Module 1

Introduction

Outline

- Overview
- Python
- Basics
- String
- Built-in Functions

Outline

- Overview
- Python
- Basics
- String
- Built-in Functions

Outline

- Overview
- Python
- Basics
- String
- Built-in Functions

This part is important! It covers the justification of

- Why we cover the topics
- Why we mark your Lab/Exam in certain way
- Pay special attention to texts in a different color (e.g., red, blue)!!

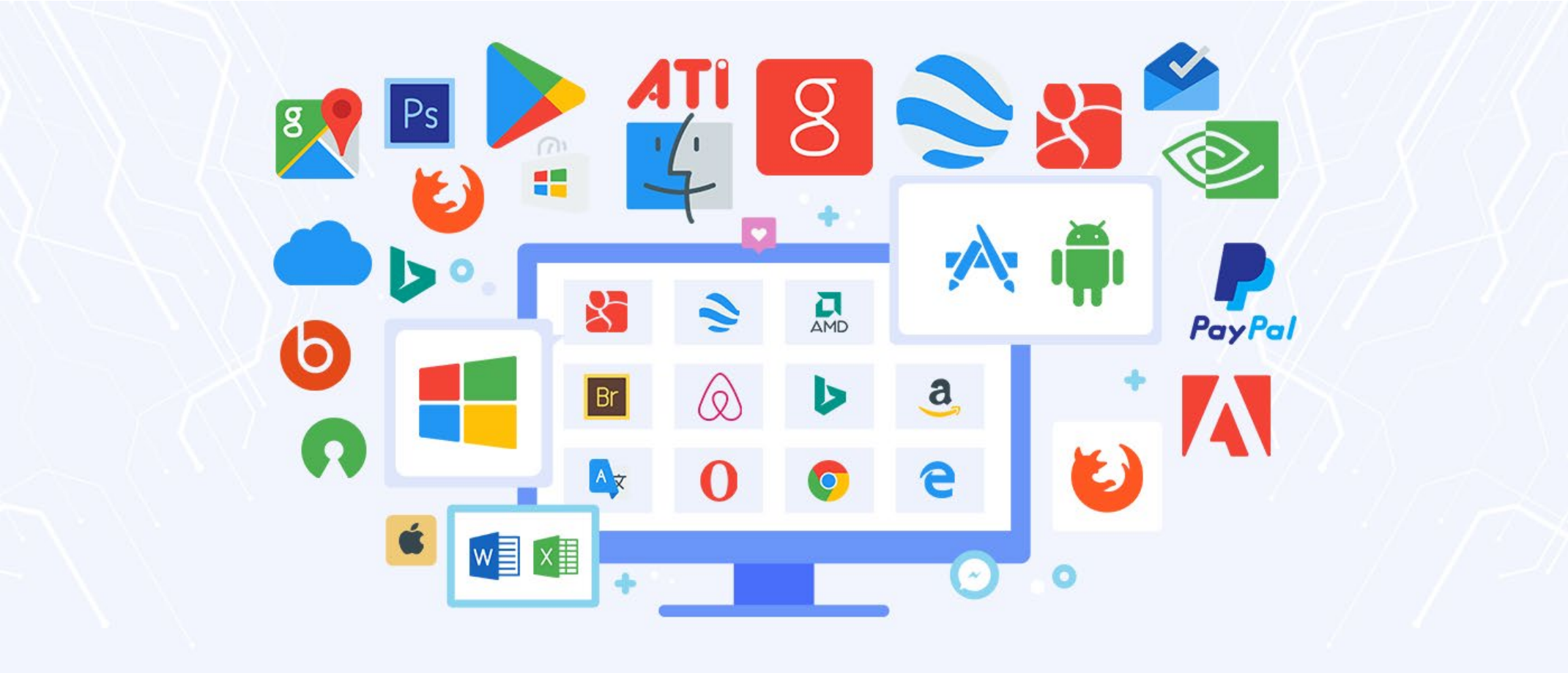
Outline

- Overview
  - What is Computer Programming?
  - How Does a Computer Take Instructions?
  - What are the Necessary Components of a High-level Programming Language?
  - Guidelines of Computer Science
- Python
- Basics
- String
- Built-in Functions

Introduction

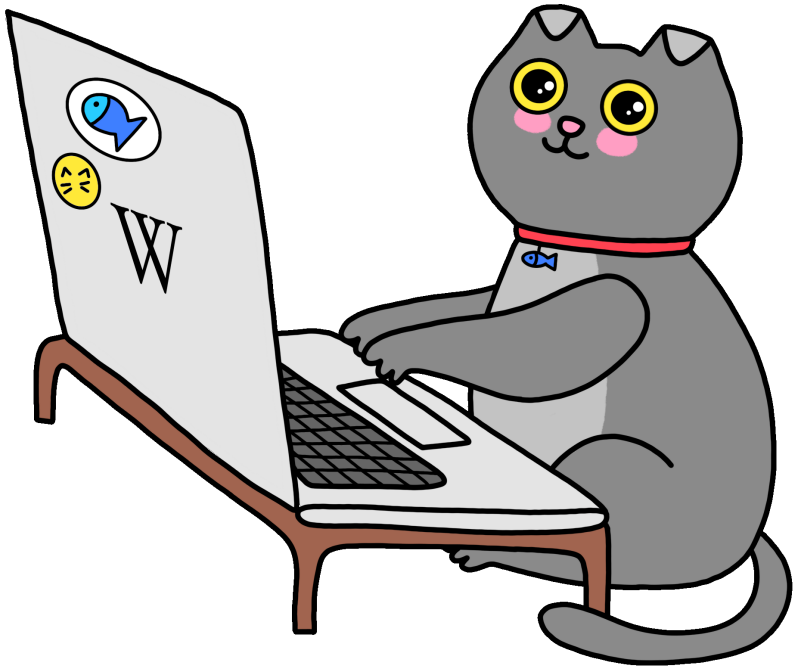
- What is computer programming?
- How does a computer take instructions?
- What are the necessary components of a high-level programming language?
- Guidelines for computer science
  - Evaluating a program (Law of computing)
  - Principles and practices

Why do we need to learn programming?



Computer Programming

- Give instructions to the computer via programming
  - Writing code



Introduction

- What is computer programming?
- How does a computer take instructions?
- What are the necessary components of a high-level programming language?
- Guidelines for computer science
  - Evaluating a program (Law of computing)
  - Principles and practices

How?

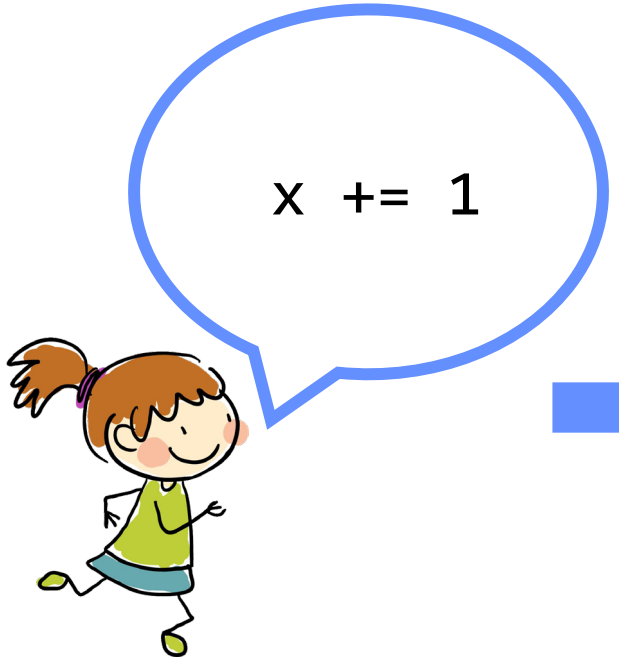
- E.g., Making a Sandwich

- **Semantics**
  - Spread: open out (something) so as to extend its surface area, width, or length.
  - The peanut butter: a food paste or spread made from ground, dry-roasted peanuts
  - On: physically in contact with and supported by (a surface)
  - The face of a bread: The side of the bread with larger surface area
- **Syntax**
  - Verb (spread) + Noun (the peanut butter) + Preposition (on) + Noun phrase (the side of the bread)

- **Contact** knife **with** bread
- **Move** knife left **to** right
- **Move** knife right **to** left
- **Move** knife left **to** right
- **Move** knife right **to** left
- **Move** knife left **to** right
- **Move** knife right **to** left
- **Lift** knife

How?

