

CSE 101

Computer Science Principle

Lecture 01: Introduction

March. 2023

Prof. Francois Rameau

francois.rameau@sunykorea.ac.kr



Introduction



Francois Rameau

Education

| | |
|---|---|
| <p>2011-2014 PhD PhD in Computer Vision Le2i University of Burgundy</p> | <p>2016-2022 Research Associate Professor RCVlab, KAIST Daejeon, South Korea</p> |
| <p>2010-2011 Master 2 Master VIBOT University of Burgundy Erasmus Mundus Master</p> | <p>2015-2016 Post-doc RCVlab, KAIST Daejeon, South Korea</p> |
| | |
| | <p>2023-Present Assistant Professor SUNY South Korea</p> |

Research interests

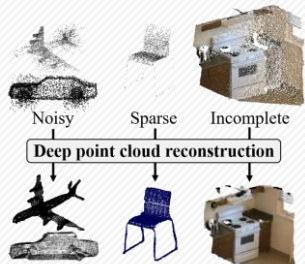
-  3D Reconstruction
-  Machine Learning
-  Connected Cars



Some of my research expertise

3D Point cloud processing

[ICLR'22, ECCV'22-submitted]



3D reconstruction

[ECCV'16, ICCV'21, ICCV'21, URAI'15]

All-around Depth from Small Motion with a Spherical Panoramic Camera

European Conference on Computer Vision (ECCV), 2016

SungHoon Im, Hyewon Ha, Francois Fleuret, Hae-Gon Jeon, Gyeongmin Choi and In So Kweon
Robotics and Computer Vision Lab, KAIST, Korea

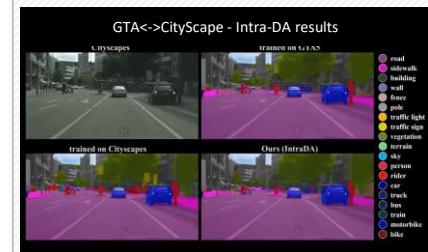
Visual Tracking

[ICCV'17, ECL'11, ICCVW'11, PSIVT'15]



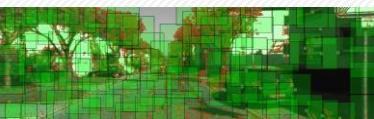
Domain adaptation

[CVPR'20]



Low-level vision

[PRL'18, IROS'19]



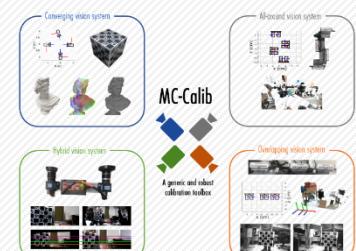
Computational photography

[AAAI'21, AAAI'21, CVPRW'21]



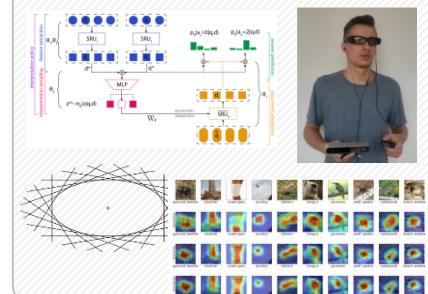
Camera Calibration

[PAMI'22, JAS'22, CVIU'22, ACCV'12]



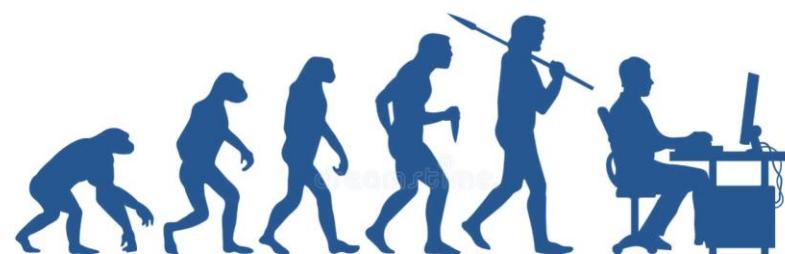
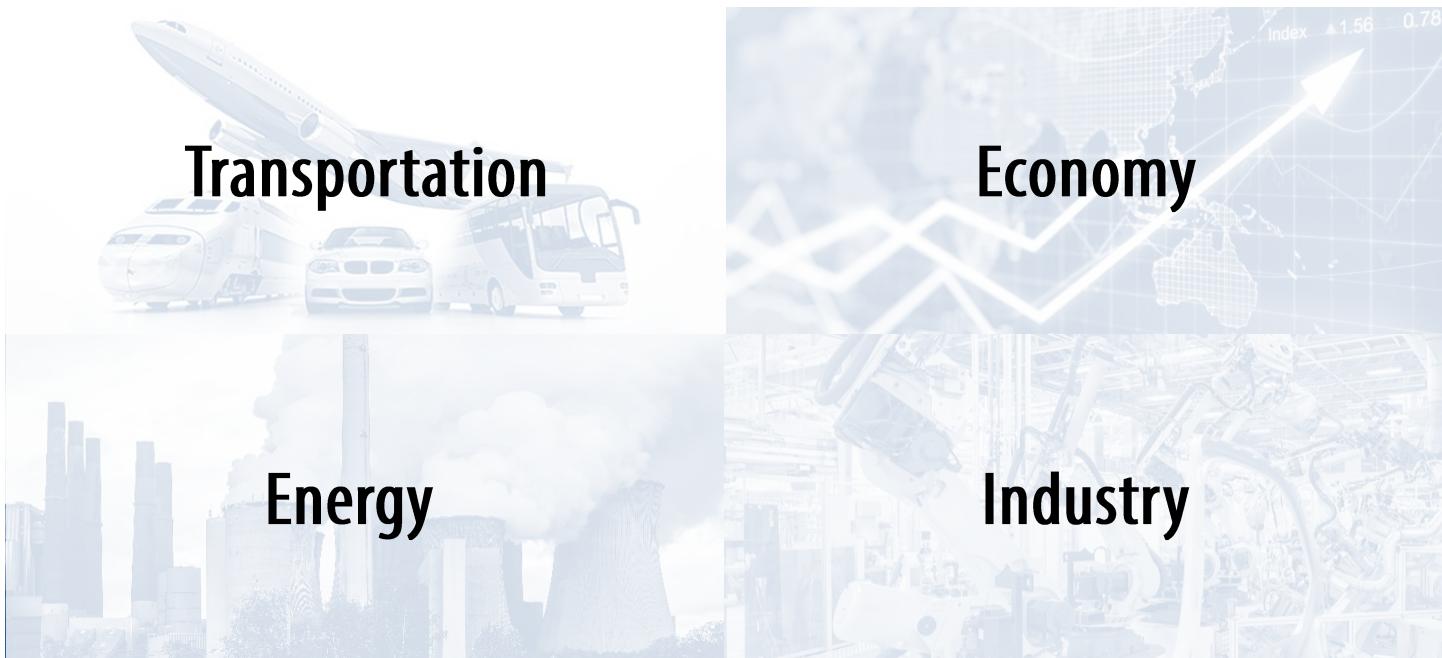
And more

[PAMI'21, TASLP'19, TCG'16, etc.]



