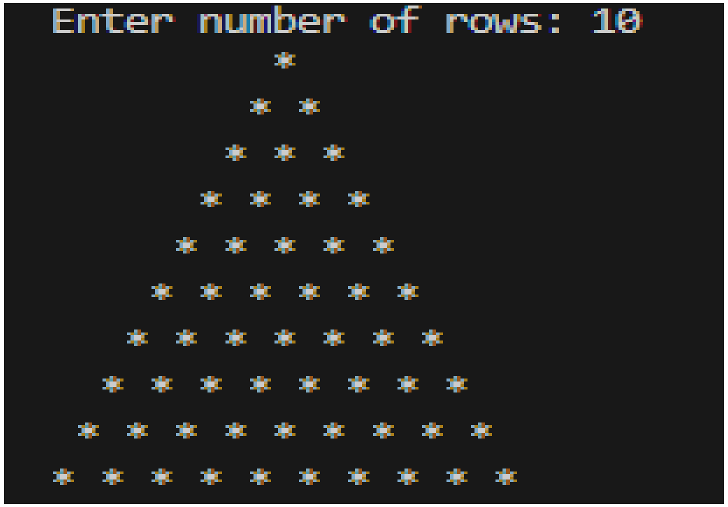
Project Pyramid

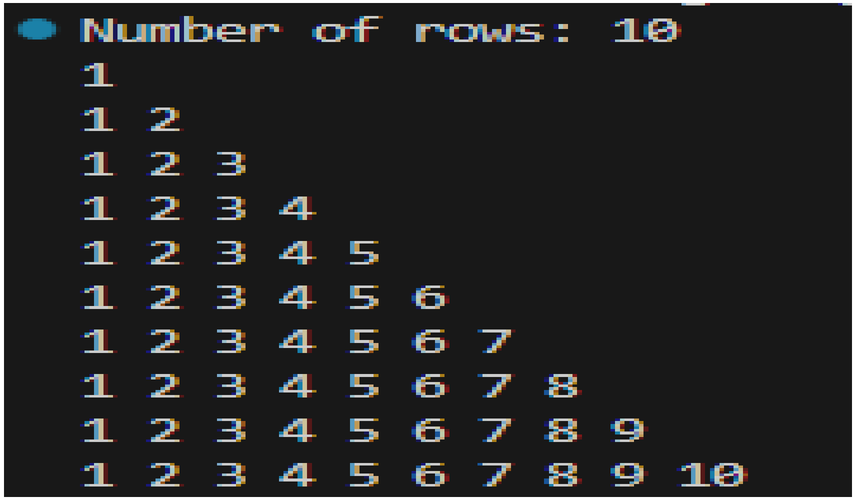
A program that calculates and displays a representation of a pyramid. The program will ask the user for the number of levels of the pyramid and then generate a visual representation.



```
Enter number of rows: 10
         *
        * *
       * * *
      * * * *
     * * * * *
    * * * * * *
   * * * * * * *
  * * * * * * * *
 * * * * * * * * *
* * * * * * * * * *
```

Project Right angle Triangle Pattern of Numbers

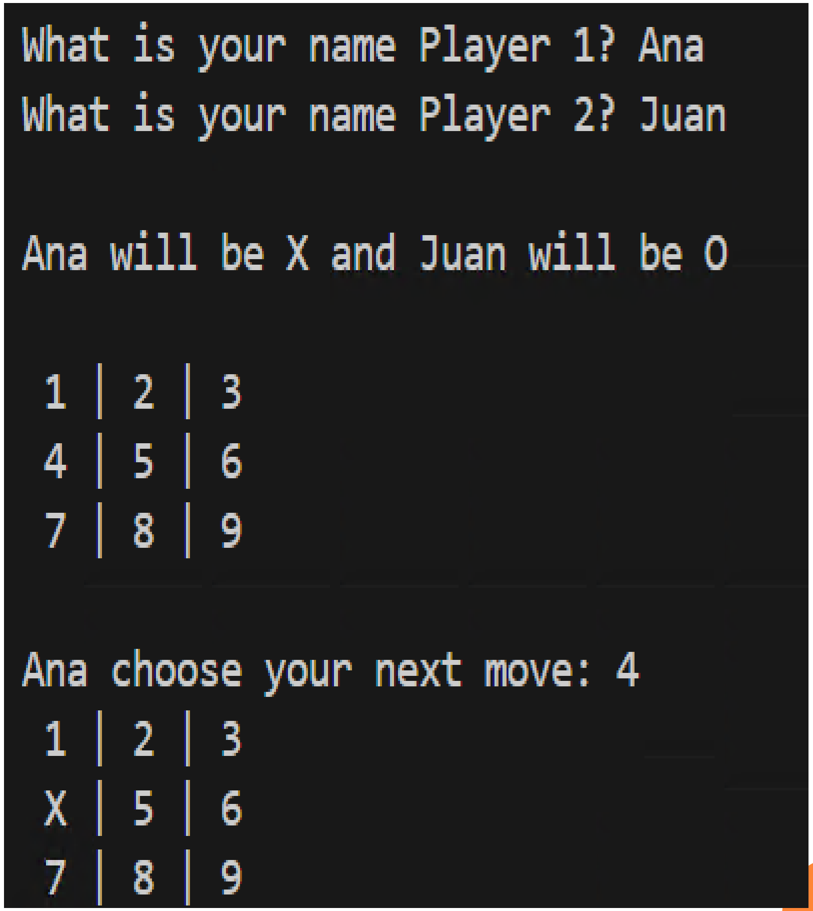
Program to display right angle triangle of numbers where user is asked for the number of rows. Display number pattern as shown in the figure below.



```
● Number of rows: 10
  1
  1 2
  1 2 3
  1 2 3 4
  1 2 3 4 5
  1 2 3 4 5 6
  1 2 3 4 5 6 7
  1 2 3 4 5 6 7 8
  1 2 3 4 5 6 7 8 9
  1 2 3 4 5 6 7 8 9 10
```