- E.g., Programming



- **Semantics**
  - x: variable
  - +: addition
  - =: assignment
  - 1: interger
- **Syntax**
  - +=

```
00 LDA 07, R1
01 LDA 08, R1
02 ADD R1, R2
03 STO 08
04 END
05
06
07 DAT 1
08 DAT 2
```

How?

- E.g., Programming

High-level Programming Language:

- Python
- Java
- C/C++
- JavaScript
- …

Interpreter

Compiler

- **Semantics**
  - x: variable
  - +: addition
  - =: assignment
  - 1: integer
- **Syntax**
  - +=

Low-level

Machine Code

```
00 LDA 07, R1
01 LDA 08, R1
02 ADD R1, R2
03
04 END
05
06
07 DAT 1
08 DAT 2
```

How?

- E.g., Programming

More info (optional):

- https://www.teach.cs.toronto.edu/~csc110y/fall/notes/01-working-with-data/01-python-language.html

High-level Programming Language:

- Python
- Java
- C/C++
- JavaScript
- …

Interpreter

Compiler

- Semantics
  - x: variable
  - +: addition
  - =: assignment
  - 1: integer
- Syntax
  - +=

Low-level

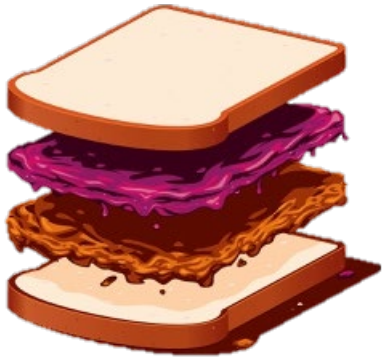
Machine Code

```
00 LDA 07, R1
01 LDA 08, R1
02 ADD R1, R2
03
04 END
05
06
07 DAT 1
08 DAT 2
```

Introduction

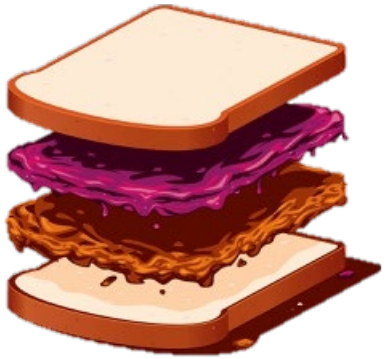
- What is computer programming?
- How does a computer take instructions?
- What are the necessary components of a high-level programming language?
- Guidelines for computer science
  - Evaluating a program (Law of computing)
  - Principles and practices

What? (1 of 9) – Literals



- E.g., Peanut butter and jelly sandwich instructions
  - Take 2 pieces of white bread
  - Pick the lid up from the peanut butter jar
  - Take a butter knife and stick it inside the peanut butter jar
  - With the knife, scoop some peanut butter out of the inside of the jar
  - Spread your scoop of peanut butter onto the face of one of your pieces of bread with a knife
  - Squeeze some jelly onto the other piece of bread
  - Spread the jelly on the bread with the butter knife
  - Put your pieces of bread peanut butter and jelly sides together

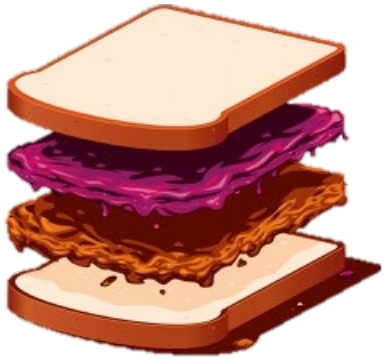
What? (1 of 9) – Literals



- E.g., Peanut butter and jelly sandwich instructions
  - Take 2 pieces of white bread
  - Pick the lid up from the peanut butter jar
  - Take a butter knife inside the peanut butter jar
  - With the knife, sco anut butter out of the inside of the jar
  - Spread your scoop of peanut butter onto the face of one of your pieces of bread with a knife
  - Squeeze some jelly onto the other piece of bread
  - Spread the jelly on the bread with the butter knife
  - Put your pieces of bread peanut butter and jelly sides together

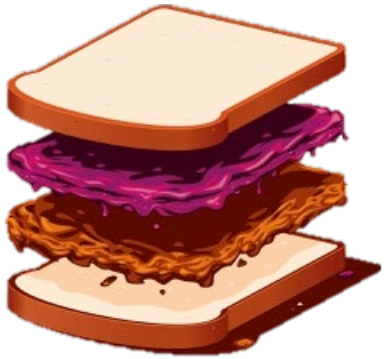
Integer

What? (1 of 9) – Literals



- E.g., Peanut butter and jelly sandwich instructions
  - Take 2 pieces of white bread
  - Pick the lid up from the peanut butter jar
  - Take a butter knife and stick it inside the peanut butter jar
  - With the knife, scoop some peanut butter out of the inside of the jar
  - Spread your scoop of peanut butter onto the face of one of your pieces of bread with a knife
  - Squeeze some jelly onto the other piece of bread
  - Spread the jelly on the bread with the butter knife
  - Put your pieces of bread peanut butter and jelly sides together

What? (1 of 9) – Literals



- E.g., Peanut butter and jelly sandwich instructions
  - Take 2 pieces of white bread
  - Pick the lid up from the peanut butter jar
  - Take a butter knife and stick it inside the peanut butter jar
  - With the knife, scoop 5.1 g a peanut butter out of the inside of the jar
  - Spread your scoop of peanut butter onto the face of one of your pieces of bread with a knife
  - Squeeze some jelly onto the other piece of bread
  - Spread the jelly on the bread with the butter knife
  - Put your pieces of bread peanut butter and jelly sides together

Float

What? (1 of 9) – Literals

- Numerical Literals
  - Integer: e.g., 0, 2, -50, 100
  - Float: e.g., 3.5, -0.112, 10023.512

What? (1 of 9) – Literals

- Numerical Literals
  - Integer: e.g., 0, 2, -50, 100
  - Float: e.g., 3.5, -0.112, 10023.512
- String literals
  - A sequence of characters
    - One character: e.g., ‘a’, ‘5’, ‘)’, ‘ ‘
    - String: e.g., “Hello World!”

