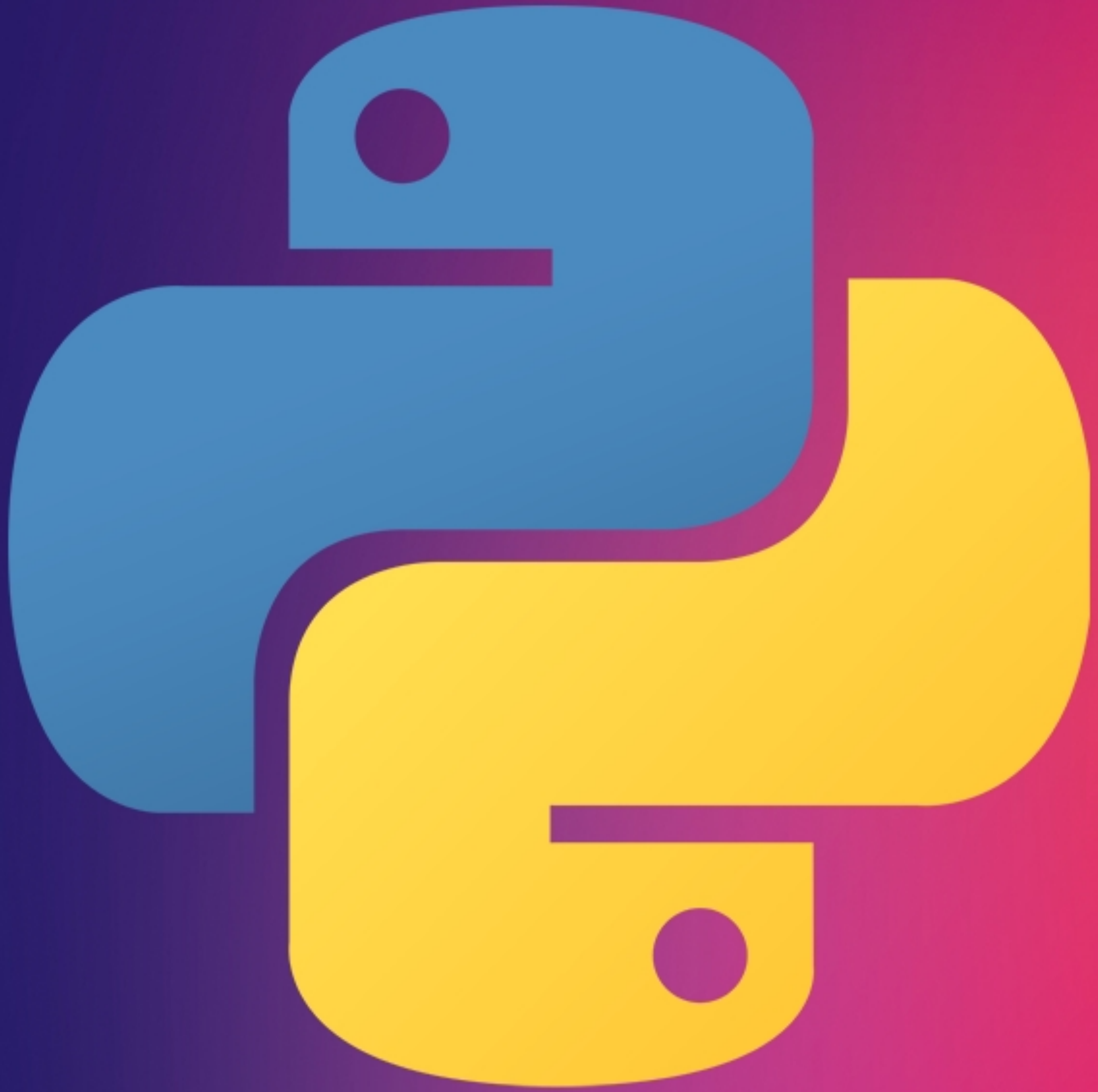
PYTHON

For Beginners

By Aman Guliani



ABOUT THE COURSE

- This course is a beginner introduction to programming using python
- We will learn python syntax, some of its applications and do some practice together
- All syntax will be be according to python 3.x
- By the end of the course, I hope that you will be able to understand a little about programming and the power of python and apply it to your application.