Project Tic Tac Toe

A program that allows two players to play Tic Tac Toe against each other



```
What is your name Player 1? Ana
What is your name Player 2? Juan

Ana will be X and Juan will be O

 1 | 2 | 3
 4 | 5 | 6
 7 | 8 | 9

Ana choose your next move: 4
 1 | 2 | 3
 X | 5 | 6
 7 | 8 | 9
```

Computational Thinking

Problem-solving approach that helps you to think **logically** and solve problems **effectively**.

Computational Thinking

Decomposition:

Breaking down a complex problem into smaller, more manageable sub-problems or tasks.

Computational Thinking

Pattern Recognition:

Identifying recurring patterns or similarities in different problems to develop generic solutions.

Computational Thinking

Abstraction:

Focusing on essential details while ignoring irrelevant information to simplify the problem.

Computational Thinking

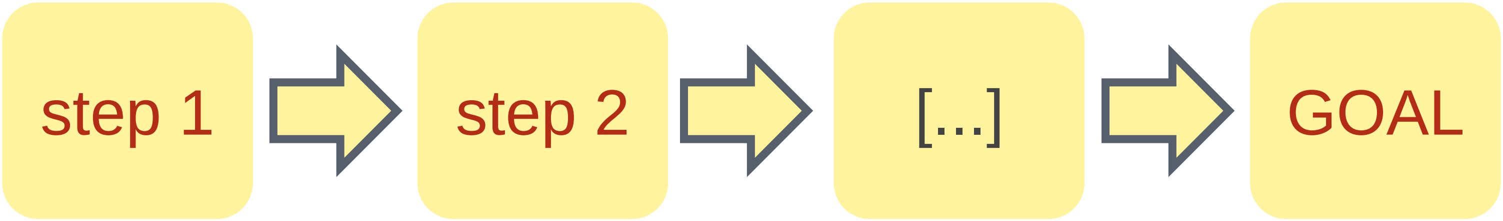
Evaluation and **Optimization**:

Evaluate the efficiency and effectiveness of solutions and refining them for better performance.

Algorithm

An Algorithm is an ordered **series of steps** that leads to the achievement of a goal or the solution to a problem.

Algorithm



Computational Thinking

It is not limited to computer science but can be applied to various disciplines and **real-world challenges**.