What? (1 of 9) – Literals

- Numerical Literals
  - Integer: e.g., 0, 2, -50, 100
  - Float: e.g., 3.5, -0.112, 10023.512
- String literals
  - A sequence of characters
    - One character: e.g., ‘a’, ‘5’, ‘)’, ‘ ‘
    - String: e.g., “Hello World!”
- The Boolean literal
  - `True` and `False`

What? (1 of 9) – Literals

- Numerical Literals
  - Integer: e.g., 0, 2, -50, 100
  - Float: e.g., 3.5, -0.112, 10023.512
- String literals
  - A sequence of characters
    - One character: e.g., ‘a’, ‘5’, ‘)’, ‘ ‘
    - String: e.g., “Hello World!”
- The Boolean literal
  - `True` and `False`
- The None literal[1]

1: Python specific. It means the data has no type. It may be null or something else in other languages

What? (1 of 9) – Literals

Literals are raw values or data

- Numerical Literals
  - Integer: e.g., 0, 2, -50, 100
  - Float: e.g., 3.5, -0.112, 10023.512
- String literals
  - A sequence of characters
    - One character: e.g., ‘a’, ‘5’, ‘)’, ‘ ‘
    - String: e.g., “Hello World!”
- The Boolean literal
  - `True` and `False`
- The None literal[1]

1: Python specific. It means the data has no type. It may be null or something else in other languages

What? (1 of 9) – Literals

- Numerical Literals
  - Integer: e.g., 0, 2, -50, 100
  - Float: e.g., 3.5, -0.112, 10023.512
- String literals
  - A sequence of characters
    - One character: e.g., ‘a’, ‘5’, ‘)’, ‘ ‘
    - String: e.g., “Hello World!”
- The Boolean literal
  - `True` and `False`
- The None literal[1]

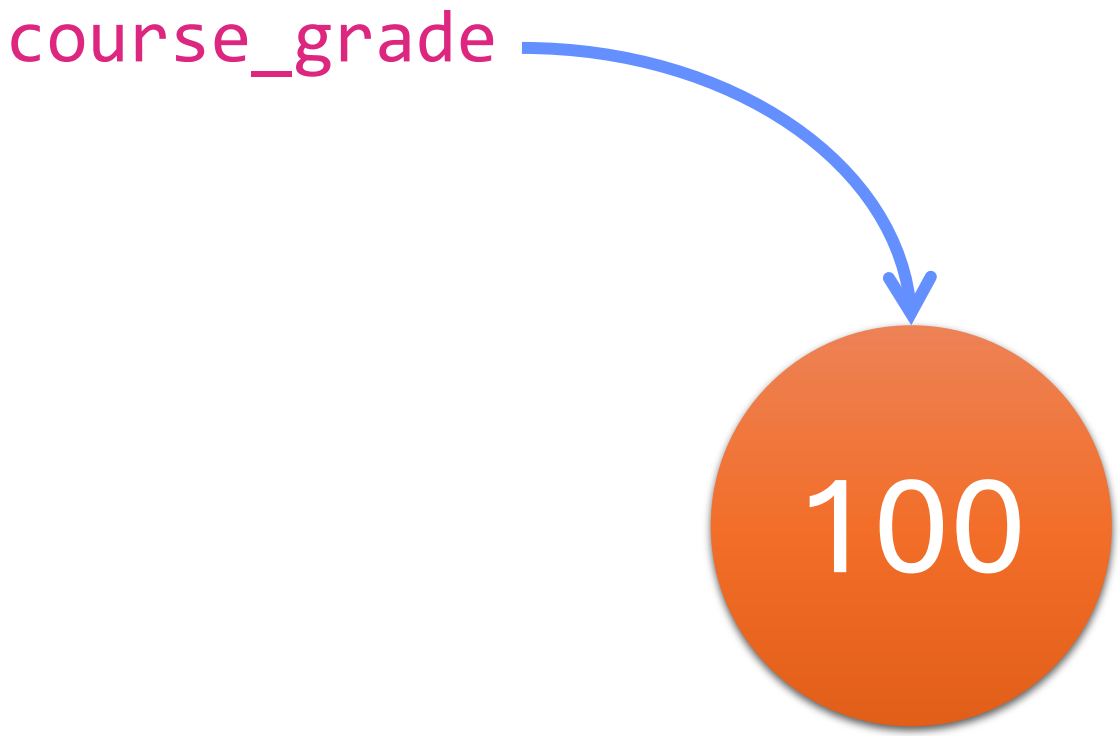
Literals are raw values or data

There are many different types of data.

1: Python specific. It means the data has no type. It may be null or something else in other languages

What? (2 of 9) – Variables

- Give a name to a literal



course_grade = 100

What? (3 of 9) – Operations

- Manipulating the literals/Variables
  - Arithmetic Operators
    - E.g., Addition, Subtraction, Division, Multiplication
  - Assignment Operators
    - E.g., Assigning
  - Comparison Operators
    - E.g., Equal, Greater than
  - Logical Operators
    - E.g., and, or, not
  - Bitwise Operators
  - …

You do not need to know what they are for now.
We will go over them later.

What? (4 of 9) – Collection of literals/Variables

- Collection of literals
  - List
    - E.g., [1, 3, 2, 5, 6, 7]
    - E.g., [“Meiying”, “Sam”, “Monty”]
  - Tuple
    - E.g., (1, 3, 2, 5, 6, 7)
    - E.g., (“Meiying”, “Sam”, “Monty”)
  - Set
    - E.g., {1, 3, 2, 5, 6, 7}
    - E.g., {“Meiying”, “Sam”, “Monty”}
  - Dictionary
    - E.g., {“Meiying”: 97, “Sam”: 50, “Monty”: 78}

Again, you do not need to know what they are for now. We will go over them one by one

What? (5 of 9) Expressions/Statements

- Expressions
  - A combination of literals, variables, and operators
  - E.g., `course_grade + 30.1`

What? (5 of 9) Expressions/Statements

- Expressions
  - A combination of literals, variables, and operators
  - E.g., `course_grade + 30.1`
- Statements
  - A unit of code that creates a variable and/or manipulating it
  - E.g., `course_grade = 100`

What? (5 of 9) Expressions/Statements

- Expressions
  - A combination of literals, variables, and operators
  - E.g., `course_grade + 30.1`
- Statements
  - A unit of code that creates a variable and/or manipulating it
  - E.g., `course_grade = 100`

You do not have to distinguish the definition of expressions and statements if you do not get it;

The point here is that it combines literals/variables (or even collections of those), and operators

What? (6 of 9) If-statement

- E.g., Peanut butter and jelly sandwich instructions
  - Take 2 pieces of white bread
  - Pick the lid up from the peanut butter jar
  - Take a butter knife and stick it inside the peanut butter jar
  - With the knife, scoop some peanut butter out of the inside of the jar
  - If you like peanut butter, scoop more peanut butter
  - Spread your scoop of peanut butter onto the face of one of your pieces of bread with a knife
  - Squeeze some jelly onto the other piece of bread
  - Spread the jelly on the bread with the butter knife
  - Put your pieces of bread peanut butter and jelly sides together

What? (7 of 9) Loops

- E.g., Making 3 peanut butter and jelly sandwich
    - Take 2 pieces of white bread
    - Pick the lid up from the peanut butter jar
    - Take a butter knife and stick it inside the peanut butter jar
    - With the knife, scoop some peanut butter out of the inside of the jar
    - Spread your scoop of peanut butter onto the face of one of your pieces of bread with a knife
    - Squeeze some jelly onto the other piece of bread
    - Spread the jelly on the bread with the butter knife
    - Put your pieces of bread peanut butter and jelly sides together

    - Take 2 pieces of white bread
    - Pick the lid up from the peanut butter jar
    - Take a butter knife and stick it inside the peanut butter jar
    - With the knife, scoop some peanut butter out of the inside of the jar
    - Spread your scoop of peanut butter onto the face of one of your pieces of bread with a knife
    - Squeeze some jelly onto the other piece of bread
    - Spread the jelly on the bread with the butter knife
    - Put your pieces of bread peanut butter and jelly sides together

    - Take 2 pieces of white bread
    - Pick the lid up from the peanut butter jar
    - Take a butter knife and stick it inside the peanut butter jar
    - With the knife, scoop some peanut butter out of the inside of the jar
    - Spread your scoop of peanut butter onto the face of one of your pieces of bread with a knife
    - Squeeze some jelly onto the other piece of bread
    - Spread the jelly on the bread with the butter knife
    - Put your pieces of bread peanut butter and jelly sides together

Tedious!!!!

