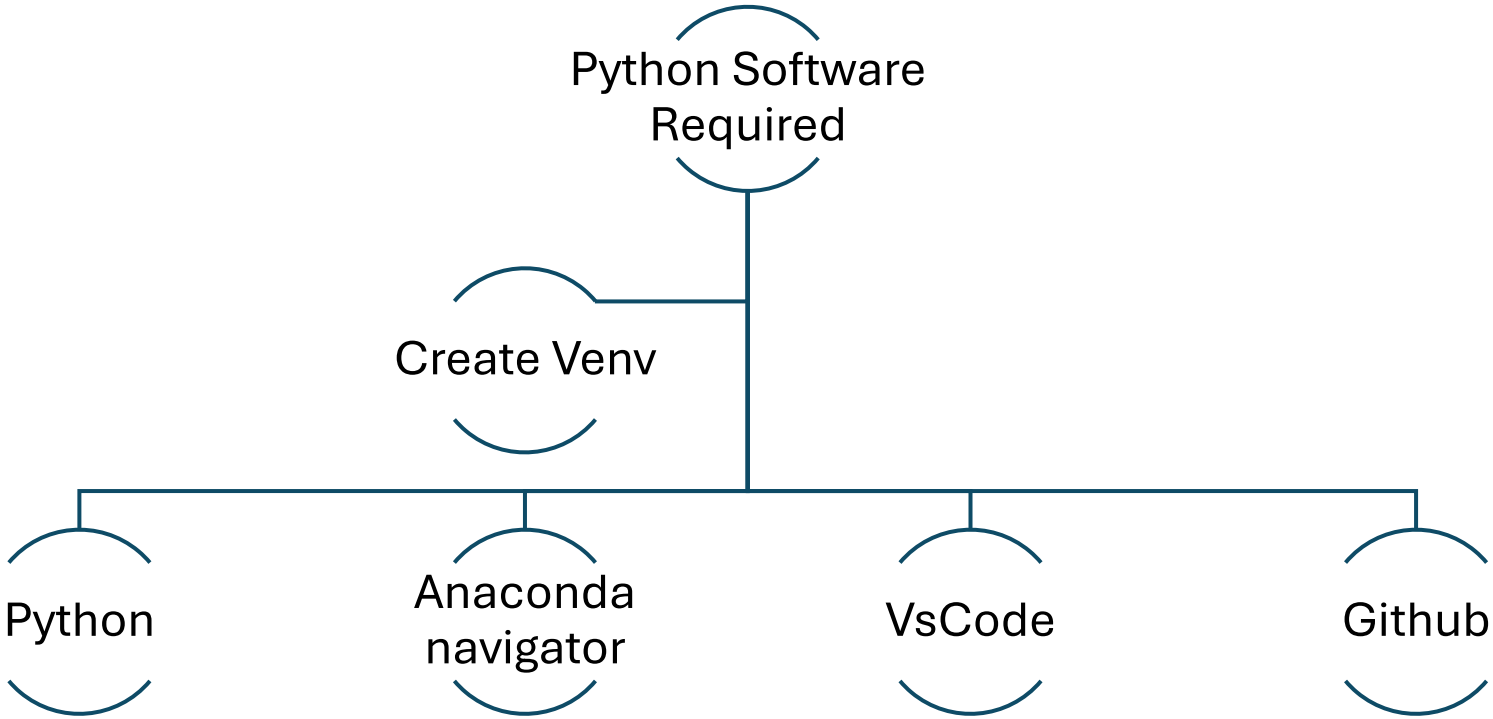
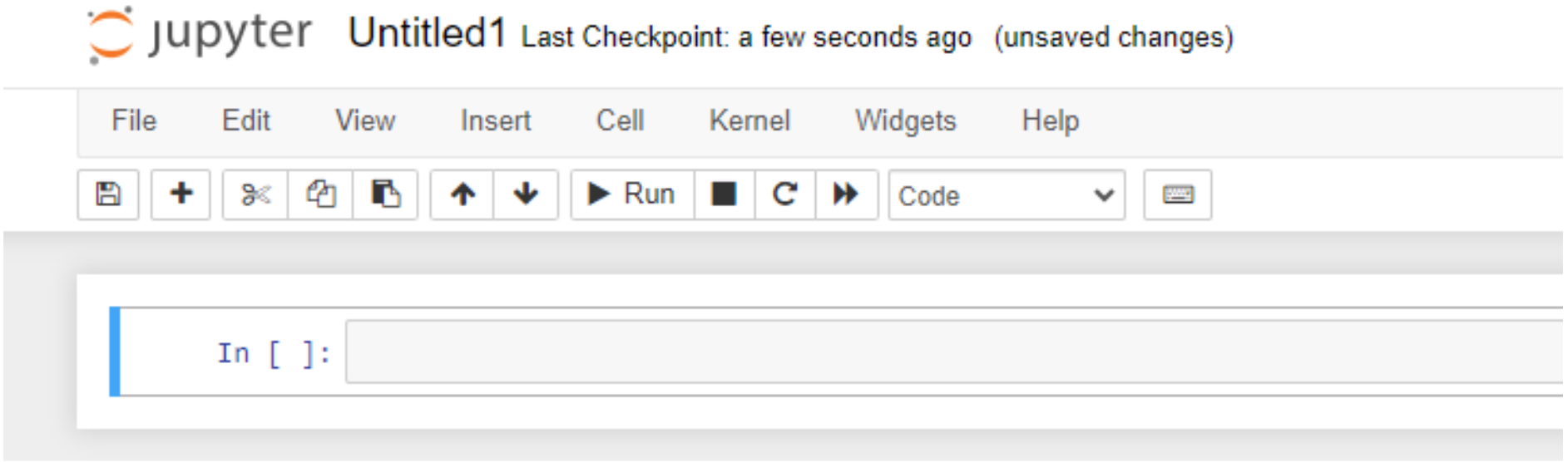
Python