Computational Thinking

Make your sandwich

Decompose

step by step

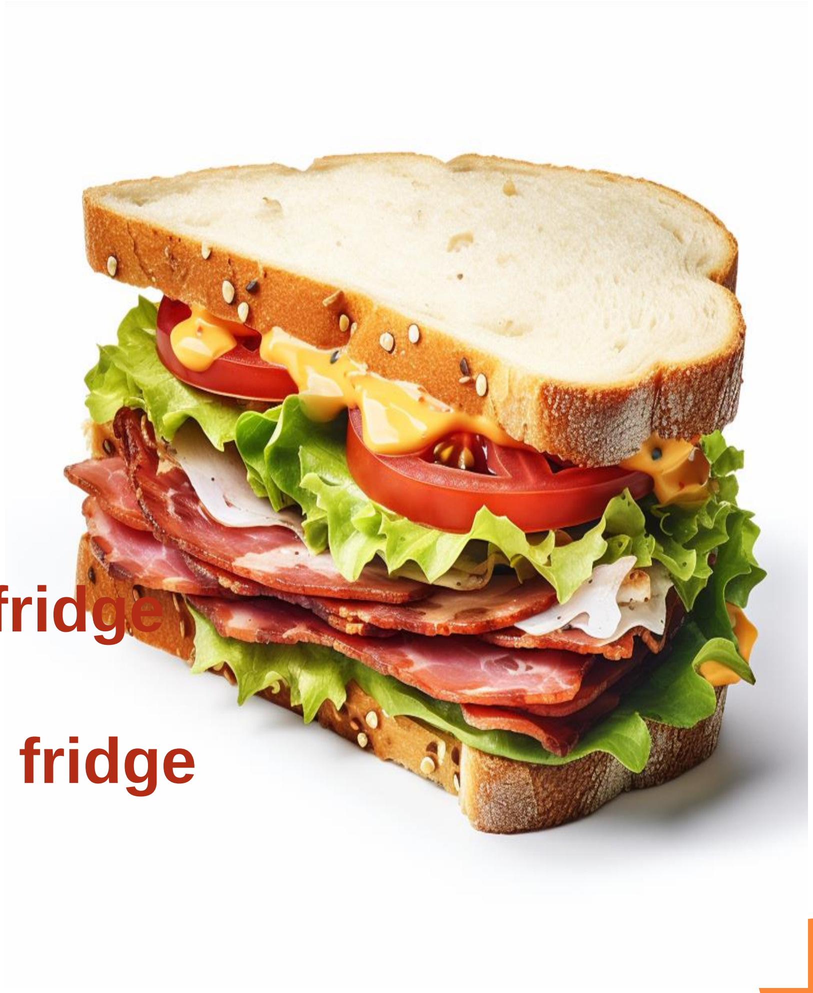
Computational Thinking

1. **Prepare the ingredients**
2. **Assemble the ingredients**
3. **Plate it**

Concept: Decomposition and Abstration

Computational Thinking

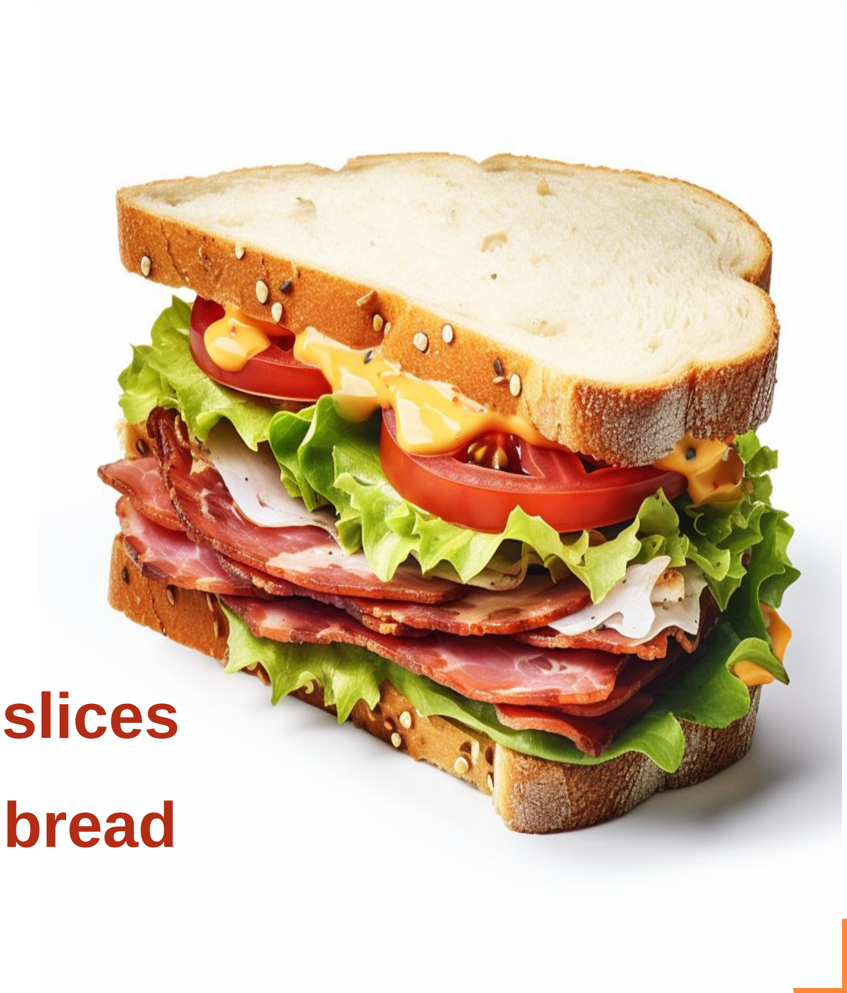
1. Prepare the ingredients
    1. **Cut the slices of bread**
    2. **Cut tomato**
    3. **Take the ham out of the fridge**
    4. **Take the salad out of the fridge**



Computational Thinking

1. Prepare the ingredients
2. Assemble the ingredients

1. **Slice of bread**
2. **Salad**
3. **Ham**
4. **Tomato slices**
5. **Slice of bread**