What? (7 of 9) Loops

- E.g., Making 3 peanut butter and jelly sandwich
  - Repeat the following 3 times:
    - Take 2 pieces of white bread
    - Pick the lid up from the peanut butter jar
    - Take a butter knife and stick it inside the peanut butter jar
    - With the knife, scoop some peanut butter out of the inside of the jar
    - Spread your scoop of peanut butter onto the face of one of your pieces of bread with a knife
    - Squeeze some jelly onto the other piece of bread
    - Spread the jelly on the bread with the butter knife
    - Put your pieces of bread peanut butter and jelly sides together

The advantage of loops is more than just repeating
We will go over them later

What? (8 of 9) Functions

- E.g., Making peanut butter and jelly sandwich Instruction
  - Take 2 pieces of white bread
  - Pick the lid up from the peanut butter jar
  - Take a butter knife and stick it inside the peanut butter jar
  - With the knife, scoop some peanut butter out of the inside of the jar
  - Spread your scoop of peanut butter onto the face of one of your pieces of bread with a knife
  - Squeeze some jelly onto the other piece of bread
  - Spread the jelly on the bread with the butter knife
  - Put your pieces of bread peanut butter and jelly sides together

What? (8 of 9) Functions

- E.g., Making A and B sandwich Instruction
  - Take 2 pieces of white bread
  - Pick the lid up from the A jar
  - Take a butter knife and stick it inside the A jar
  - With the knife, scoop some A out of the inside of the jar
  - Spread your scoop of A onto the face of one of your pieces of bread with a knife
  - Squeeze some B onto the other piece of bread
  - Spread the B on the bread with the butter knife
  - Put your pieces of bread A and B sides together

What? (8 of 9) Functions

- E.g., Making sandwich(A, B)
  - Take 2 pieces of white bread
  - Pick the lid up from the A jar
  - Take a butter knife and stick it inside the A jar
  - With the knife, scoop some A out of the inside of the jar
  - Spread your scoop of A onto the face of one of your pieces of bread with a knife
  - Squeeze some B onto the other piece of bread
  - Spread the B on the bread with the butter knife
  - Put your pieces of bread A and B sides together

What? (8 of 9) Functions

- E.g., Making sandwich(A, B)

Making sandwich
(Peanut butter , jelly)

Peanut butter
and jelly
sandwich

Making sandwich
(Nutella, Strawberry jelly)

Nutella
and Strawberry jelly
sandwich

Making sandwich
(Humus, Mayo)

Humus
and Mayo
sandwich

What? (8 of 9) Functions

- Advantages
  - You can reuse your instructions/code for slightly different things

What? (8 of 9) Functions

- Advantages
  - You can reuse your instructions/code for slightly different things
  - Your instruction/code can be more concise and more readable

- Take 2 pieces of white bread
- Pick the lid up from the peanut butter jar
- Take a butter knife and stick it inside the peanut butter jar
- With the knife, scoop some peanut butter out of the inside of the jar
- Spread your scoop of peanut butter onto the face of one of your pieces of bread with a knife
- Squeeze some jelly onto the other piece of bread
- Spread the jelly on the bread with the butter knife
- Put your pieces of bread peanut butter and jelly sides together

- Take 2 pieces of white bread
- Pick the lid up from the Nutella jar
- Take a butter knife and stick it inside the Nutella jar
- With the knife, scoop some Nutella out of the inside of the jar
- Spread your scoop of Nutella onto the face of one of your pieces of bread with a knife
- Squeeze some strawberry jelly onto the other piece of bread
- Spread the strawberry jelly on the bread with the butter knife
- Put your pieces of bread Nutella and strawberry jelly sides together

Make sandwich(Peanut butter, jelly)

Make sandwich(Nutella, strawberry jelly)

What? (8 of 9) Functions

- Advantages
  - You can reuse your instructions/code for slightly different things
  - Your instruction/code can be more concise and more readable
- Built-in Functions
- You can also write your own functions

What? (9 of 9) Classes

- You can define your own data type!
- You can define functions in a class to manipulate it
  - E.g., Define a new data type “Sandwich”

What?

Summary

- Literals
- Variables
- Operators
- Collections
- Expressions and Statements
- If-statement
- Loops
- Functions
- Classes

What?

We will mostly focus on these in this semester

Summary

- Literals
- Variables
- Operators
- Collections
- Expressions and Statements
- If-statement
- Loops
- Functions
- ~~Classes~~

Introduction

- What is computer programming?
- How does a computer take instructions?
- What are the necessary components of a high-level programming language?
- Guidelines for computer science
  - Evaluating a program (Law of computing)
  - Principles and practices

Introduction

- What is computer programming?
- How does a computer take instructions?
- What are the necessary components of a high-level programming language?
- Guidelines for computer science
  - Evaluating a program (Law of computing)
  - Principles and practices

Comparison

Computer Programming

- Focus on writing code
- Creates computer processes

Computer Science

- Includes computer programming, but broader
- Studies computer processes (e.g., develop theories)
  - Data structures
  - Algorithms (e.g., sorting data)
  - Database Theory
  - Developing mathematical models
  - Improving efficiency and performance

Comparison

More info (optional):

- https://ca.indeed.com/career-advice/finding-a-job/computer-programmer-vs-computer-science#:~:text=The%20biggest%20difference%20is%20that,accomplish%20tasks%20using%20applied%20technology.
- https://csweb.rice.edu/academics/graduate-programs/online-mcs/blog/programming-vs-computer-science
- https://medium.com/@adamjgordon24/computer-science-vs-computer-programming-whats-the-difference-5e3764be9532

Computer Programming

- Focus on writing code
- Creates computer processes

Computer Science

- Includes computer programming, but broader
- Studies computer processes (e.g., develop theories)
  - Data structures
  - Algorithms (e.g., sorting data)
  - Database Theory
  - Developing mathematical models
  - Improving efficiency and performance

Introduction

- What is computer programming?
- How does a computer take instructions?
- What are the necessary components of a high-level programming language?
- Guidelines for computer science
  - Evaluating a program (Law of computing)
  - Principles and practices

Law of computing

- For certain problems, there are "Laws" that govern how fast they can be performed
  - For example:
    - There is a limit to how fast can you search a list with data!
    - There is a limit to how fast your GPS map can find a route between point A to point B!
    - There is a limit to how fast you can sort a list of data!
    - You will learn these later as your study computer science.

