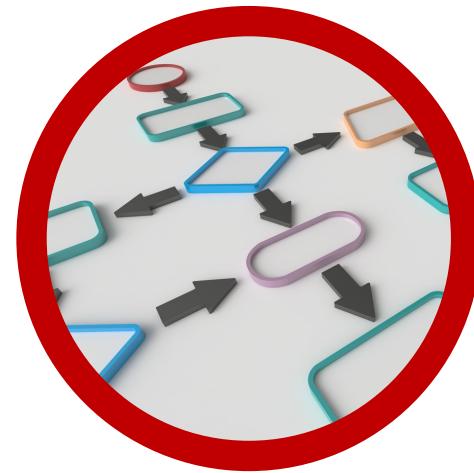
# La programmation



L'ordinateur



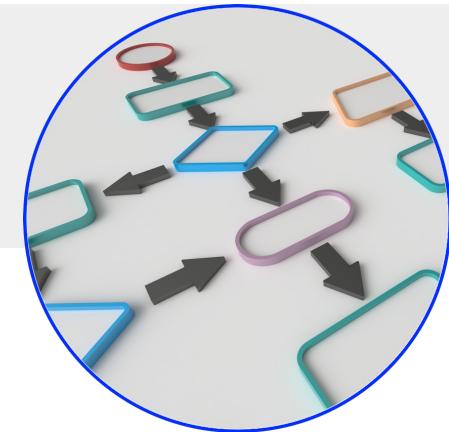
Le langage



Les algorithmes

# Écrire un programme

- Comme une **recette** (que vous écrivez)
  - Une série d'instructions à exécuter dans un certain ordre
  - Une instruction se composent de mots du «vocabulaire» du langage, assemblés selon la syntaxe du langage («grammaire»)
  - *L'ordre des instructions est bien sûr important !*



# La programmation

***l'ordinateur:***  
***la machine universelle du monde de l'information***

***la programmation:***  
***une technique de communication structurée avec la machine;***  
***l'art d'exprimer de façon élégante un processus***

***l'approche computationnelle:***  
***un «nouveau» moyen d'opérer en tant que scientifique***

traitement des signaux

simulation de modèles  
(météo, cerveau, ...)

automatisation de tâches répétitives

acquisition de données (capteurs)

interfaces graphiques

calcul numérique

bases de données (enquêtes)

# Faire des calculs en Python

Démo

```
side = 4  
area = side * side  
  
print(area)
```

# Faire des calculs en Python

Variante :

```
side: int = 4
area: int = side * side

print(area)
```

- Déclarer les types est en principe optionnel en Python
- Mais : faites-le autant que vous pouvez
  - Plus de vérifications par le compilateur (ou linter)
  - Code plus expressif et plus facile à (re)lire pour vous
  - Force à mieux réfléchir à ce qu'on écrit

# Faire des calculs en Python

```
side: int = 4
```

nom      type      valeur

«Prends un bout de mémoire, rappelle-toi que je vais y faire référence avec le nom side, et stockes-y la valeur 4, sachant que c'est un nombre entier»

```
area: int = side * side
```

«Prends un autre bout de mémoire, que je vais appeler maintenant area, et stockes-y le produit de ce qui est dans l'emplacement mémoire qui s'appelle side avec lui-même, sachant que le résultat est un nombre entier.»

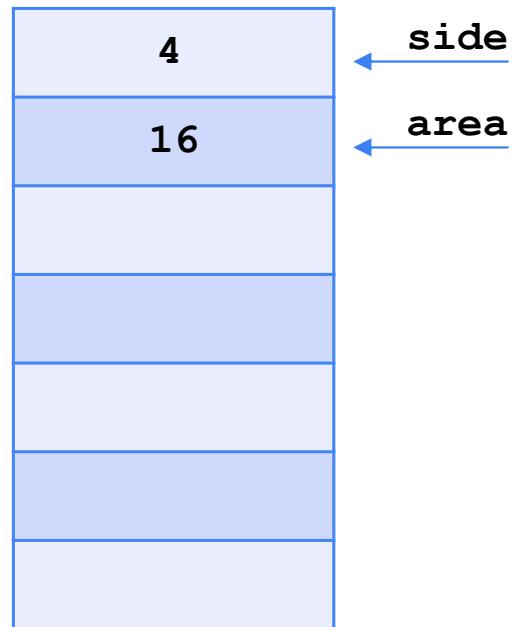
```
print(area)
```

«Imprime sur la console ce qu'il y a dans l'emplacement mémoire que j'ai appelé area.»