Computer science

Computer science is powering our entire society



General information



For CS majors

- *This course is the starting point of your CS career!*



For non-CS majors

- *This course is an excellent way to get an introduction to what computer science is all about and learn how to program*
- *Many non-CS jobs require or benefit from some programming: Data scientist, researchers, project manager, designer, most jobs at technology companies*

This course assumes no background knowledge of programming

Overview

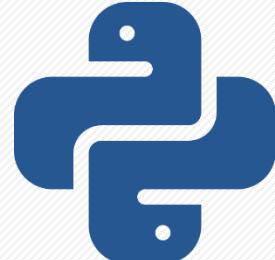
- CSE 101
 - Introduces the important, central ideas of computer science
 - Explores computational thinking and problem solving
 - Covers the fundamentals of computer programming
- Computer science is the *study of problem-solving with computers*
 - Astronomers don't study telescopes. They *use* telescopes to study the stars!
 - Likewise, computer professionals *use* computers to solve important problems in the modern world
 - Computer scientists also build computers and software that makes the computer run.
- An important thread of this course is **computational thinking**, which is the way computer scientists think about and solve problems

Major topics

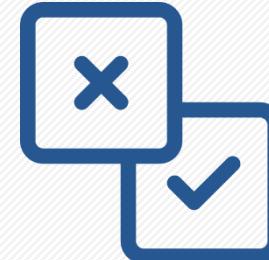
Computational thinking & problem solving



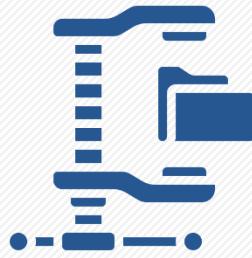
Fundamental programming in Python



Boolean logic



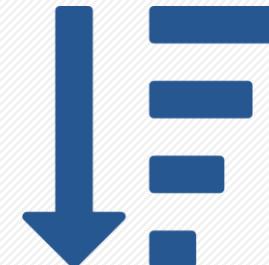
Data representation & compression



Social, legal & ethical issues in computing



Basic algorithms for searching and sorting



Additional topics may be covered and this list may be modified based on interest and time

Course information

CSE 101: Computer Science Principles



| Tue | Wed | Thu |
|------------------------|--------------------|------------------------|
| Lecture 2:00-3:20PM | Lab 2:00-3:20PM | Lecture 2:00-3:20PM |

Spring 2023: ~14 weeks

● Holidays

● Finals

| MARCH | | | | | | |
|-------|----|----|----|----|----|----|
| S | M | T | W | T | F | S |
| 1 | 2 | 3 | 4 | 5 | 6 | |
| 7 | 8 | 9 | 10 | 11 | 12 | 13 |
| 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| 21 | 22 | 23 | 24 | 25 | 26 | 27 |
| 28 | 29 | 30 | 31 | | | |

| APRIL | | | | | | |
|-------|----|----|----|----|----|----|
| S | M | T | W | T | F | S |
| | | | | 1 | 2 | 3 |
| 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 11 | 12 | 13 | 14 | 15 | 16 | 17 |
| 18 | 19 | 20 | 21 | 22 | 23 | 24 |
| 25 | 26 | 27 | 28 | 29 | 30 | |

| MAY | | | | | | |
|-----|----|----|----|----|----|----|
| S | M | T | W | T | F | S |
| | | | | 1 | | |
| 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| 16 | 17 | 18 | 19 | 20 | 21 | 22 |
| 23 | 24 | 25 | 26 | 27 | 28 | 29 |
| 30 | 31 | | | | | |

| JUNE | | | | | | |
|------|----|----|----|----|----|----|
| S | M | T | W | T | F | S |
| | | 1 | 2 | 3 | 4 | 5 |
| 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| 13 | 14 | 15 | 16 | 17 | 18 | 19 |
| 20 | 21 | 22 | 23 | 24 | 25 | 26 |
| 27 | 28 | 29 | 30 | | | |

Teaching staff & website

Instructor



Prof. Francois Rameau

✉ : francois.rameau@sunnykorea.ac.kr

🏢 : Office B423

☎ : +82.32.626.1230

Office Hours: M 10:00 – 11:45

W 10:00 – 11:45

Teaching assistants



Shubhangi Garnaik

✉ : shubhangisaile.garnaik@stonybrook.edu

Office Hours: Thu 18:00 -22:00



Yelin Cheon

✉ : yelin.cheon@stonybrook.edu

Office Hours: Tues 18:00 – 23:30



Inuk Baik

✉ : inuk.baik@stonybrook.edu

Office Hours: Wed 17:00 – 22:30



Website:

<https://rameau-fr.github.io/suny-cse101/>

E-mail policy



- Title should include the tag [CSE101]
- Please introduce yourself
- Add the TAs in Cc
- I cannot guarantee timely response (but I will try my best)
- Before asking, please try to resolve it yourself. Google and StackOverflow are your best friends!

Acknowledgments

Lecture developed with the support of previous material by:



Prof. Alex Kuhn



Prof. Byungkon Kang

Grading

Quizzes



Weight: 30%
Quantity: ~ 5
Duration: 60-90min

Attendance



Weight: 10%
80% attendance expected

Homeworks



Weight: 30%
Quantity: ~ 5
Duration: 1 week

Exam



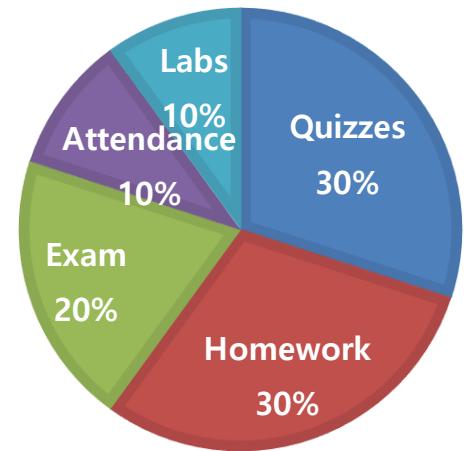
Weight: 20%

Labs

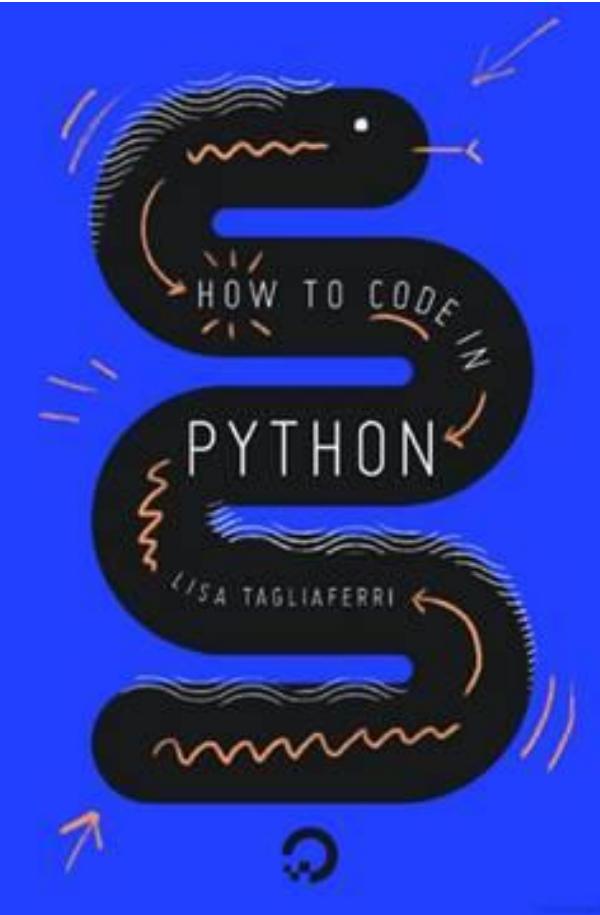
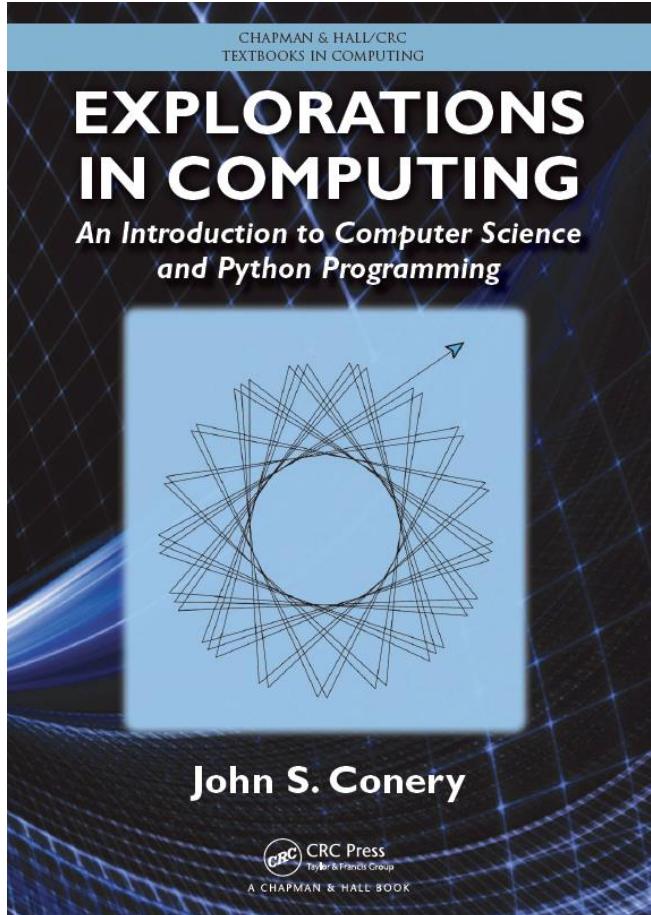