Law of computing

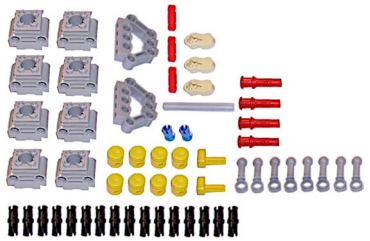
- For certain problems, there are "Laws" that govern how fast they can be performed
  - For example:
    - There is a limit to how fast can you search a list with data!
    - There is a limit to how fast your GPS map can find a route between point A to point B!
    - There is a limit to how fast you can sort a list of data!
    - You will learn these later as your study computer science.
  - The big “O” notation
    - This is one way of evaluating how efficient your programs are

Introduction

- What is computer programming?
- How does a computer take instructions?
- What are the necessary components of a high-level programming language?
- Guidelines for computer science
  - Evaluating a program (Law of computing)
  - Principles and practices

Principles and Practices (1 of 3)

- Use building blocks to build systems



Building blocks

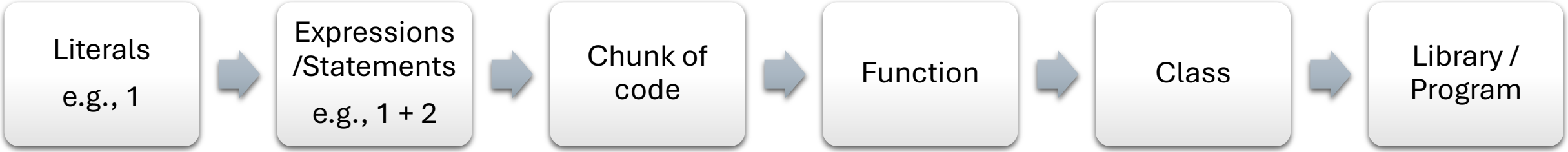
Engine

Car

Racing Event

Principles and Practices (1 of 3)

- Use building blocks to build systems



Principles and Practices (1 of 3)

- Use building blocks to build systems
  - As a result
    - When we approach a problem, we try to discretize everything
      - E.g., Memory into addressable units (bytes/words), Bandwidth is quantized into "packets/frames"
    - Optimize for simple (or most common) tasks
      - E.g., CPUs Core functions: add, sub, move memory, jump/branch; Graphics Cards: Draw triangles really fast!
    - Hierarchical Organization
      - E.g., OSI model of network: frames→ packets → segments→ session →application; File systems: blocks → cluster → file

Principles and Practices (1 of 3)

- Use building blocks to build systems
  - As a result
    - When we approach a problem, we try to discretize everything
      - E.g., Memory into addressable units (bytes/words), Bandwidth is quantized into "packets/frames"
    - Optimize for simple (or most common) tasks
      - E.g., CPUs Core functions: add, sub, move memory, jump/branch; Graphics Cards: Draw triangles really fast!
    - Hierarchical Organization
      - E.g., OSI model of network: frames→ packets → segments→ session →application; File systems: blocks → cluster → file

Important problem-solving skill (Computation thinking): How to break down a problem into smaller ones

The first question of all the labs (except the first one) shows step-by-step how to break down a problem. This is what you should learn from the labs.

This allows you produce code that can achieve your goal

Principles and Practices (2 of 3)

- Docstring/documentation/convention – Make your code more readable!
  - E.g.,
    - How to define the variables
    - How to comment your code
  - As we always develop a project as a team!

  - Interesting reading (optional): https://dzone.com/articles/reading-code-is-a-skill

Principles and Practices (2 of 3)

- Docstring/documentation/convention – Make your code more readable!
  - E.g.,
    - How to comment your code

Important coding habit

We will require you to write documentation

This allows you produce high-quality code that prepares you to become a software developer in industry

Principles and Practices (3 of 3)

- Know “thy user”!!
  - Good software/applications are not about features!
    - It’s about designing software for what the user really needs!

Principles and Practices (3 of 3)

- Know “thy user”!!
  - Good software/applications are not about features!
    - It’s about designing software for what the user really needs!

Though this is not one of the focuses of our course, this is something important that you may learn in future courses

This allows you produce user-friendly software

Outline

- Overview
- Python
- Basics
- String
- Built-in Functions

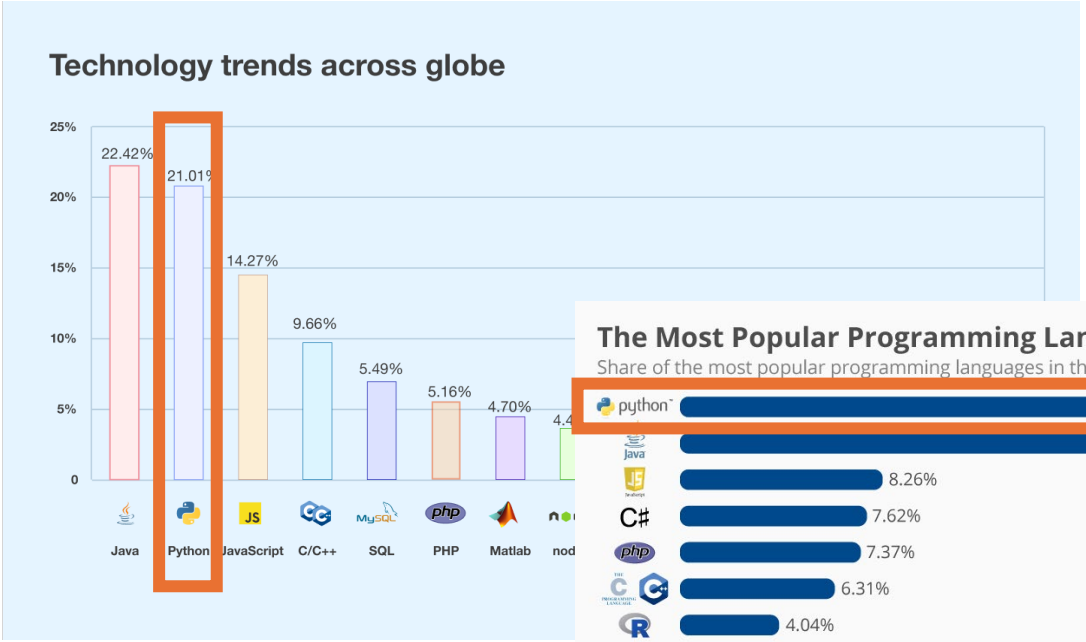
What is Python?

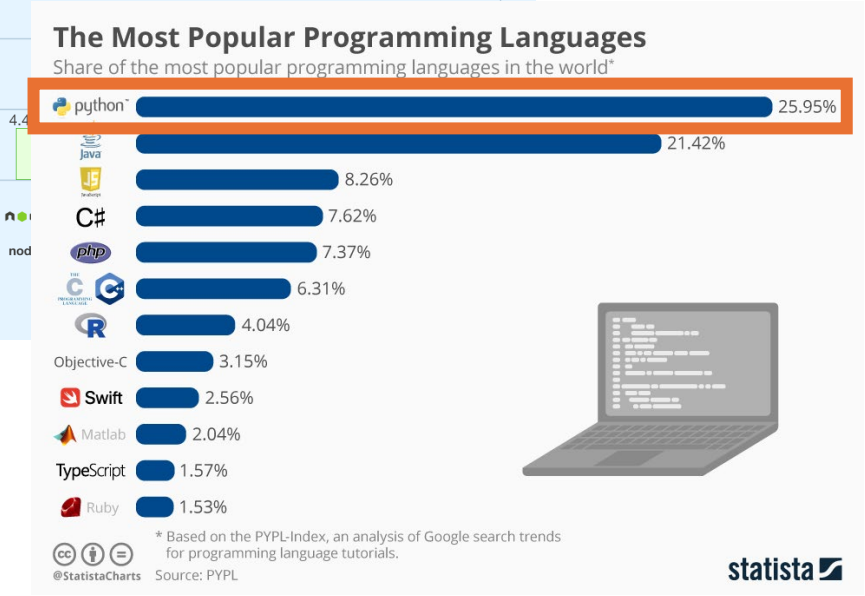
- A high-level programming language

Why Python?