Computational Thinking

1. Prepare the ingredients
2. Assemble the ingredients
3. Plate it
    1. **Put it on a plate**

Computational Thinking

1. Prepare the ingredients
2. Assemble the ingredients
3. Plate it

    1. **Put it on a plate**

Concept: Algorithm

Computational Thinking

Now, make a Lasagna

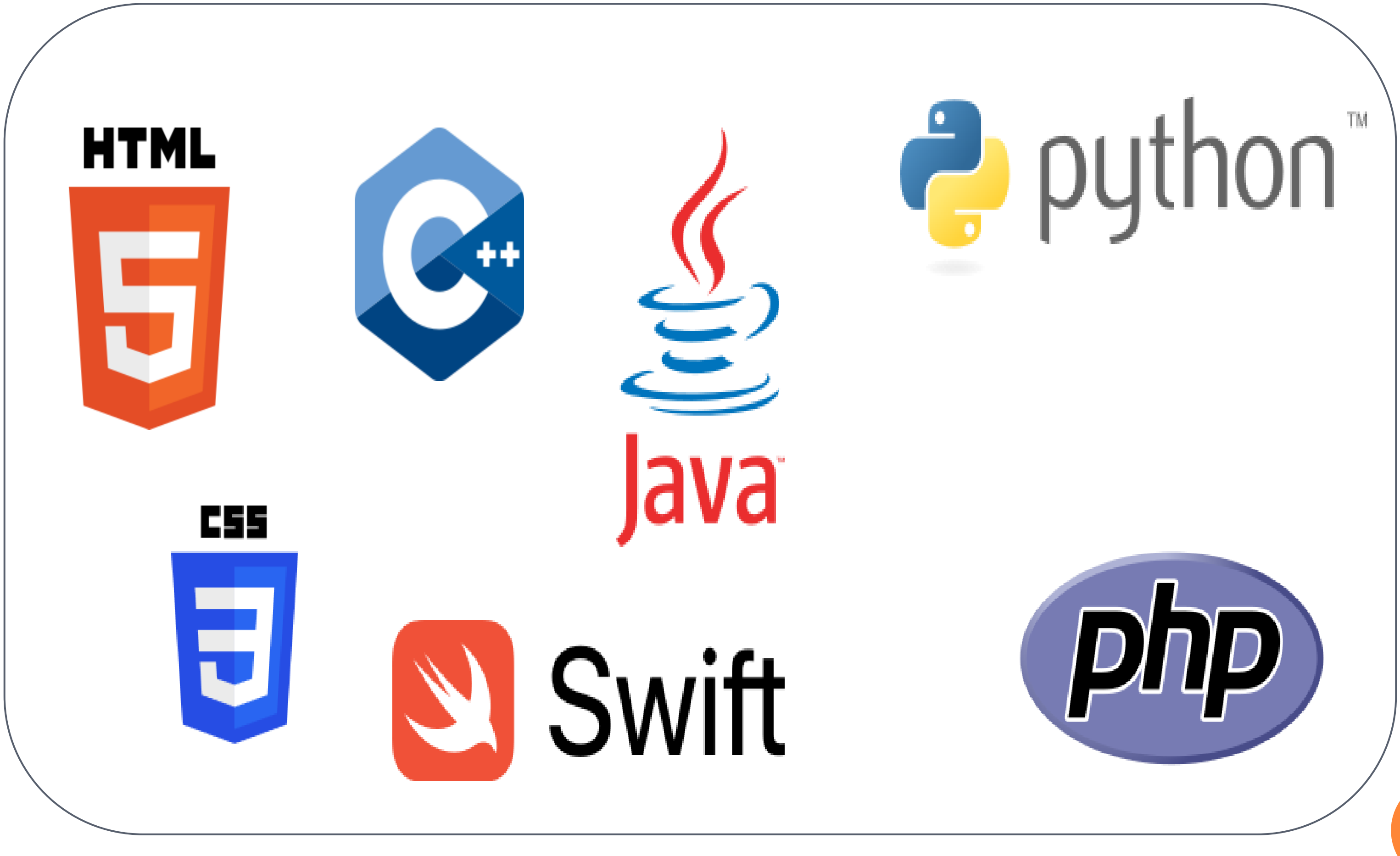
1. Prepare the ingredients
2. Assemble the ingredients
3. Cook it in the oven

Concept: Pattern

Programming Language

is a language that humans can use and that the machine can understand.