LOGISTICS

- Before we jump in, lets get things setup.
- There are few ways to practice:
  - Online (Do it now)
    - Go to https://colab.research.google.com/ and create a new notebook
  - Local (Do it later)
    - Install python - Instruction https://wiki.python.org/moin/BeginnersGuide/Download
    - Install Anaconda - Instructions https://docs.anaconda.com/anaconda/install/

RESOURCES

- Book I m using and recommend - Python Programming , An introduction to Computer Science by John Zelle

- Google Education python tutorial - https://developers.google.com/edu/python

- Official Python Beginner Guide - https://wiki.python.org/moin/BeginnersGuide

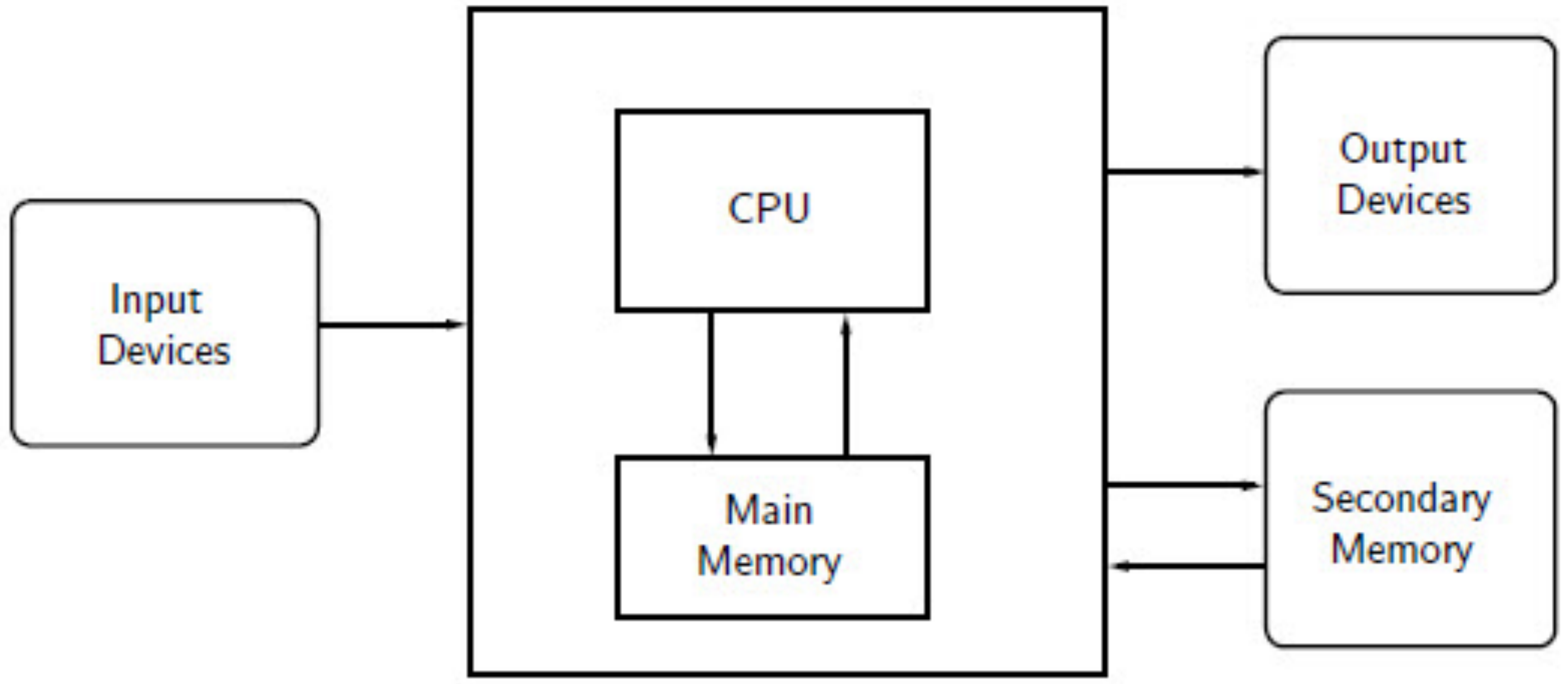
- And many more…. (stackoverflow.com, github.com, python.org)

LETS GO...