- We need to choose a concrete language to teach you programming
  - We emphasize the commonalities so that you can transfer to other languages (e.g., Java, C/C++)
  - We will also describe things that are specific to Python (i.e., the Pythonic way)
- It is friendly for beginners
- It has a lot of libraries
- It is widely used in many different applications (e.g., data science, robotics)

Why Python?





Python is quickly becoming one of the most popular languages.

Python is a general-purpose language.

Python and IDE

- IDE is short for Integrated development environment

Abc,
Dffg
We
...
Hwryw

Texts

Notepad

Microsoft Word

LibreOffice Writer

Emacs

Text editors

Python and IDE

- IDE is short for Integrated development environment

```
def addition(a, b):
      return a + b

addition(3, 5)
```

Python code

Python Wing IDE
Recommended for beginners

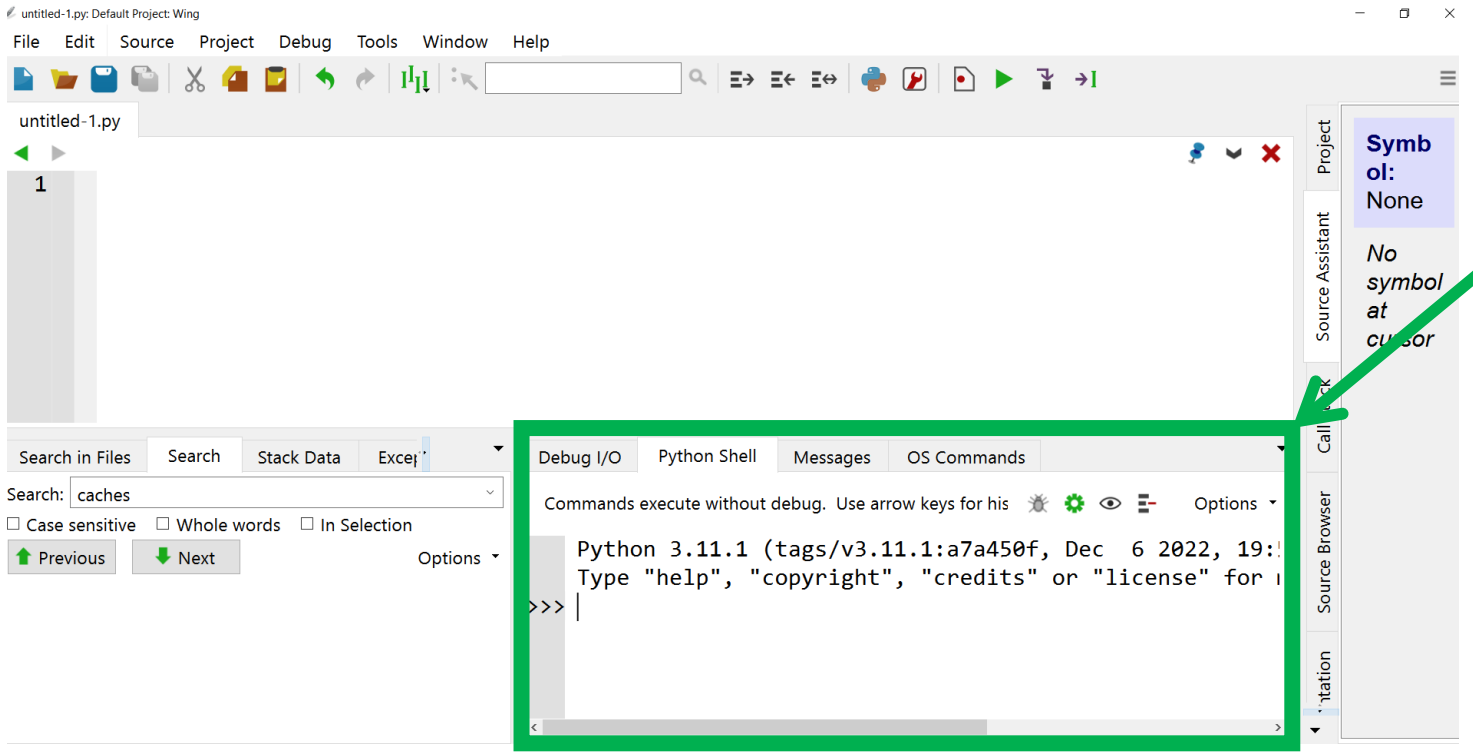
PyCharm

IDE: Code editors

if you have questions about IDE other than PyCharm and Wing, myself and the TAs may not be able to help you