# Représentation de la machine



Mémoire vive  
→



```
side: int = 4
area: int = side * side
        4      * side
        4      * 4
                16
```

# Afficher le contenu d'une variable avec un message

```
side: int = 4
area: int = side * side

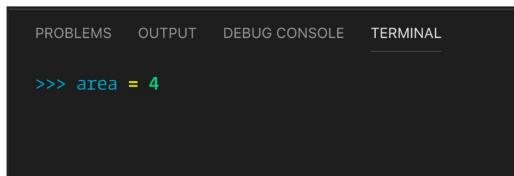
print("l'aire vaut", area)
print(f"l'aire vaut {area}")
print("l'aire vaut {}".format(area))
print("l'aire vaut " + str(area))
print("l'aire vaut %d" % area)
```

# Faire tourner du code

## Interpréteur

Menu **Terminal** → **New Terminal** → tapez **python3**

**Puis:** taper des lignes dans le terminal qui s'ouvre en bas à droite

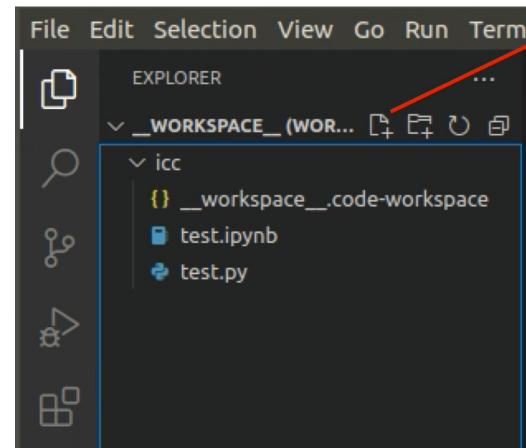


```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL  
>>> area = 4
```

Terminez par **Ctrl-D**

Exécution **ligne par ligne**

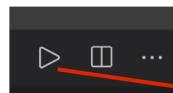
## Via fichier



Bouton New File...

Puis: donner un nom qui se termine par .py

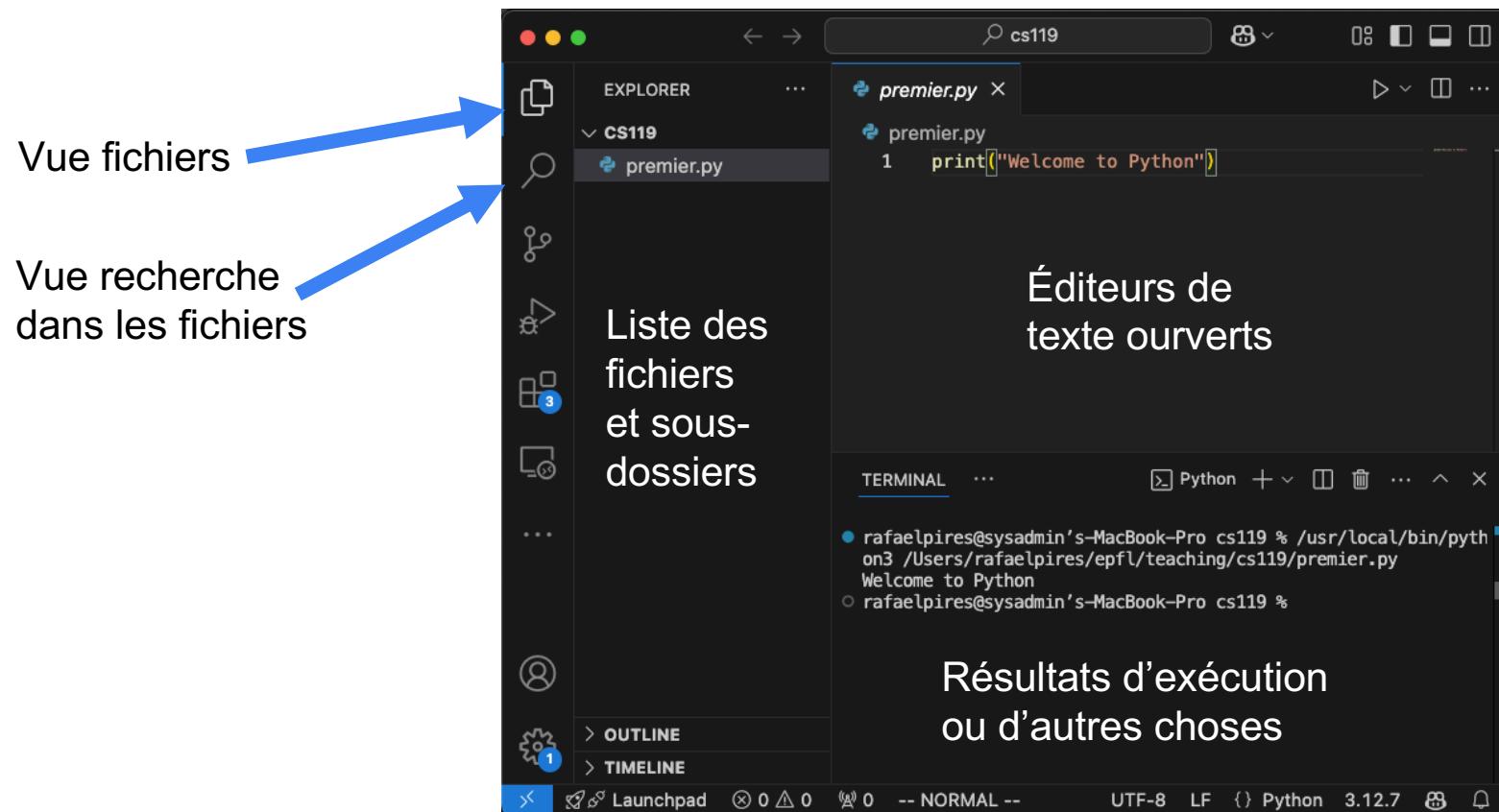
**Puis:** éditer le fichier dans l'éditeur qui s'ouvre à droite