VERY QUICK INTRO TO COMPUTERS

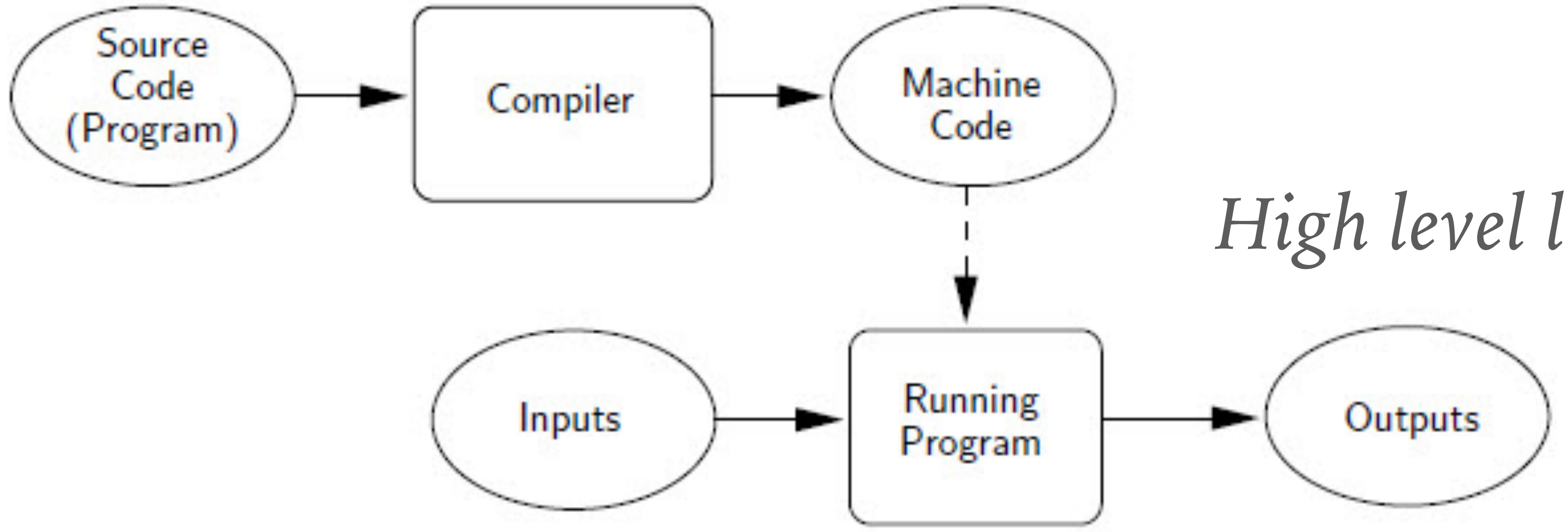
At core the computer understands 0 and 1s

CPU understands machine instruction which it then converts to 1s and 0s



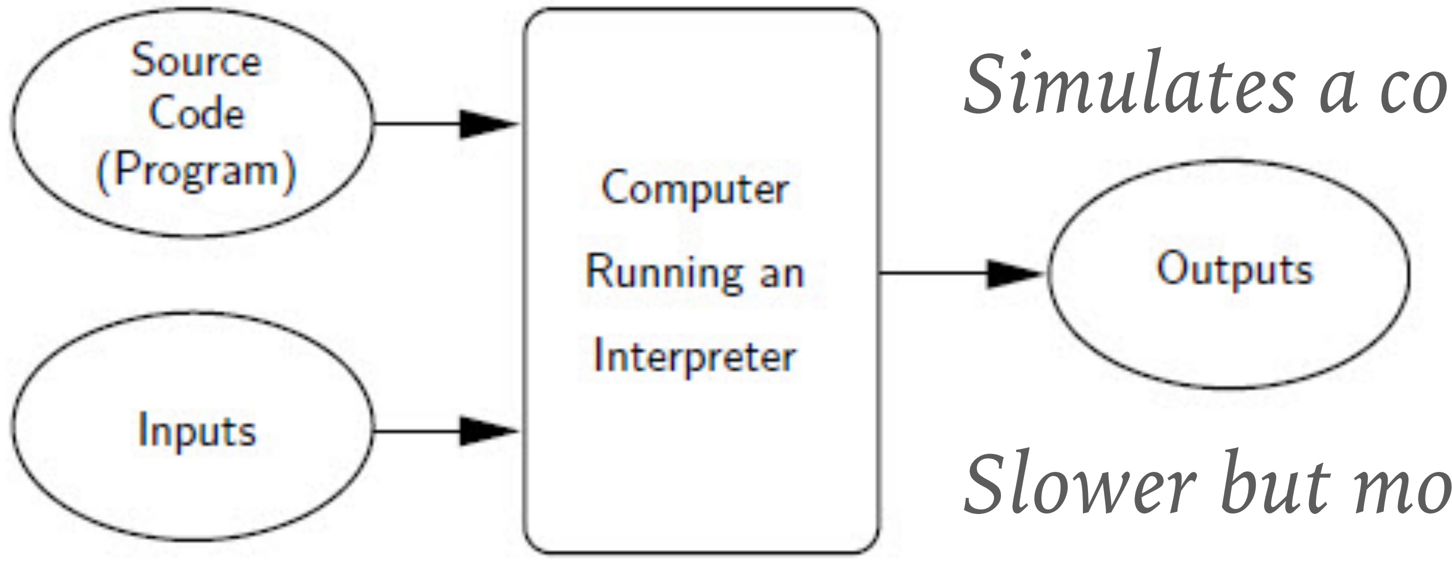
Programming languages allow us to write more of a language which then gets converted to machine instructions

