Programming Principles 1 (#1)

> “Computers are good at following instructions, but not at reading your mind.” –Donald Knuth, the “father of analysis of algorithms”

🔄 Recap of last week

- Course Outline:

AWD-111-Programming-Principles-1-with-Python/Helper Docs at main · Pacific-Design-Academy-CA/AWD-111-Programming-Principles-1-with-Python

This repo contains the assignments and course content for AWD 111: Programming Principles 1 with Python - AWD-111-Programming-Principles-1-with-Python/Helper Docs at main · Pacific-Design-Academy-C...

https://github.com/Pacific-Design-Academy-CA/AWD-111-Programming-Principles-1-with-Python/tree/main/Helper%20Docs

- GitHub:
  - Create a GitHub account:

GitHub: Where the world builds software

GitHub is where over 83 million developers shape the future of software, together. Contribute to the open source community, manage your Git repositories, review code like a pro, track bugs and features, power your CI/CD and DevOps workflows, and secure code before you commit it.

https://github.com/

  - Course link:

AWD-111-Programming-Principles-1-with-Python/Course Content/week-1 at main · Pacific-Design-Academy-CA/AWD-111-Programming-Principles-1

This repo contains the assignments and course content for AWD 111: Programming Principles 1 with Python - AWD-111-Programming-Principles-1-with-Python/Course C 1 at main · Pacific-Design...

https://github.com/Pacific-Design-Academy-CA/AWD-111-Programming-Principles-1-with-Python/tree/main/Course%20Content/week-1

- Resource Handbook:

AWD-111-Programming-Principles-1-with-Python/Resource-handbook.md at main · Pacific-Design-Academy-CA/AWD-111-Programming-Principles-1-with

In this class we will be using a lot of online resources, which includes websites, videos, software documents, etc. This resource handbook will contain links to the resources wi be used during this semester. We will update the handbook on the go,

https://github.com/Pacific-Design-Academy-CA/AWD-111-Programming-Principles-1-with-Python/blob/main/Helper%20Docs/Resource-handbook.md

- Install Python:

Download Python

Information about specific ports, and developer info Source and binary executables are signed by the release manager or binary builder using their OpenPGP key.

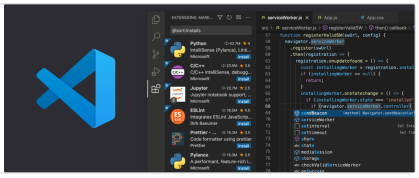
https://www.python.org/downloads/

- Install VSCode:

Visual Studio Code - Code Editing. Redefined

Visual Studio Code is a code editor redefined and optimized for building and debugging modern web and cloud applications. Visual Studio Code is free and available on your favorite platform - Linux, macOS, and Windows.

https://code.visualstudio.com/



⚖️ Python Basic Syntax and Rules

- **Variables :** A Python variable is **a reserved memory location to store values**

```
# Creating variable
a = 10
```

- Semicolon is optional

```
# No Semicolon
print(“Hello”)
# Using Semicolon
print(“Hello”);
```

- **Indentation:** Indentation is a style of code writing to structure program. It is used to structure the code of function, loop or blocks. It increase code readability and represents the flow of the program. See an example given below, indented by tabs

```
# Indentation
a = 10
if a<20:
    print("a is less than 20")
else:
    print("a is greater than 20")
```

- **Python Import:** import is a keyword in python which is used to link module in the program

```
# Single Import
import sys
# Multiple Import
import sys,os
```

- **Python function:** a group of related statements that performs a specific task

```
def greet(name):
    """
    This function greets to
    the person passed in as
    a parameter
    """
    print("Hello, " + name + ". Good morning!")

greet('Paul')
```

🔑 Python Keywords and Identifiers

Keywords are the reserved words in Python.

We cannot use a keyword as a variable name, function name or any other identifier. They are used to define the syntax and structure of the Python language.

False	await	else	import	pass
None	break	except	in	raise
True	class	finally	is	return
and	continue	for	lambda	try
as	def	from	nonlocal	while
assert	del	global	not	with
async	elif	if	or	yield

Rules for writing identifiers,

- 1. Identifiers can be a combination of letters in lowercase **(a to z)** or uppercase **(A to Z)** or digits **(0 to 9)** or an underscore `_`. Names like `myClass`, `var_1` and `print_this_to_screen`, all are valid example.
- 1. An identifier cannot start with a digit. `1variable` is invalid, but `variable1` is a valid name.
- Keywords cannot be used as identifiers.

```
global = 1
```

Output

```
  File "<interactive input>", line 1
    global = 1
           ^
SyntaxError: invalid syntax
```

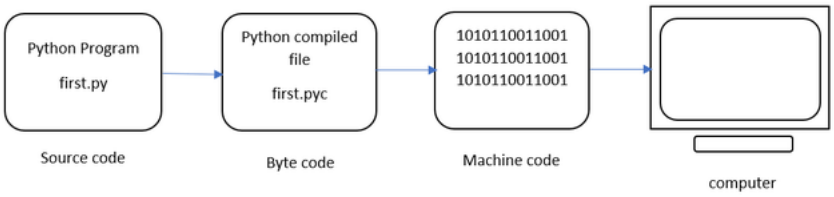
- We cannot use special symbols like **!** , @, #, **$, %,**etc. in our identifier.
- An identifier can be of any length.

🛠️ Execution of a Python Program

Under the hood, execution involves the following steps,

- Compilation: The program is converted into **byte code.**
  Byte code is a fixed set of instructions that represent arithmetic, comparison, memory operations, etc. It can run on any operating system and hardware
- Interpretation:

  The next step involves converting the byte code (.pyc file) into machine code. This step is necessary as the computer can understand only machine code (binary code). Python Virtual Machine (PVM) first understands the operating system and processor in the computer and then converts it into machine code. Further, these machine code instructions are executed by processor and the results are displayed.



References:

- https://www.geeksforgeeks.org/understanding-the-execution-of-python-program/
- https://www.programiz.com/python-programming/function#:~:text=In Python%2C https://www.programiz.com/python-programming/keywords-identifier
- https://www.javaexercise.com/python/python-basic-syntax