Python Software Required
Create Venv
Python
Anaconda navigator
VsCode
Github

How To Use Jupyter Notebook



Select cell + press b => create Cell below your Cell
Select cell + press x => delete Cell above your Cell
Select cell + press c => copy your Cell
Select cell + press v => past your Cell

How To Use GitHub

Create Repository

Step 1
Press your profile picture
Step 2
Choose your repositories
Step 3
Press new
Step 4
Enter Repository Name and make sure your repository is public and add readme file
Step 5
Press create repository

Delete Repository

Step 1
Select Settings
Step 2
Scroll Down and Select Delete this repository

Upload File in Repository

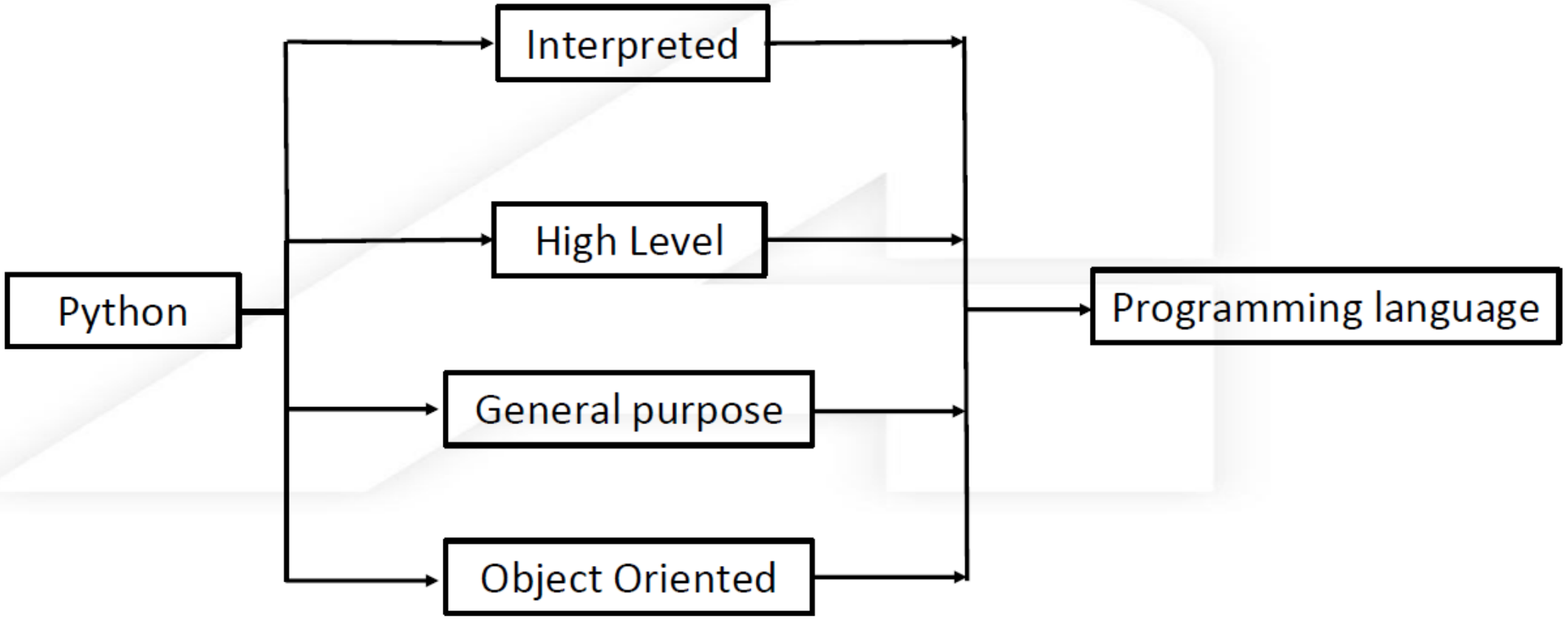
Step 1
Select Add File and Upload Files
Step 2
Select Choose your Files