It does this by understand a form (syntax) and meaning (semantics)



High level like C++ Java

Python →



Simulates a computer

Slower but more flexible

LET'S TRY SOME SYNTAX !

- Open your Notebooks , +Code

```
print (“Hello World”)
```

- Run !
- Another Example

```
# This is a function
def hello():
  print (“Hey there!”)

hello()
```

- Now modify the function above

```
# This is a function
def hello(name):
  print (“Hey there ”, name)

hello(“YourName”)
```

- Lot to unpack here !!
  - Notebook runs a virtual python env
  - Print tells the interpreter to print that string
  - hello() is a function, its few lines of code that can be run over and again.
  - Mind the indentation
  - When we modified it , it takes a parameter which can then we used as a variable

BEFORE WE DIG IN

- Whats an Algorithm ? Why is it needed?
    - Its like a recipe.
        - Instructions to solve a problem.
        - Think about the process in your head to solve a problem
    - Lets take an example, Can you describe the algorithm for the following
        - Separating white clothes from colored.
        - How about looking up a word in an English dictionary ?
        - How about baking muffins?

BASICS OF A PROGRAM !

- Naming
  - Name according to what it describes.
  - Personal preference I name all lowercase with _ in python , for example var_name.
- Expressions
  - Just like math, think ahead and store data in variables so it can be reused.
  - x = 5; y = x+5
- Output statement
  - Very important to print things, to debug or understand what’s the state of the program, print statement is your friend

VARIABLES & ASSIGNMENT

```
# Variable and Assignment
# <variable> = <expression / value >
my_number = 7

# Lets print my_number , two ways
my_number
print(my_number)

# Lets add something to my original value
my_number + 3

# Did this change your variable
my_number

# Lets try this instead
my_number = my_number + 3
```

CONTROL STATEMENT, LOOPS

```python
# if or else syntax
# if control_statement:
#   logic
# elif control_statement:
#   logic
# else:
# logic

num = 3
if num < 3:
  print("Number is less than 3")
elif num < 5:
  print("Number is between 3 and 5")
else:
  print("Number is greater than 5")
```

```python
# Loops
# for <var> in <sequence>
#   <body>
for num in [1,2,3,4,5]:
  print(num)

# where did range come from
for i in range(10):
  print(i*i)
```