Weight: 10%
Quantity: ~ 12
Duration: Direct

COEFFICIENTS



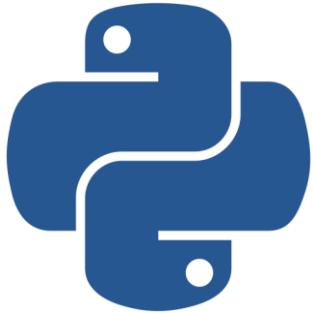
Textbooks



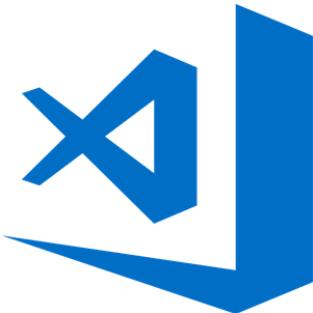
Other sources

I will likely rely on much more open source lectures, codes and data. When a slide or an exercise is inspired by some additional material it will always be specifically mentioned

Software



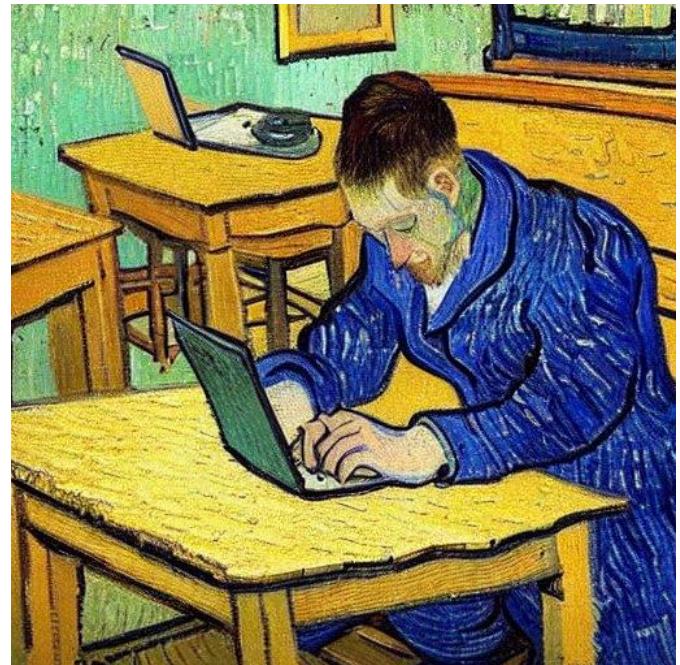
Python 3 is a high-level programming language widely used for web development, data analysis, and artificial intelligence. It offers easy-to-read syntax and powerful libraries, making it popular among beginners and experienced programmers alike.



Visual Studio Code is a free and open-source code editor developed by Microsoft for Windows, Linux, and macOS. It has a streamlined user interface and supports many programming languages, with features like debugging, syntax highlighting, and code completion.

Hardware

- Please try to bring your laptop for each class as we might have some in-class activities in addition to the lecture.
- Labs will involve coding, you might use desktop computer provided by SUNY but I recommend using your own laptop



Online classes & Recording



- I will try to record the classes and share the videos on the class website
- This might be useful when you are absent or if a new pandemic situation occurs
- I may sometimes fail due to technical difficulties so please keep taking good notes!

Note on slides



- I'll make them available by the end of the lecture **day** at the latest
 - In most cases, I'll try to deliver them **before** the lecture
 - Use your stonybrook.edu account to access them

LLM & cheating

A Large Language Model (LLM) is an AI model trained on vast amounts of text data to understand language. It uses deep learning algorithms to process and generate natural language and can perform various NLP tasks.



Typical LeetCode question

F

Given two sorted arrays nums1 and nums2 of size m and n respectively, return the median of the two sorted arrays.

The overall run time complexity should be $O(\log(m+n))$.



LLM response



To find the median of two sorted arrays nums1 and nums2 of sizes m and n respectively with overall time complexity $O(\log(m+n))$, we can use the following approach:

1. Merge the two sorted arrays into a single sorted array of size $m+n$.
2. Calculate the middle index $mid = (m+n)/2$.
3. If $(m+n)$ is even, the median is the average of the elements at indices $mid-1$ and mid in the merged array. If $(m+n)$ is odd, the median is the element at index mid in the merged array.
4. To merge the two sorted arrays, we can use the merge operation of merge sort in $O(m+n)$ time.

Here's the Python code that implements this approach:



```
def findMedianSortedArrays(nums1, nums2):
    m, n = len(nums1), len(nums2)
    if m > n:
        nums1, nums2, m, n = nums2, nums1, n, m

    start, end = 0, m
    while start <= end:
        partition1 = (start + end) // 2
        partition2 = (m + n + 1) // 2 - partition1

        maxLeft1 = float('-inf') if partition1 == 0 else nums1[partition1 - 1]
        minRight1 = float('inf') if partition1 == m else nums1[partition1]

        maxLeft2 = float('-inf') if partition2 == 0 else nums2[partition2 - 1]
        minRight2 = float('inf') if partition2 == n else nums2[partition2]

        if maxLeft1 <= minRight2 and maxLeft2 <= minRight1:
            if (m + n) % 2 == 0:
                return (max(maxLeft1, maxLeft2) + min(minRight1, minRight2)) / 2
            else:
                return max(maxLeft1, maxLeft2)
        elif maxLeft1 > minRight2:
            end = partition1 - 1
        else:
            start = partition1 + 1
```

LLM & cheating

A **Large Language Model** (LLM) is an AI model trained on vast amounts of text data to understand language. It uses deep learning algorithms to process and generate natural language and can perform various NLP tasks.



- industrial revolution marked a significant turning point in human history. The AI revolution will also profoundly change our life
- It is already impacting the programming field and you will likely use these tools in the future
- But ... refrain from using LLM to do your homework for this class! Learn and understand by yourself



GitHub
Copilot

LLM & cheating



Cheating or plagiarism will not be tolerated

If you cheat, you fail. No exceptions, no excuses

Class Schedule*

| Period | Topics |
|---------|---|
| 1 Week | Introduction to Computer Science Principles |
| 2 Week | Arithmetic operations |
| 3 Week | Variables & Functions |
| 4 Week | Functions & Conditionals |
| 5 Week | Quizz 1 - String & Error and exceptions |
| 6 Week | Loops & List |
| 7 Week | Quizz 2 - Loops & List |
| 8 Week | Loops & List |
| 9 Week | Quizz 3 - Search & Sort |
| 10 Week | Search & Sort |
| 11 Week | Quizz 4 - Recursion |
| 12 Week | Recursion |
| 13 Week | Quizz 5 - Object Oriented Programming |
| 14 Week | Object Oriented Programming |
| 15 Week | Final exam |

*Subject to changes

Introductions

A blurred background photograph showing a group of diverse individuals, including men and women of various ethnicities, smiling and shaking hands. This imagery serves as a visual metaphor for the process of introducing oneself to a new community or group.