Programming Language



Python

One of the **most used** programming language. For example, it’s used when **analysing data** or to create **softwares**.

python™
NETFLIX
YouTube
Quora
Google
amazon
Instagram
IBM
Spotify
Dropbox
NASA
UBER
facebook

Code Editor

To develop in a programming language, you need a code editor that understands and executes the language. In this course we will mainly use a very common one:

Visual Studio Code

Output

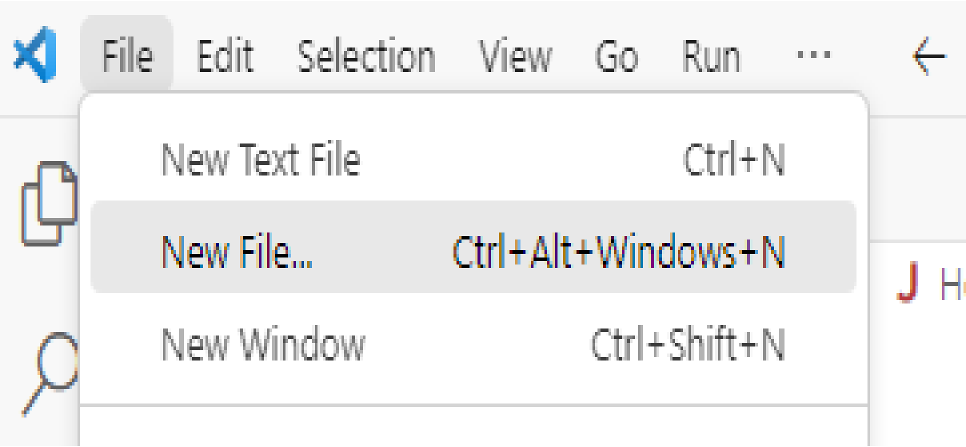
```
Print(«Hello world !»)
```

Code Editor

Create a new program

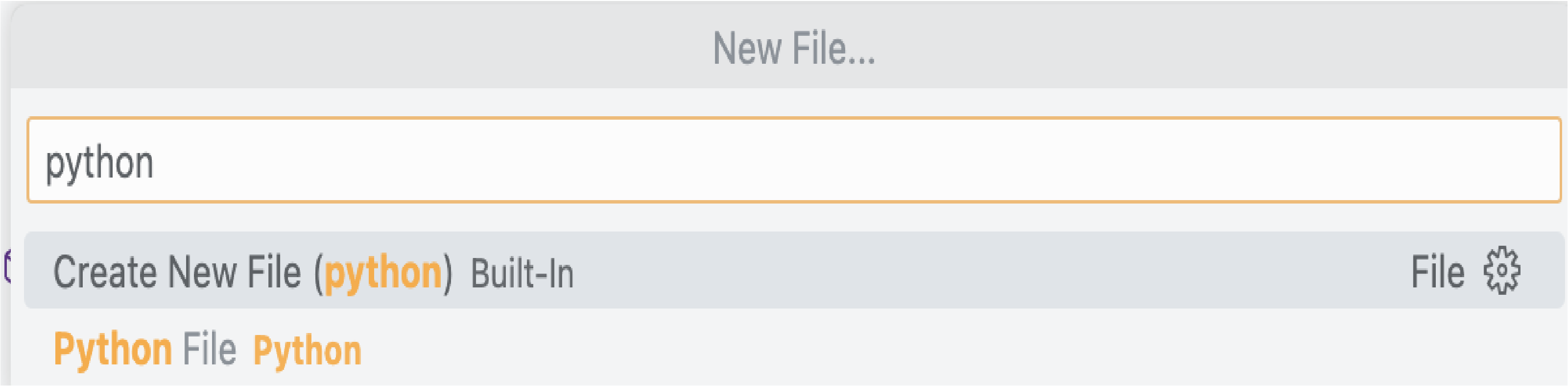
Code Editor

Step 1: **Create** a file.



Code Editor

Step 2: Create New File (**python**) Build-in



New File...