Delete File in Repository

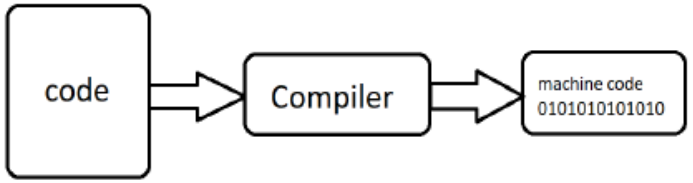
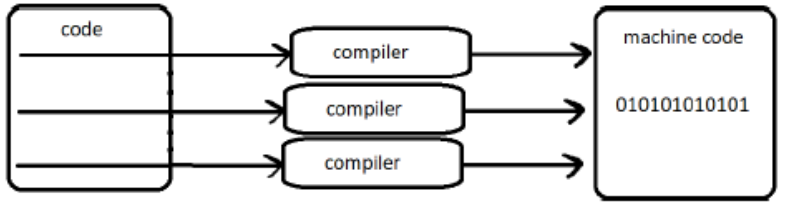
Step 1
Select File
Step 2
Press (...) & Choose Delete File

Introduction to python

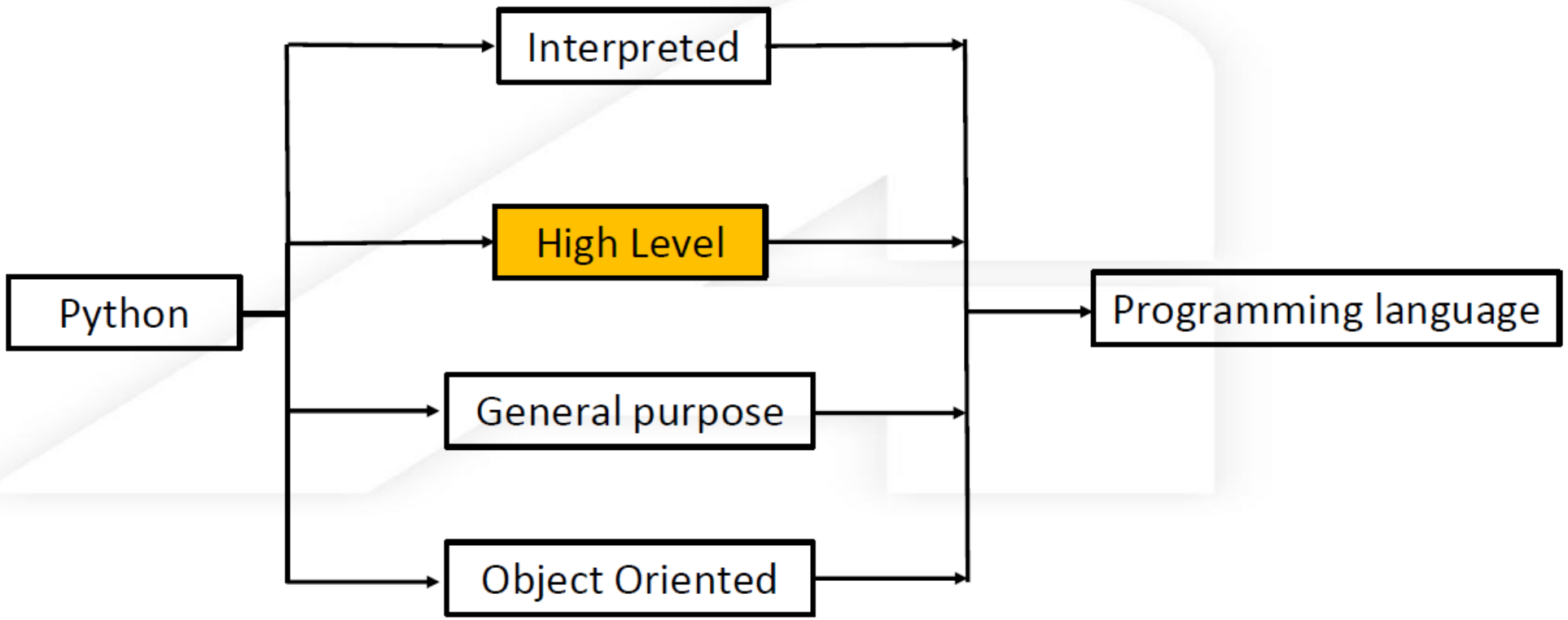
What is Python ?