Let's take turns introducing ourselves

Setting your environment

Forewords

Setting your **personalized** programming environment is important

Comfort



Efficiency

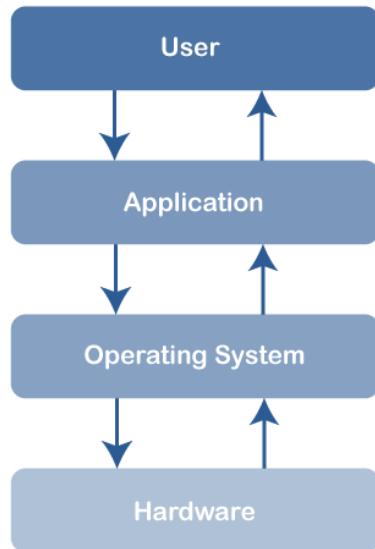


Consistency

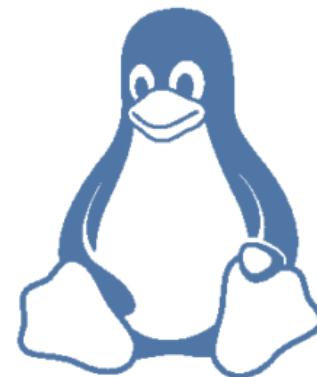
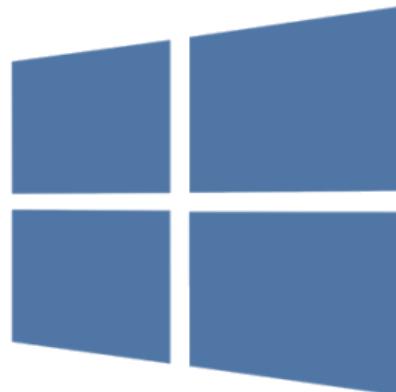


In this lecture, I give recommendations about what to set it up but
feel free to use the tools you prefer to program in Python

Operating system

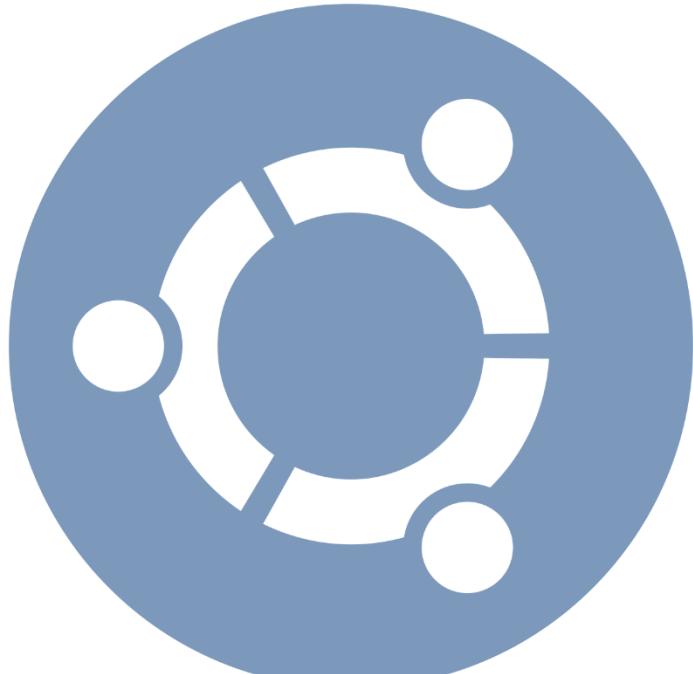


An **operating system** is a program that manages and controls the hardware and software resources of a computer. It provides a platform for running applications and allows users to interact with the computer through a graphical user interface.



Python is cross-platform; feel free to use the one you prefer

Operating system



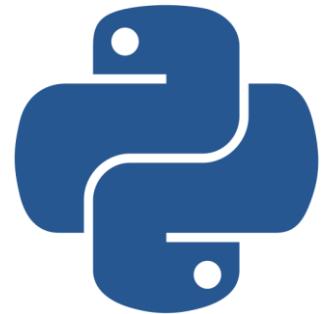
UBUNTU

- Free
- Open-source
- Good user interface
- Privacy
- Security
- Programmer oriented
- Used by all large companies

Python

Python is a **programming language**

- A programming language is a formal language designed to express instructions to a computer in a way that humans can read and write.
- So far, computers do not understand human language . . . They need a clear sequence of commands to perform
- Many languages exist, and they all have their specificity and fields of applications



Python

So what language should I learn?



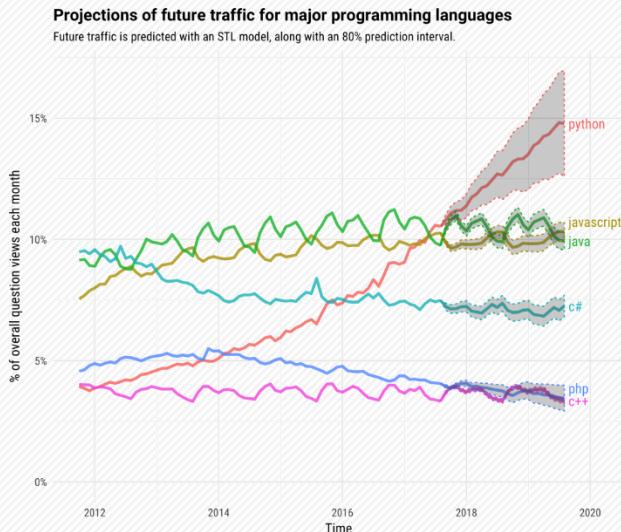
To learn to program, the language does not entirely matter.

What is essential in the first place is the general programming concept shared among all major languages.

Why start with **Python**?

Python

Most popular language



Used across many fields



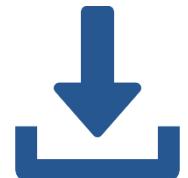
Python is a high-level language:

- User-friendly
 - Independent from hardware
- You can focus on the Principle of programming

Installing python

1. Download python

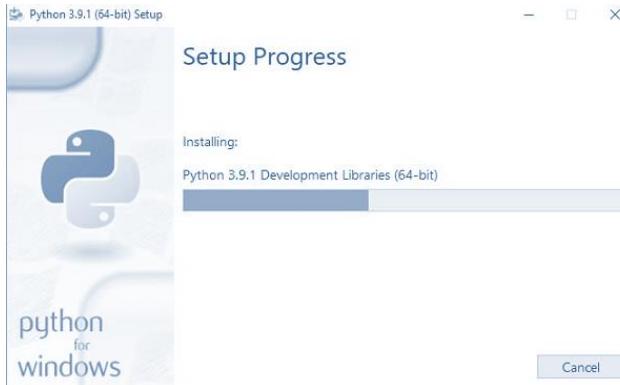
<https://www.python.org/downloads/>



(any version of Python 3 will do for this class)