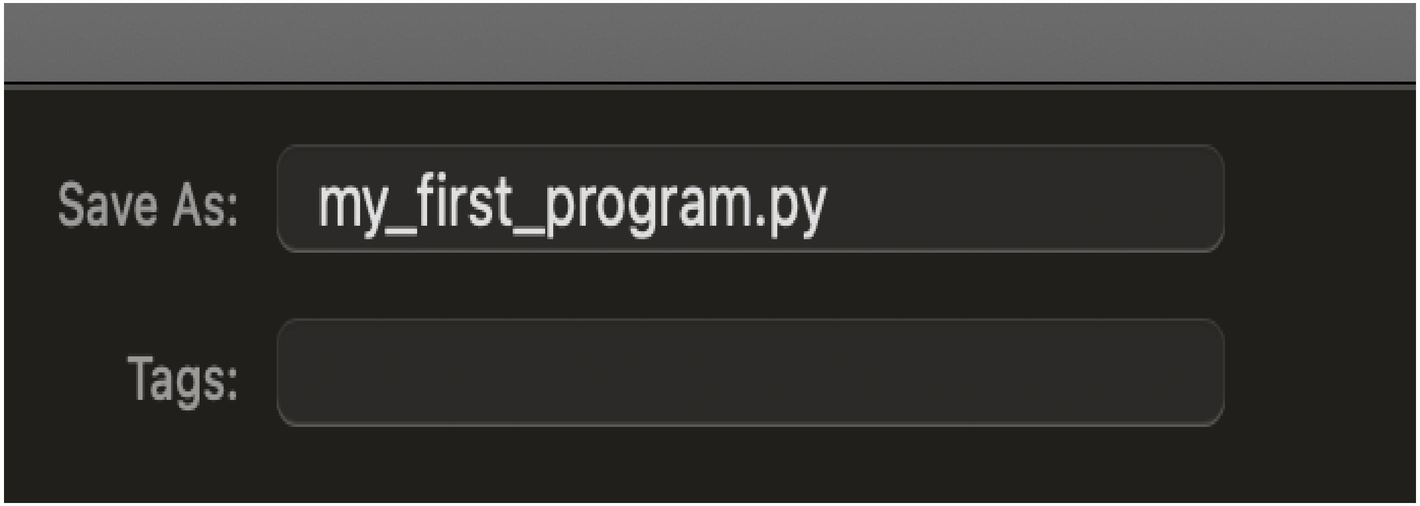
python

Create New File (python) Built-In

Python File Python

Code Editor

Step 4: **Name** of your file



my_first_program.py

Let’s create your first

Hello world !

```
print(‘HelloWorld !’)
```

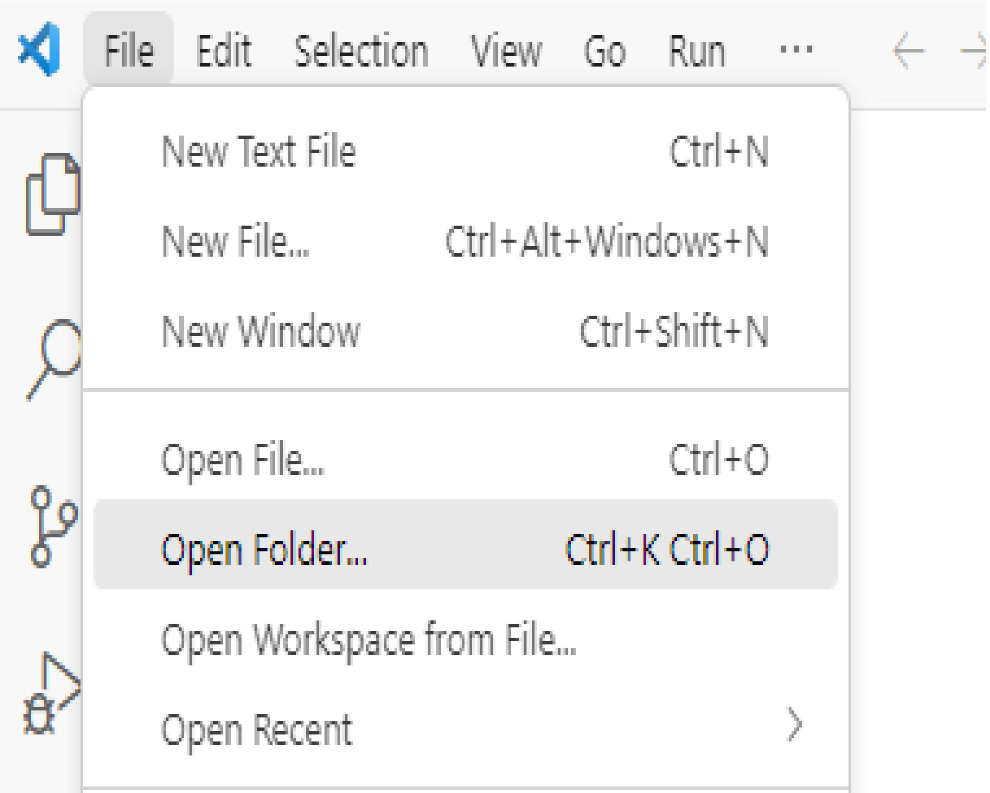
Code Editor

Open and run an existing program

Code Editor

Step 1: **Open** the main folder.