Types Of Programming languages

	Compiled programming language	Interpreted programming languages
Compilation Process	Code is translated into machine code or an intermediate code by a compiler before execution. 	Code is translated and executed line by line by an interpreter during runtime. 
Execution Speed	Generally faster execution as the entire code is translated into machine code beforehand.	Generally slower compared to compiled languages.
Debugging	Errors are detected during the compilation process, making debugging more challenging.	Errors are identified during runtime, making it easier to pinpoint and fix issues.
Memory Usage	more efficient memory usage	May result in less optimized memory usage compared to compiled languages
Example	C , C++ , C#	**Python** , Ruby

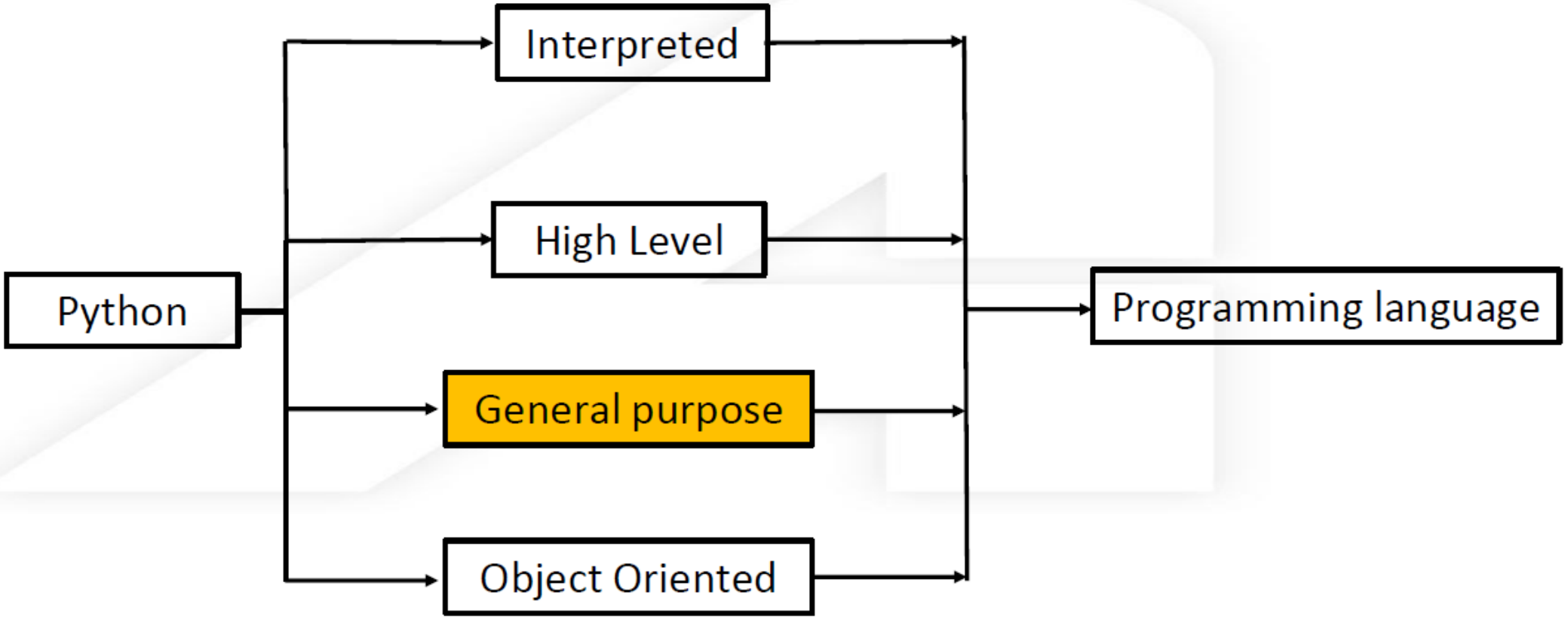
What is Python ?