2. Install python

Windows



Ubuntu

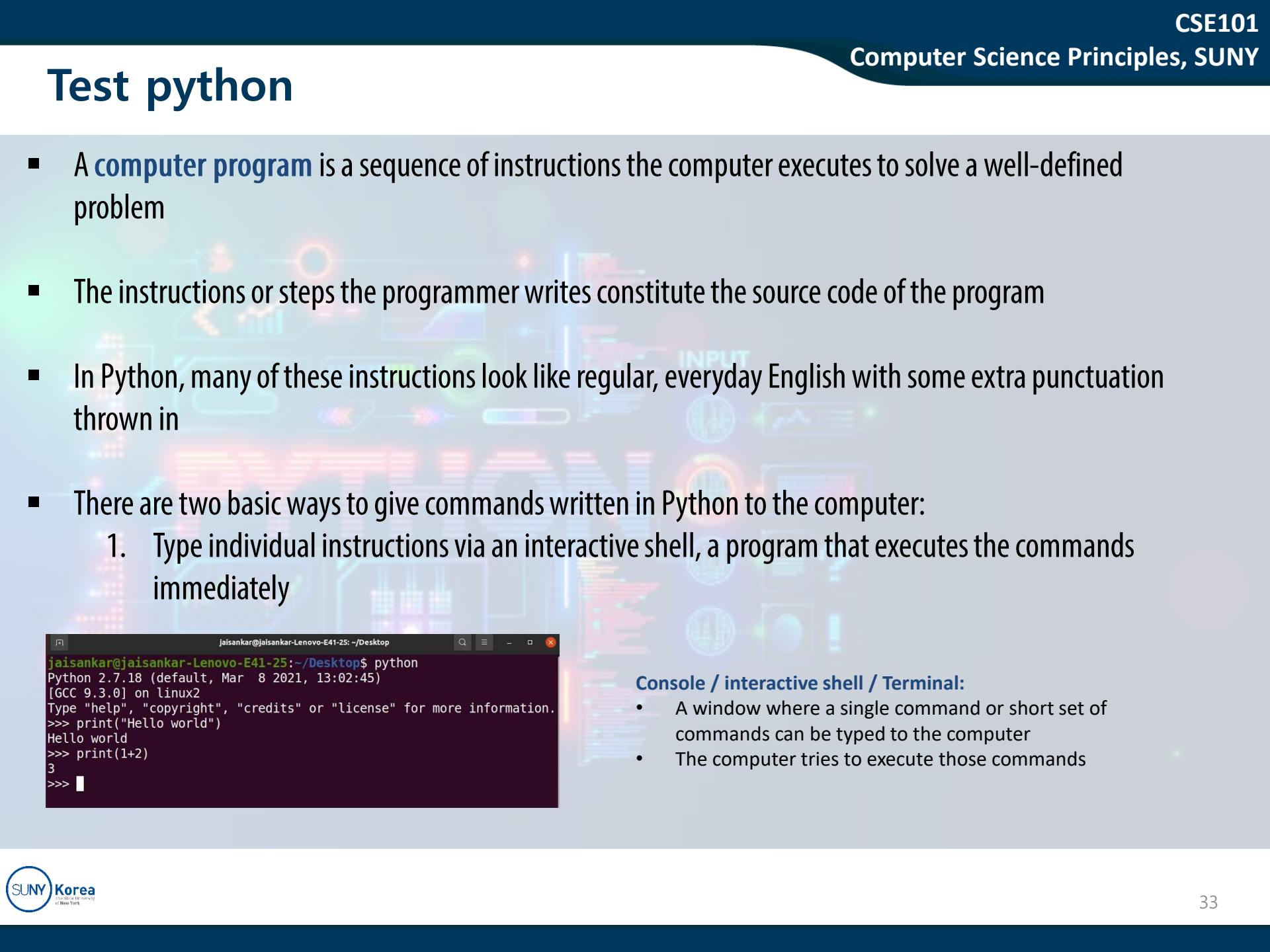


sudo apt-get install python3

(In the latest versions of Ubuntu python is pre-installed)

Test python

- A **computer program** is a sequence of instructions the computer executes to solve a well-defined problem
- The instructions or steps the programmer writes constitute the source code of the program
- In Python, many of these instructions look like regular, everyday English with some extra punctuation thrown in
- There are two basic ways to give commands written in Python to the computer:
 1. Type individual instructions via an interactive shell, a program that executes the commands immediately



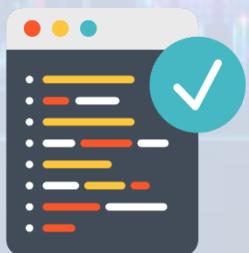
```
jaisankar@jaisankar-Lenovo-E41-25:~/Desktop$ python
Python 2.7.18 (default, Mar  8 2021, 13:02:45)
[GCC 9.3.0] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> print("Hello world")
Hello world
>>> print(1+2)
3
>>> 
```

Console / interactive shell / Terminal:

- A window where a single command or short set of commands can be typed to the computer
- The computer tries to execute those commands

Test python

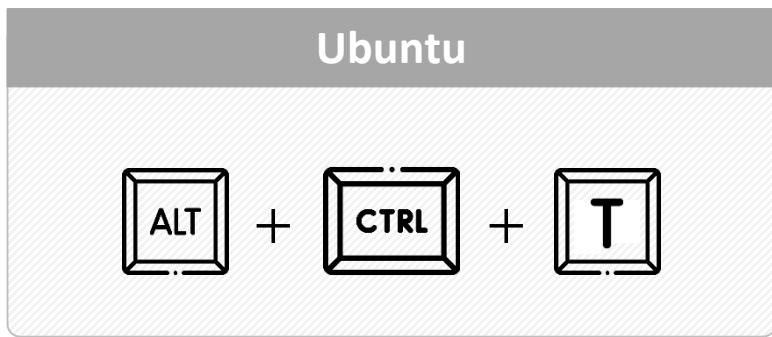
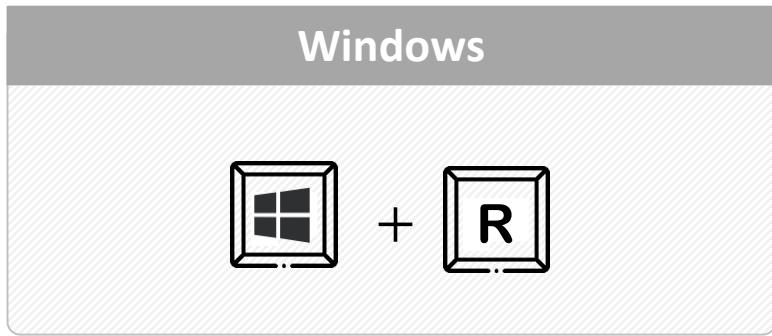
- A **computer program** is a sequence of instructions the computer executes to solve a well-defined problem
- The instructions or steps the programmer writes constitute the source code of the program
- In Python, many of these instructions look like regular, everyday English with some extra punctuation thrown in
- There are two basic ways to give commands written in Python to the computer:
 1. Type individual instructions via an interactive shell, a program that executes the commands immediately
 2. Write a complete, stand-alone application that we can run over and over



Application:

- Entire set of commands forming a code that can be saved and executed at will

Test python

A screenshot of a Linux terminal window titled "mark@linux-desktop: ~". The window includes a menu bar with "File", "Edit", "View", "Search", "Terminal", and "Help". The command line shows "mark@linux-desktop:~\$". The terminal is dark-themed with white text.A screenshot of a Windows Command Prompt window titled "Command Prompt". The window shows the title bar with "Command Prompt", the version "Microsoft Windows [Version 10.0.22621.1265]", and copyright information "(c) Microsoft Corporation. All rights reserved.". The command line shows "C:\Users\DELL>". The window has a standard Windows title bar with minimize, maximize, and close buttons.