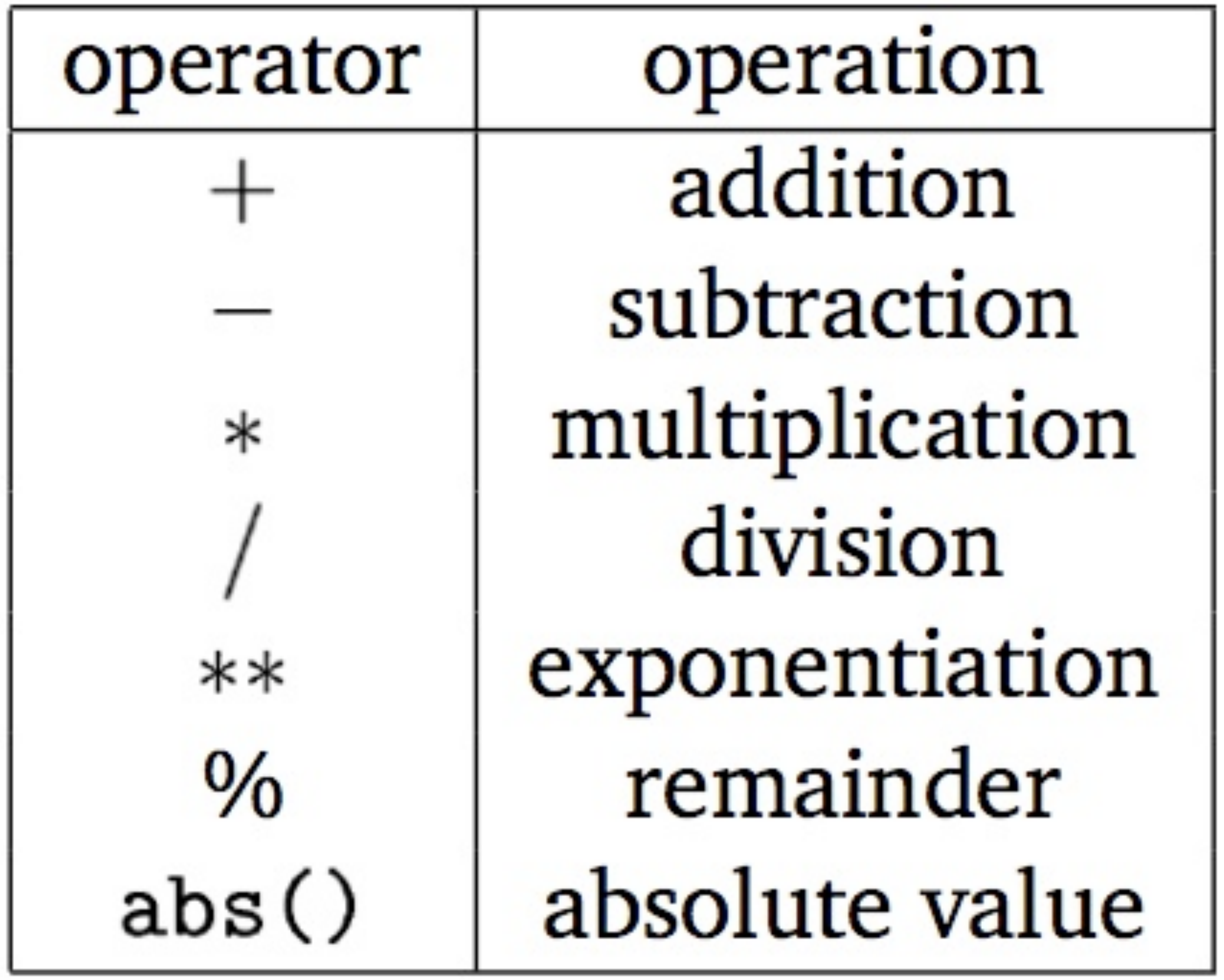
DATA TYPES & NUMBERS

- Classification or Categorization of data, for example, integers, float, strings etc.
- It determines what kind of operations can be performed on that data.
- Numeric Data type, which has a number value, two worth mentioning
  - **Integer:** Positive or negative whole numbers (without a fractional part)
  - **Float:** Any real number with a floating point representation in which a fractional component is denoted by a decimal symbol or scientific notation

```
# Lets use the built in type function
type(1234)
# Expected output <class 'int'>
type(55.50)
# Expected output <class 'float'>
type("hello")
# Expected output <class 'str'>
```

COMPUTING WITH NUMBERS

- Lets try them out in your notebook
- Remember order of expression matters.
  - When we string operators together - Python must know which one to do first
  - For example, try x = 1 + 2 * 3 - 4 / 5 ** 6 and see what value of x do you get.
  - Order is- Parenthesis -> Power -> Multiplication -> Addition (Left to Right)



operator	operation
+	addition
−	subtraction
*	multiplication
/	division
**	exponentiation
%	remainder
abs()	absolute value

HW, SLIDES, SUBMISSION !

- Every week, there will be two part HW.
  - Knowledge Test - multiple choice and long form answers.
  - Coding Test - for which you have to write actual code.
- HW will be posted with the slides in github repo and Google Classroom
- You can use your notebook or write python code on your computer for coding part of the problem.

HOW TO RUN PYTHON CODE ON YOUR COMPUTER

- Use a text editor to open a new file. (Preferable Sublime or Notepad++)
- Write your code and then save your file with some name as <file_name>.py
- Run your code by opening terminal going to directory where you stored your file and run by
  - python <file_name>.py