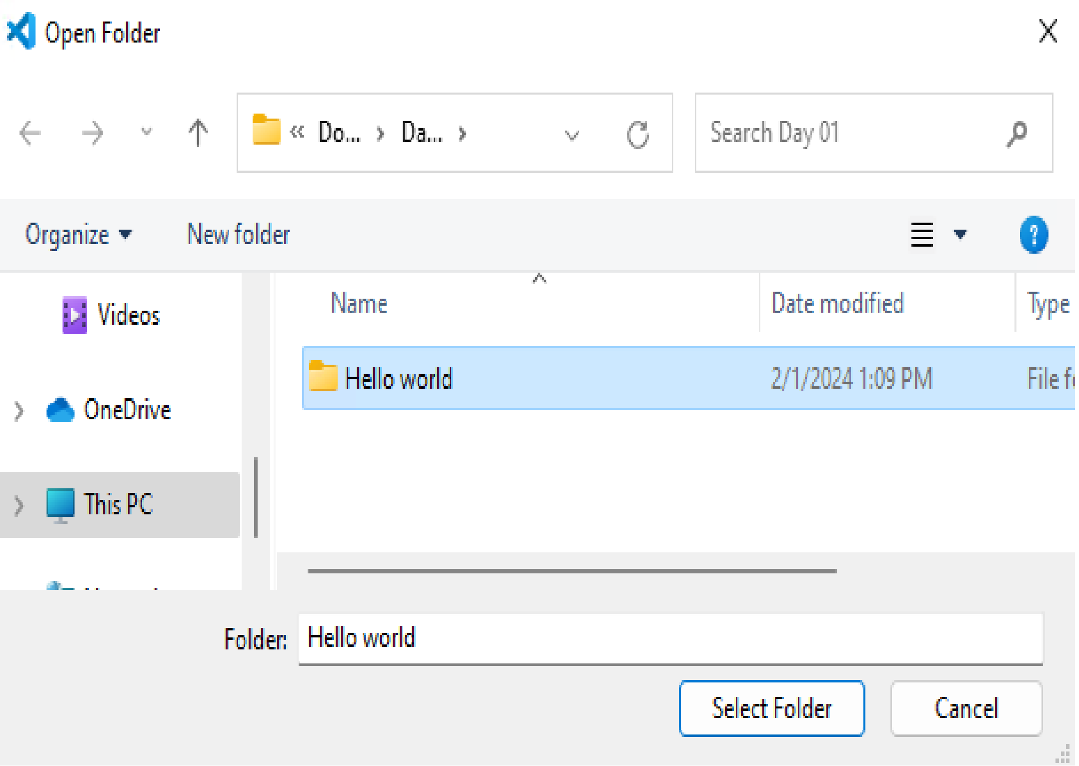
It w



Code Editor

Step 2: **Choose** the main folder.