Mode

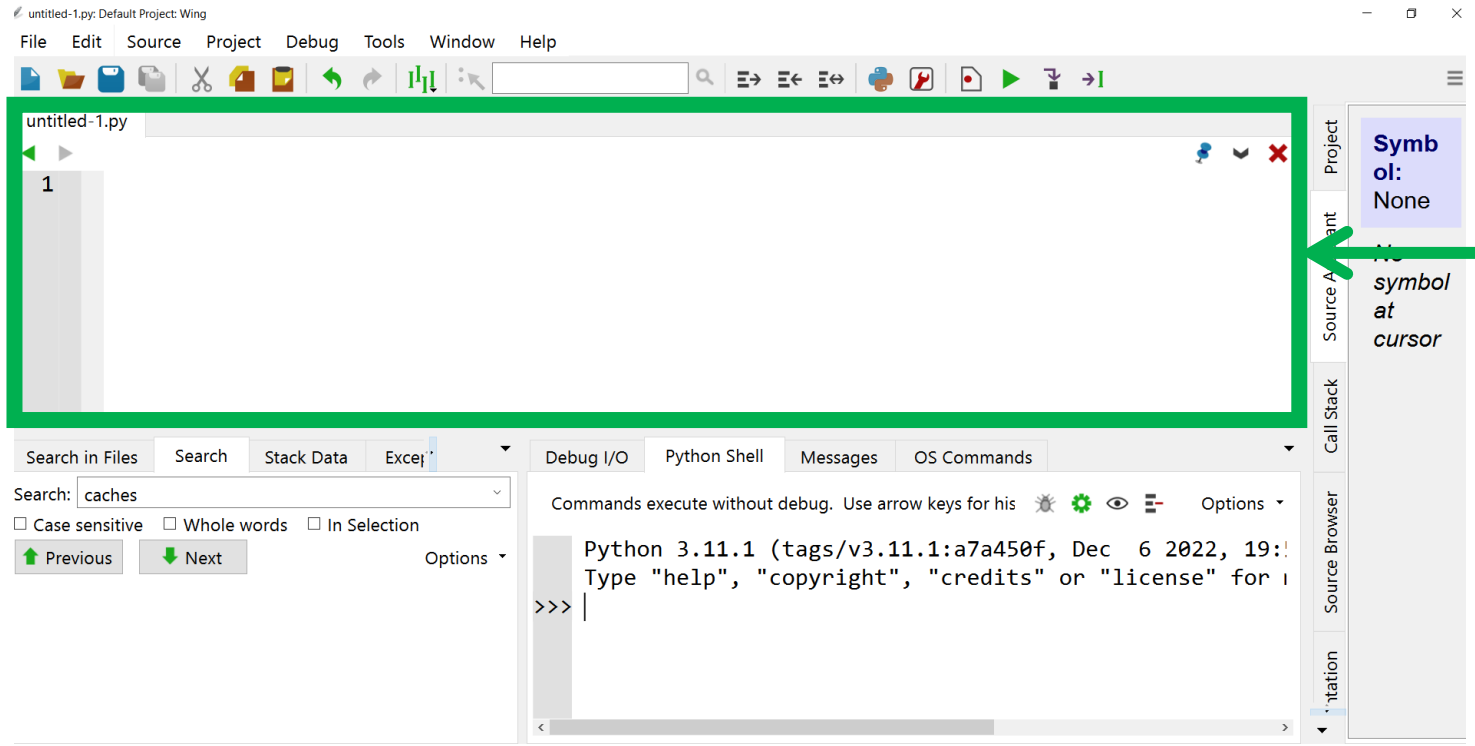
- Wing IDE



Interactive Mode

Mode

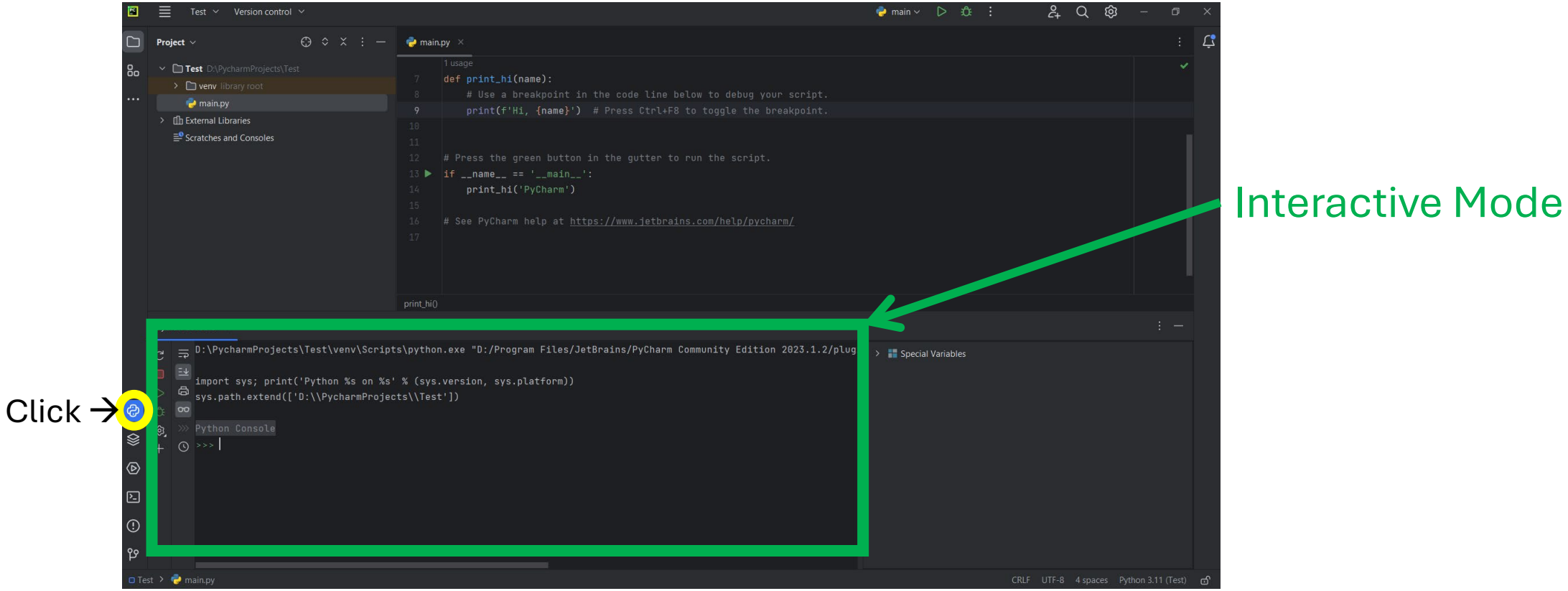
- Wing IDE



Script Mode

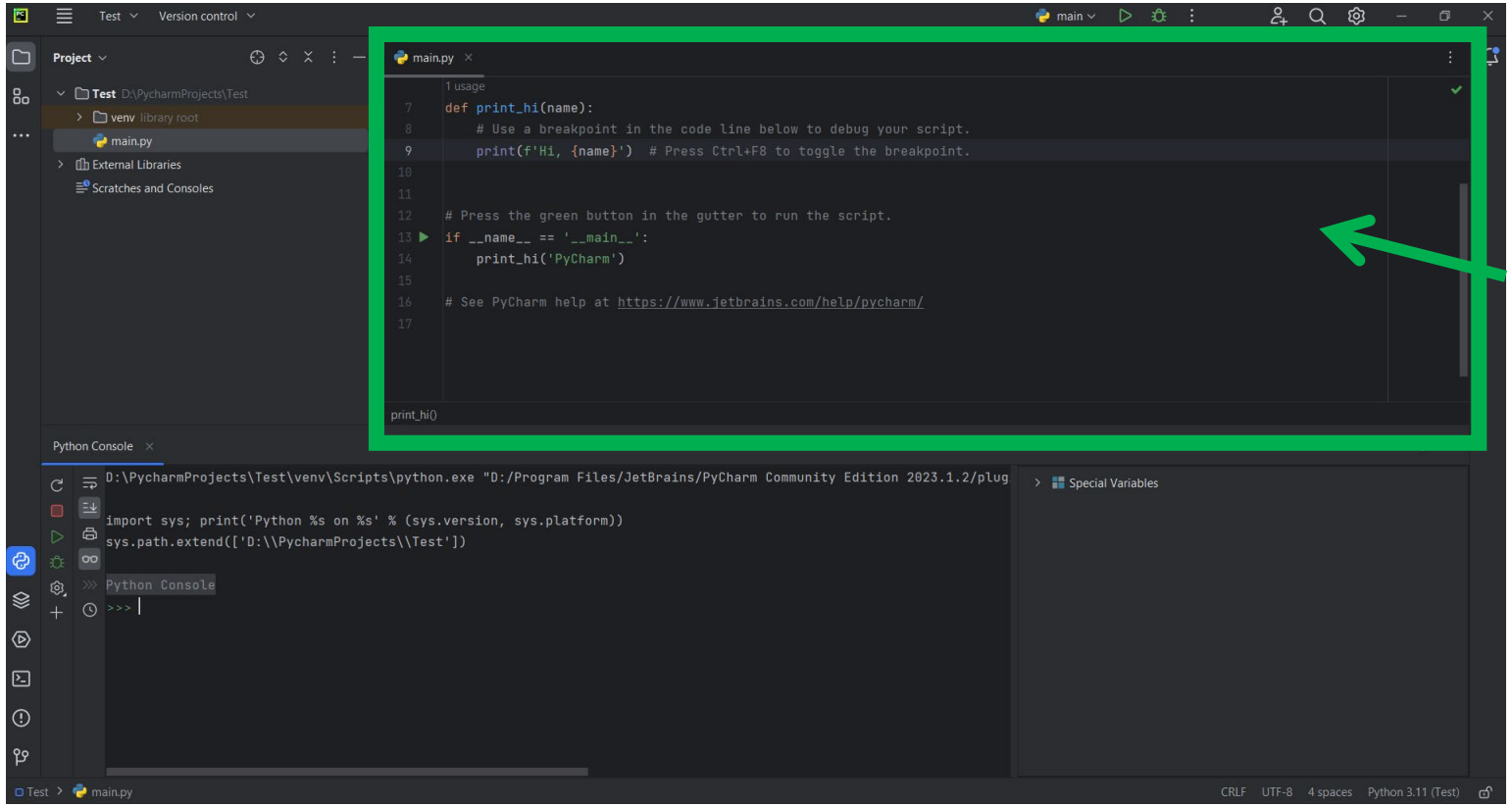
Mode

- PyCharm



Mode

- PyCharm



Script Mode

Script Mode VS Interactive Mode

- Script Mode
  - It will not show the value of the variable in each line
    - It will be terrible for a large program
  - The file is saved and you can reopen it and continue working with it after restarting the IDE
- Interactive Mode
  - It will show you the value immediately after you hit enter
  - The progress is lost if you restart the python shell or the IDE
  - Good for quick testing