Start the python interpreter

python

OR

python3

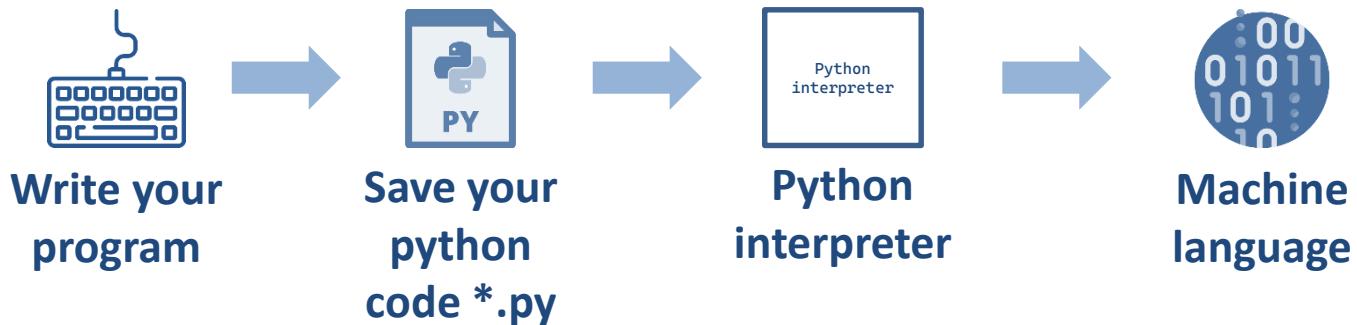
Test python

- `print("hello world")`
- `1 + 5`
- `a = 1`
- `b = 2`
- `a + b`
- `name = "SUNY"`
- `country = "Korea"`
- `print(name + country)`
- `Pi = 22/7`
- `print(type(name))`
- `print(type(Pi))`

```
>>> print("hello world")
hello world
>>> 1 + 5
6
>>> a = 1
>>> b = 2
>>> a + b
3
>>> name = "SUNY"
>>> country = "Korea"
>>> print(name + country)
SUNYKorea
>>> Pi = 22/7
>>> print(type(name))
<class 'str'>
>>> print(type(Pi))
<class 'float'>
>>>
```

Editor / IDE

In practice, you will rarely use the python command via the **terminal**



- What tools do we use to write your code, and why does it matter?
- We simply need a text editor and to save the program with a *.py extension? Right?
- Do you know a program able to do it?



Editor / IDE

IDE or programming editors provide very useful features to support your coding

Integrated Development Environment (IDE)



Editor



Compiler



Debugger

Centralized interface with all of the tools that a developer requires

Many IDEs:



Editor / IDE

Why using an IDE?
To make our life easier

- Syntax Highlighting

Def:

highlighting or colouring specific syntax

This enables you to simply see, easily and quickly determine, whether you are creating any syntax issues, so that when you do, you can immediately fix them.



```
247  /**
248  * @brief Compute the reprojection error for this object observation
249  *
250  * @return mean reprojection error for this observation
251  */
252 float Object3DObs::computeReprojectionError() const {
253     float sum_err_object = 0.0;
254     std::vector<cv::Point3f> object_pts_temp;
255     object_pts_temp.reserve(pts_id_.size());
256     for (const auto &pt_id : pts_id_) {
257         auto object_3d_ptr = object_3d_.lock();
258         if (object_3d_ptr)
259             object_pts_temp.emplace_back(object_3d_ptr->pts_3d_[pt_id]);
260     }
261
262     // Project the 3D pts on the image
263     std::vector<cv::Point2f> repro_pts;
264     std::vector<float> error_object_vec;
265     std::shared_ptr<Camera> cam_ptr = cam_.lock();
266     if (cam_ptr) {
267         projectPointsWithDistortion(object_pts_temp, getRotVec(), getTransVec(),
268                                     cam_ptr->getCameraMat(),
269                                     cam_ptr->getDistortionVectorVector(), repro_pts,
270                                     cam_ptr->distortion_model_);
271         for (std::size_t j = 0; j < repro_pts.size(); j++) {
272             float rep_err = std::sqrt(std::pow((pts_2d_[j].x - repro_pts[j].x), 2) +
273                                     std::pow((pts_2d_[j].y - repro_pts[j].y), 2));
274             error_object_vec.push_back(rep_err);
275             sum_err_object += rep_err;
276             // if (rep_err > 6.0)
277             // LOG_WARNING << "LARGE REPROJECTION ERROR ::: " << rep_err ;
278         }
279         LOG_DEBUG << "Frame :: " << this->frame_id_
280             << " object :: " << this->object_3d_id_ << " --- Mean Error ::"
281             << sum_err_object / error_object_vec.size()
282             << " Nb pts :: " << error_object_vec.size();
283     }
284     // return mean error for the board
285     return error_object_vec.size() > 0 ? sum_err_object / error_object_vec.size()
286                                         : sum_err_object;
287 }
288 }
```

Editor / IDE

Why using an IDE?
To make our life easier

- Syntax Highlighting
- Auto-indentation



Def:

Auto-indentation is a feature in text editors that automatically adjusts the indentation of lines of code or text based on the structure of the document. It helps to maintain consistent and readable formatting, making it easier to navigate and understand the content.



```
import time      # statement 1

def count(limit):    # statement 2
    result = 0      # statement 3
    for a in range(1, limit + 1):      # statement 4
        for b in range(a + 1, limit + 1):
            for c in range(b + 1, limit + 1):
                if c * c > a * a + b * b:
                    break
                if c * c == (a * a + b * b):
                    result += 1
    return result
```

Editor / IDE

Why using an IDE?
To make our life easier

- Syntax Highlighting
- Auto-indentation
- Auto-completion and brace matching



Def:

Auto-completion is a feature in code editors that suggests possible code completions as the user types, making programming more efficient and reducing errors. Brace matching is a feature that highlights the corresponding opening or closing bracket of a code block, helping the user identify errors and navigate code more easily.



```
E:\My_Projects>Test>JS patterns.js > [o] myModule > ↵ <function>
1 const myModule = (function()
2 {
3     const testModule = {}
4     testModule.hello = function()
5     {
6         console.log( "Hello World!" );
7     }
8     return test;
9 })();
10 myModule.hel
11 myModule.hel
[...] testModule
[...] TextDecoderStream
[...] TextEncoderStream
[...] TextTrackList
[...] TextTrackCueList
[...] TextMetrics
[...] TransitionEvent
[...] DOMSettableTokenList
[...] DataTransferItemList
[...] DataTransferItem
[...] PageTransitionEvent
[...] SVGFEComponentTransferElement
const testModule: {
    hello(): void;
}
```

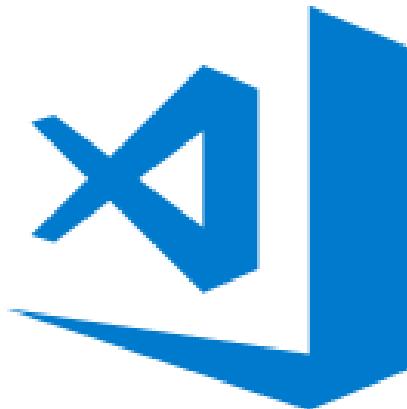
Editor / IDE

These are great tools, but deep down, we all know that word is the best IDE! (sarcasm) source: Joma Tech



Editor / IDE

For this lecture, I invite you to use:



Visual Studio Code

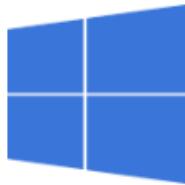
Let's install it!!

<https://code.visualstudio.com/download>

Install VSCode

Download Visual Studio Code

Free and built on open source. Integrated Git, debugging and extensions.