Bouton Run en haut à droite

Exécution de **tout le fichier**

# Visual Studio Code



# Les principaux types de données

```
side: int = 4  
  
length: float = 3.5  
  
my_name: str = "Rafael"  
  
my_last_name: str = 'Pires'  
  
# Commentaire
```

int — Un **nombre entier**

float — Un **nombre à virgule**

str — Du **texte**; une *chaîne de caractères* (string = chaîne), définie avec " ou ' au début et à la fin

*Pas un type de données, mais un **commentaire** à vous, pas interprété comme code*

# Conversion float / int / String

*On ne peut pas faire toutes les opérations avec tous les types, mais on peut convertir d'un type à un autre*

```
# Conversion depuis un int vers un float ou vers du texte
some_int: int      = 34
some_int_as_float = float(some_int) # 34.0, mais souvent inutile en Python
some_int_as_string = str(some_int)   # "34"

# some_int_as_string + 2, ne marche pas! On ne peut pas ajouter un nombre à du texte
# some_int_as_string + "2", on peut "ajouter" deux str: concaténation

# Conversion depuis un float
import math
some_float: float      = 0.182
some_float Rounded_up  = math.ceil(some_float) # 1
some_float Rounded_down = math.floor(some_float) # 0
some_float_as_int       = int(some_float)       # 0
```

# Appel de fonctions

Forme générique :

```
nom_de_fonction(argument)  
print(some_variable)
```

Depuis une bibliothèque :

```
import math  
math.ceil(some_float)
```

Méthodes d'objet :

```
some_string.upper()
```

# Autres manipulations utiles

```
my_string = "programmation"
# Vous choisissez le nom de la variable;
# la valeur est toujours entre "" ou ''

# la fonction len() retourne la longueur d'un string : 14
length = len(my_string)

# la méthode upper() s'écrit après un point et
# crée une version tout en majuscules de la valeur
# indiquée avant le point : PROGRAMMATION
my_string_upper = my_string.upper()

# le slicing (indexage d'une variable entre [])
# permet d'extraire une partie du string : « rog »
my_substring = my_string[1:4]
```

# Se documenter sur Python



**Documentation officielle**  
<https://docs.python.org/3.10/>



« **python convert int to string** »  
« **python get string lenght** »  
« **python check if string contains other string** »



<https://stackoverflow.com/>

Site spécialisé en programmation, questions avec réponses triés par ordre de pertinence selon votes de la communauté.

# IA générative



ChatGPT



EPFL



deepseek



Claude



CURSOR



Gemini



Copilot



Qwen

# Résumé Cours 1 – ICC-P

- Python est un langage moderne **avec une syntaxe minimale**
- VS Code est un IDE pour Python (notamment) qui permet **d'éditer** les fichiers et **d'exécuter** le programme
- **L'interpréteur** interactif permet de facilement tester de petits bouts de code
- En Python, on peut **déclarer le type** des variables. Les types aident à vérifier que le programme est correct
- Des **notations précises** permettent de calculer de nouvelles valeurs  
(fonctions, méthodes, slicing — **pas de panique, on en reparlera !**)

[rafael.pires@epfl.ch](mailto:rafael.pires@epfl.ch)

**EPFL**

Merci