Outline

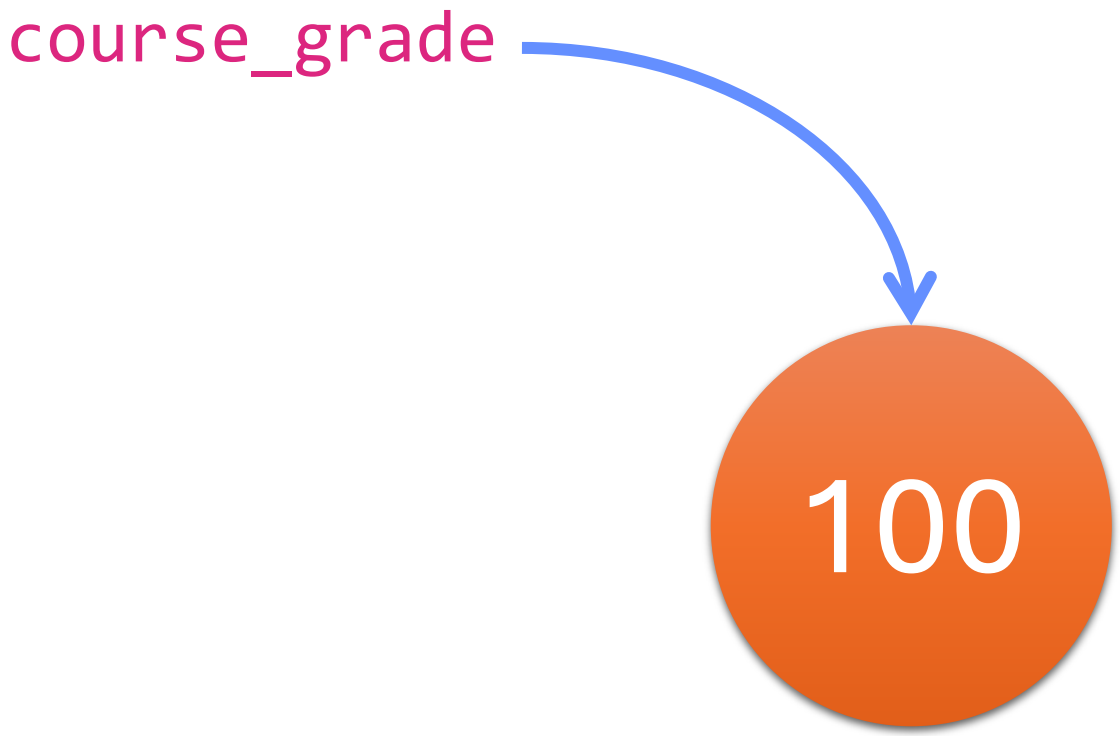
- Overview
- Python
- Basics
- String
- Built-in Functions

Demo: Python as a Calculator

- In interactive mode
- Simple expressions

Variable

- Give a name to a literal



course_grade = 100

Outline

- Overview
- Python
- Basics
- String
- Built-in Functions

Representation

```
>>> 'a'
'a'
>>> "abcde"
'abcde'
>>> "3457SERG@%^dfh"
'3457SERG@%^dfh'
```

Representation

- In Python, you can use either single quote or double quote to represent a string
  - E.g., `'abc ed@^#$$ 21'`, `"abc ed@^#$$ 21"`
  - Both are equivalent
  - You can use any alphabet (case-sensative), numbers or special characters

Outline

- Overview
- Python
- Basics
- String
- Built-in Functions

Statements

- So far, we talked about expression (including assignments), that only consists of identifiers (e.g., variables), literals and operators
- Statements is something more general, and can include things like calling functions

Functions

- Instructions
  - Take 2 pieces of white bread
  - Pick the lid up from the peanut butter jar
  - Take a butter knife and stick it inside the peanut butter jar
  - With the knife, scoop some peanut butter out of the inside of the jar
  - Spread your scoop of peanut butter onto the face of one of your pieces of bread with a knife
  - Squeeze some jelly onto the other piece of bread
  - Spread the jelly on the bread with the butter knife
  - Put your pieces of bread peanut butter and jelly sides together

  - Take 2 pieces of white bread
  - Pick the lid up from the Nutella jar
  - Take a butter knife and stick it inside the Nutella jar
  - With the knife, scoop some Nutella out of the inside of the jar
  - Spread your scoop of Nutella onto the face of one of your pieces of bread with a knife
  - Squeeze some strawberry jelly onto the other piece of bread
  - Spread the strawberry jelly on the bread with the butter knife
  - Put your pieces of bread Nutella and strawberry jelly sides together

  - Take 2 pieces of white bread
  - Pick the lid up from the humus jar
  - Take a butter knife and stick it inside the humus jar
  - With the knife, scoop some humus out of the inside of the jar
  - Spread your scoop of humus onto the face of one of your pieces of bread with a knife
  - Squeeze some mayo onto the other piece of bread
  - Spread the mayo on the bread with the butter knife
  - Put your pieces of bread humus and mayo sides together

To make more different sandwiches, without functions, we need to:

- Copy and paste the instructions for making peanut butter and jelly sandwiches
- Manually update the spread

This result in instructions that are

- Long
- Difficult to read (hard to get the overall structure)

Functions

- Instructions
  - Making sandwich (Peanut butter , jelly)
  - Making sandwich (Nutella, Strawberry jelly)
  - Making sandwich (Humus, Mayo)

With functions, we can:

- Reuse instructions

This result in instructions that are

- Short
- Easier to read

Example

```
>>> min(1, 3, -10, 3)
-10
```

- `min:` function name
- It takes a couple of arguments
- This function returns the minimum number

Example

```
>>> min(1, 3, -10, 3)
-10
```

Making a function call

- `min:` function name
- It takes a couple of arguments
- This function returns the minimum number

Example

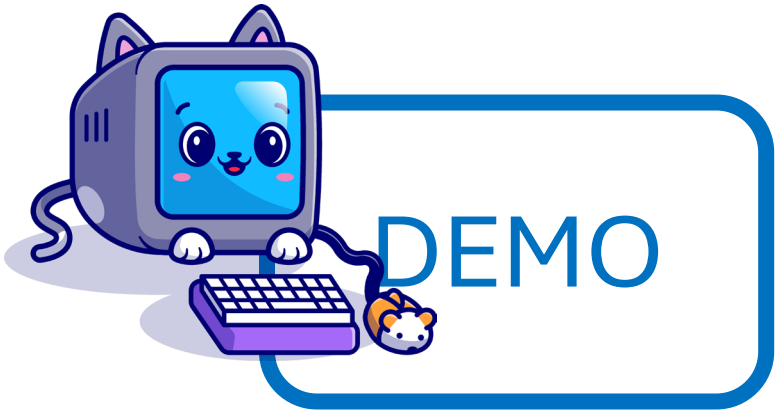
```
>>> a = min(1, 3, -10, 3)
>>> a
-10
```

- The returned value can be assigned to a variable so that you can use it

`input` Function

```
input(prompt=None, /)
```

- What does it do?
  - Get user input
- Parameter
  - `prompt=None`
    - It has one parameter: prompt
    - And it is optional
- Output
  - A str of whatever the user typed



`print` function