↓ Windows

Windows 8, 10, 11

| | | | |
|------------------|-----|-----|-------|
| User Installer | x64 | x86 | Arm64 |
| System Installer | x64 | x86 | Arm64 |
| .zip | x64 | x86 | Arm64 |
| CLI | x64 | x86 | Arm64 |

↓ .deb

Debian, Ubuntu

↓ .rpm

Red Hat, Fedora, SUSE

↓ Mac

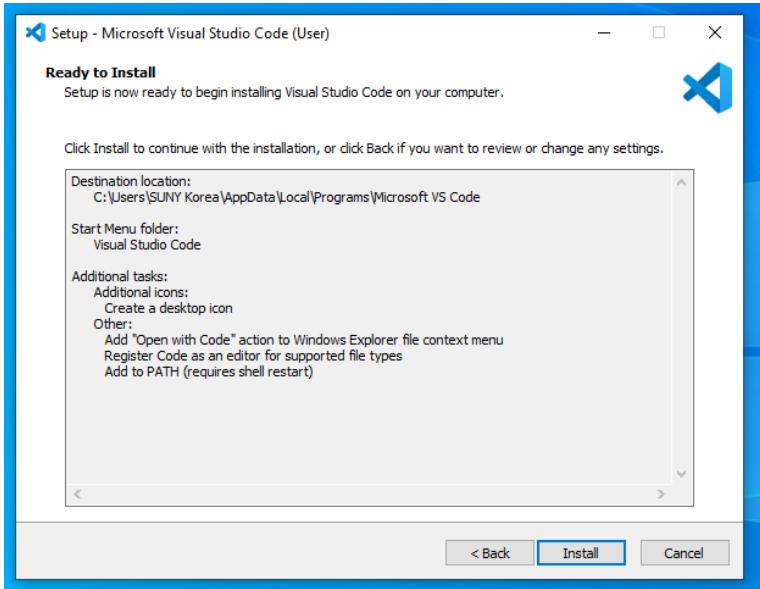
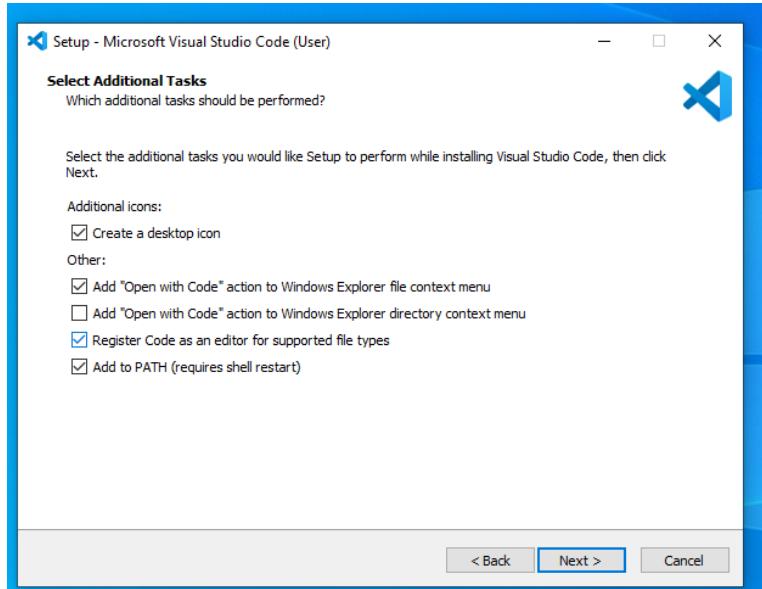
macOS 10.11+

| | | | |
|---------|-----------------|-------|-------|
| .deb | x64 | Arm32 | Arm64 |
| .rpm | x64 | Arm32 | Arm64 |
| .tar.gz | x64 | Arm32 | Arm64 |
| Snap | Snap Store | | |
| CLI | x64 Arm32 Arm64 | | |

.zip Intel chip Apple silicon Universal

CLI Intel chip Apple silicon

Install VSCode



```
sudo apt install ./<file>.deb  
sudo apt install apt-transport-https  
sudo apt update  
sudo apt install code
```

Setup VSCode for Python

Feel comfortable, set up your color theme!

File > Preferences > Theme > Color Theme

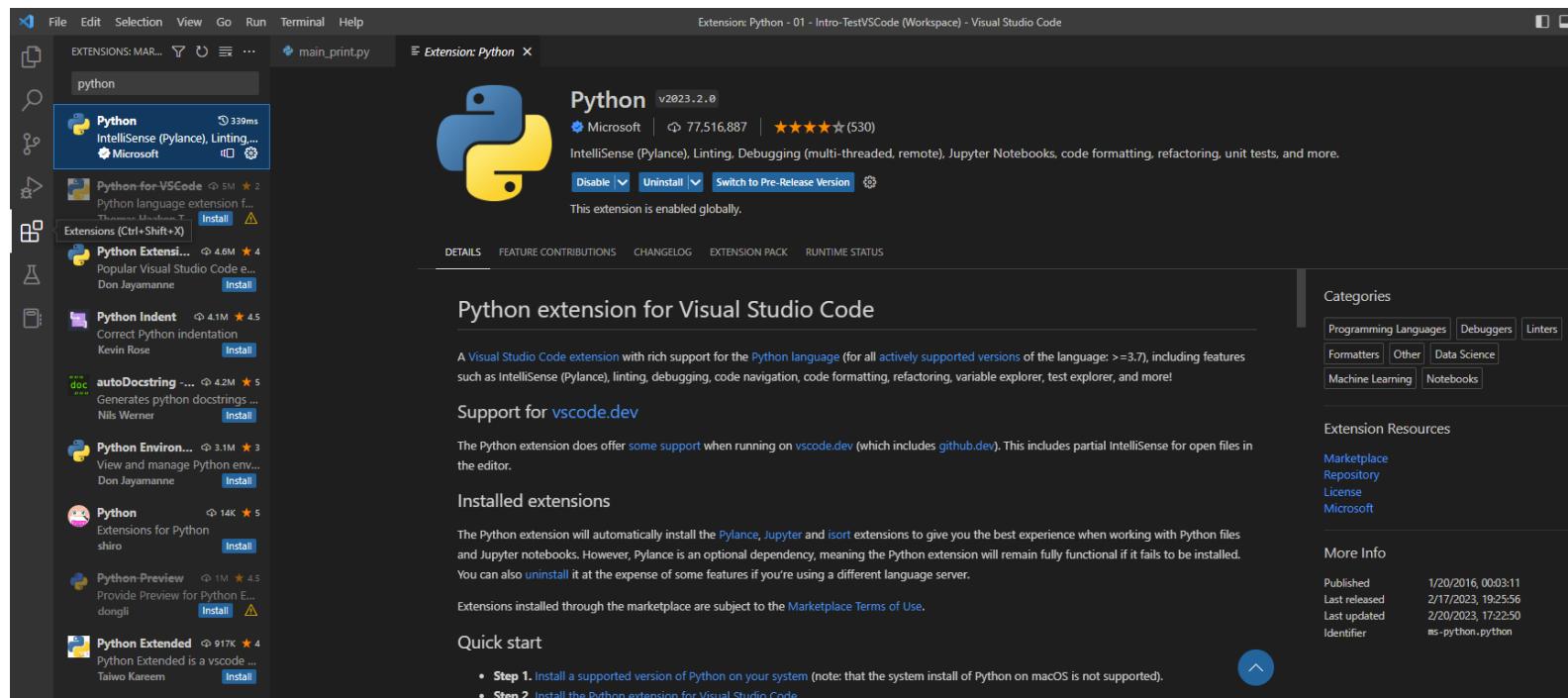


The screenshot shows the Visual Studio Code interface. On the left, a code editor displays a file named 'configuration.ts' with TypeScript code. On the right, a dropdown menu titled 'Select Color Theme' is open, listing various themes: Abyss, Dark (Visual Studio), Dark+ (default dark), High Contrast, Kimbie Dark (which is currently selected), Light (Visual Studio), Light+ (default light), Monokai, Monokai Dimmed, Quiet Light, and Red.

```
configuration.ts src\vs\platform\configuration
1  /*
2   * Copyright (c) Microsoft Corporation. All rights reserved.
3   * Licensed under the MIT License.
4   */
5
6  import {createDecorator}
7  import {IEventEmitter}
8  import Event from 'vs/base/common/event'
9  import winjs = require('winjs')
10
11 export const IConfigurationService = require('./configuration').IConfigurationService;
12
13 export interface IConfigurationService {
14     serviceId: ServiceIdentifier<any>;
15
16     /**
17      * Fetches the appropriate section of the configuration JSON file.
18      * This will be an object keyed off the section name.
19      */
20     loadConfiguration(section?: string): winjs.TPromise<any>;
21 }
```

Setup VSCode for Python

Install the **Python IDE extension** on VS Code

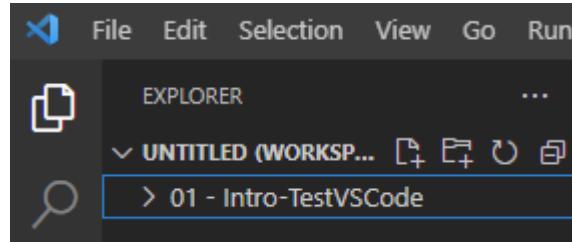


It will unlock many functionalities like auto-completion, choosing its interpreter, linter, etc...