It

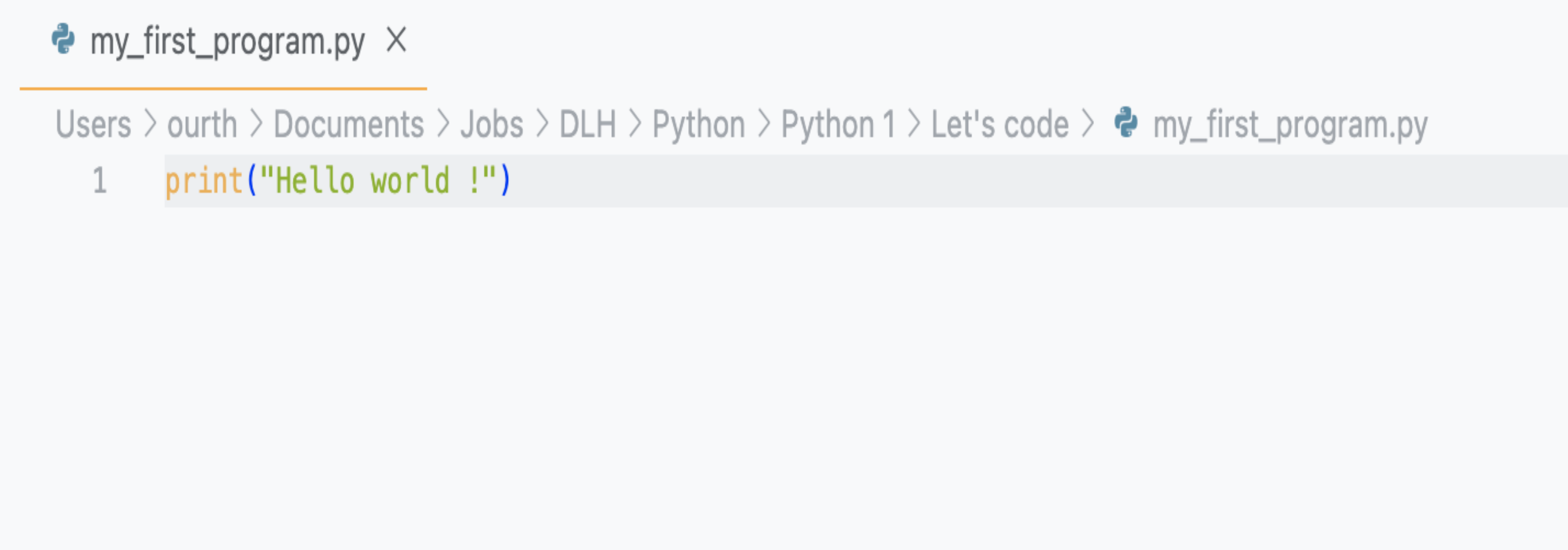


Name	Date modified	Type
Hello world	2/1/2024 1:09 PM	File f

Folder: Hello world

Code Editor

Ste



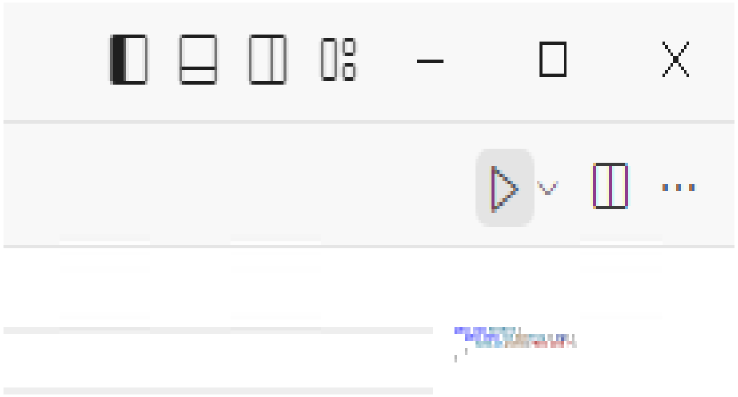
my_first_program.py

Users › ourth › Documents › Jobs › DLH › Python › Python 1 › Let's code › my_first_program.py

```
print("Hello world !")
```

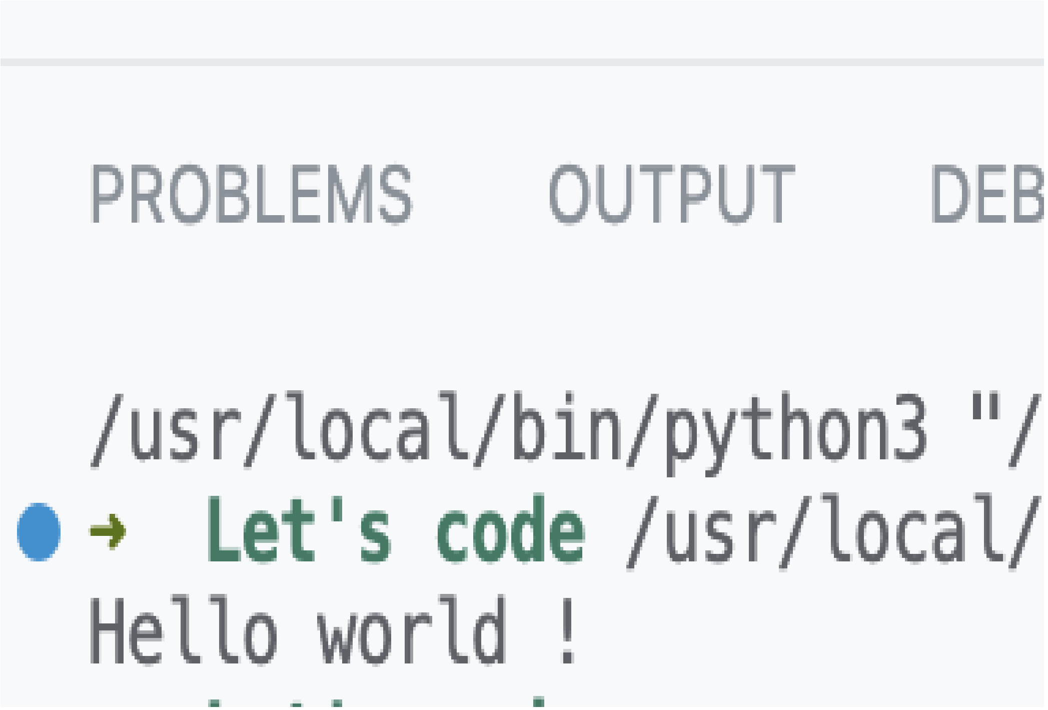
Code Editor

Step 3: **Run** the program



Code Editor

Step 4: See the **output**



```
/usr/local/bin/python3 "/
● ➜  Let's code /usr/local/
Hello world !
```

Python syntax rules

Python is case-sensitive!

Print() is not print()

Python syntax rules

Line continuation

```
sum =  123 + \
        456 + \
        789
```

Python syntax rules

Line comments

```
# This is a comment

'''
And this is a multi-line comment
'''


""" This is also a multi-line comment """
```

Python syntax rules

Code indentation

THE MOST IMPORTANT RULE !!!

Use tab for indentation, this delimits the block of code