Setup your workspace

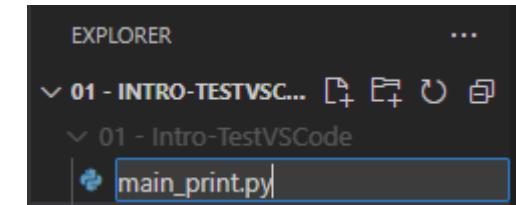
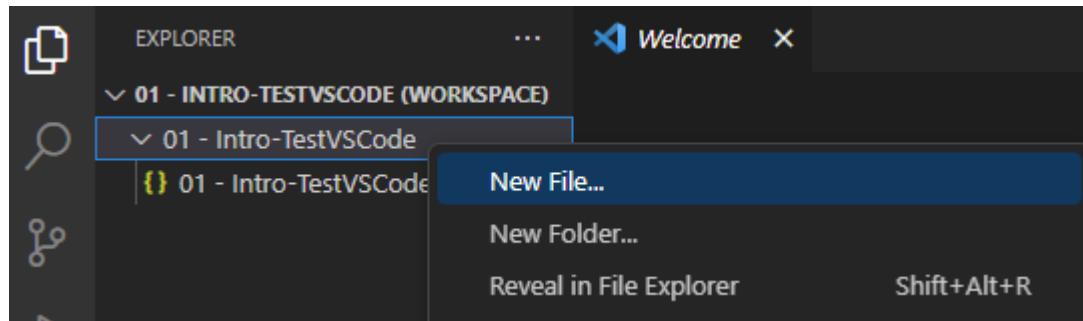
1. Create a **workspace** for your project

File > Add Folder to Workspace



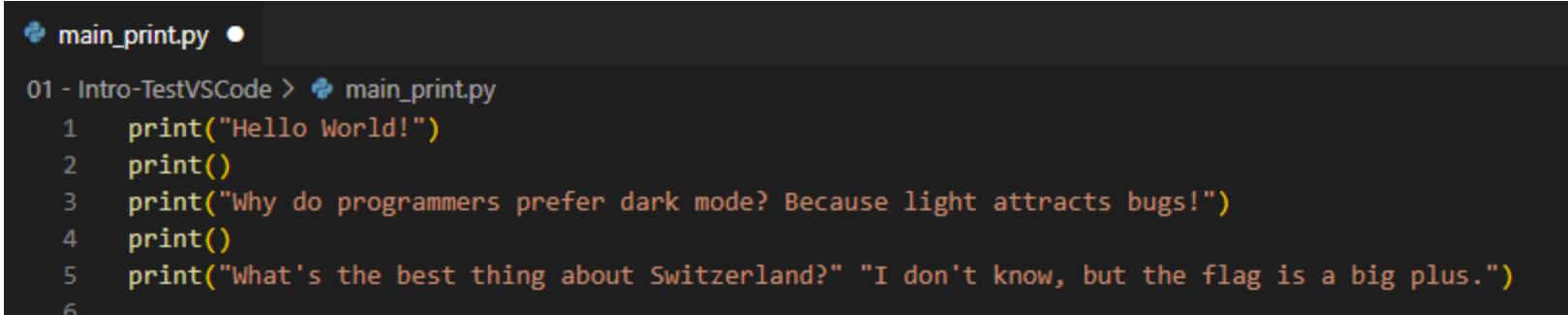
File > Save Workspace as ...

2. Create a python file (*.py) where you will **code**



Setup your workspace

3. Now you can **start** coding!!



```
main_print.py
01 - Intro-TestVSCode > main_print.py
1     print("Hello World!")
2     print()
3     print("Why do programmers prefer dark mode? Because light attracts bugs!")
4     print()
5     print("What's the best thing about Switzerland? " "I don't know, but the flag is a big plus.")
6
```

Great code !!! But how do we run it now?

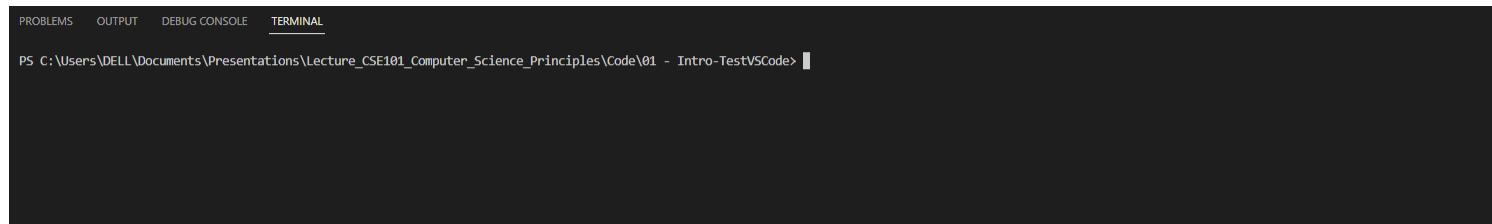
Setup your workspace

Note:

You can also do Run > Run without Debugging

4. Run a terminal

Terminal > New Terminal

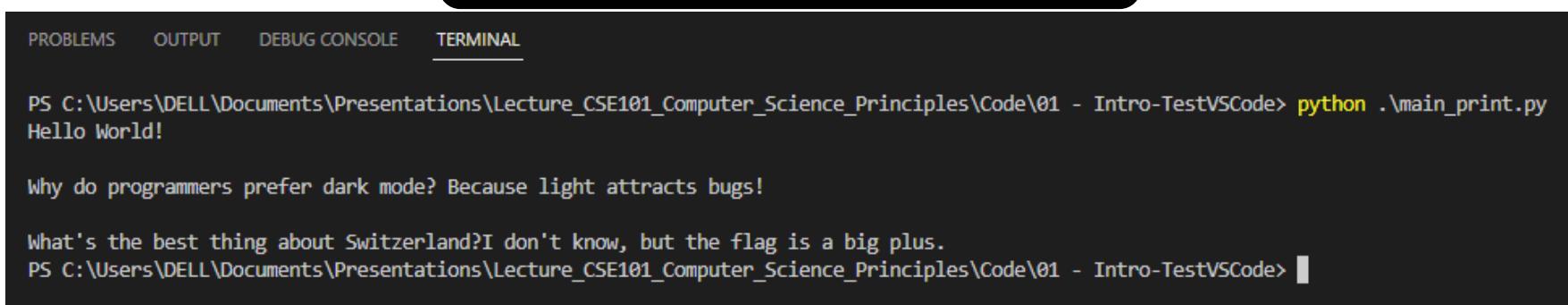


A screenshot of the Visual Studio Code interface. The top navigation bar has tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, and TERMINAL, with TERMINAL being the active tab. The main area shows a terminal window with the following text:
PS C:\Users\DELL\Documents\Presentations\Lecture_CSE101_Computer_Science_Principles\Code\01 - Intro-TestVSCode> |

5. Run your code

In the terminal write

python my_code.py



A screenshot of the Visual Studio Code interface showing the terminal output. The top navigation bar has tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, and TERMINAL, with TERMINAL being the active tab. The main area shows a terminal window with the following text:
PS C:\Users\DELL\Documents\Presentations\Lecture_CSE101_Computer_Science_Principles\Code\01 - Intro-TestVSCode> python .\main_print.py
Hello World!

Why do programmers prefer dark mode? Because light attracts bugs!

What's the best thing about Switzerland? I don't know, but the flag is a big plus.
PS C:\Users\DELL\Documents\Presentations\Lecture_CSE101_Computer_Science_Principles\Code\01 - Intro-TestVSCode> |