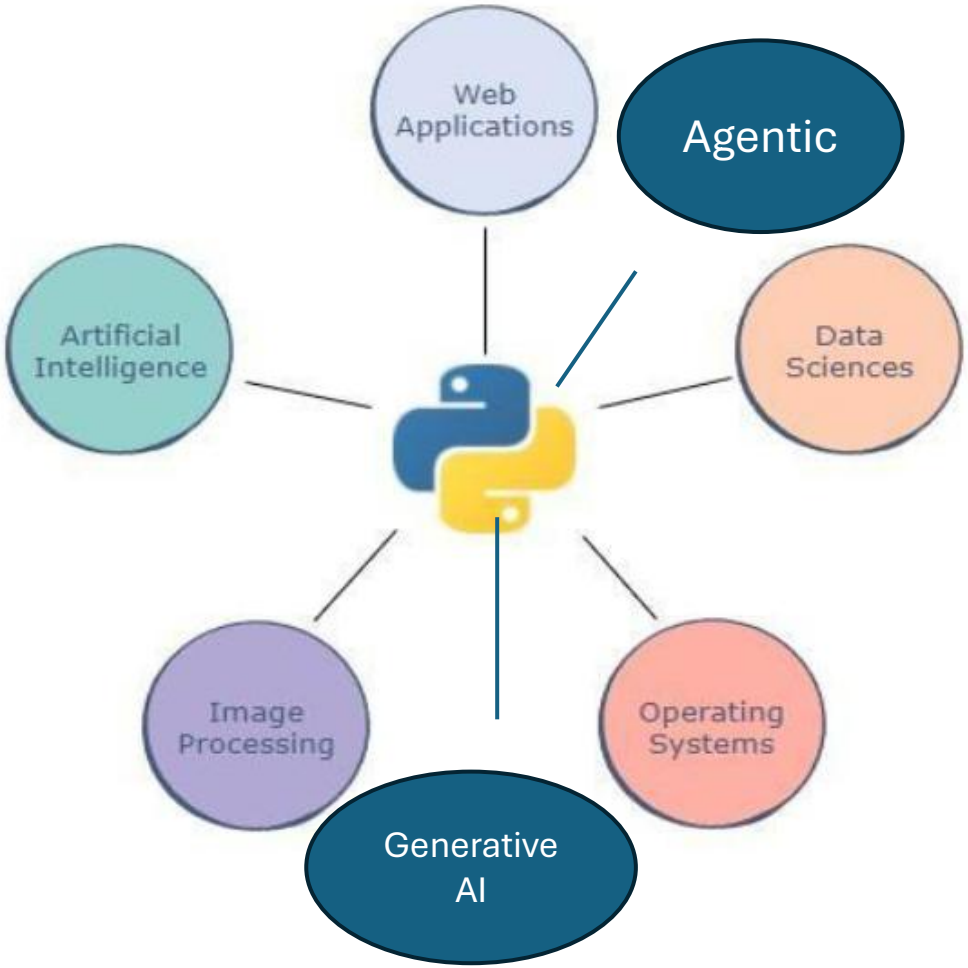
What is Python ?

Programming languages Levels		
	Low Level Language	**High level language**
Definition	- These languages are like talking directly to the computer's hardware. - They're a basic set of instructions that the computer easily understands.	- programming language that's easier for humans to understand. - They use words and structures that resemble everyday language, making programming more user-friendly.
Example	assembly languages and machine code.	**Python**, Java, C++, Ruby, Swift

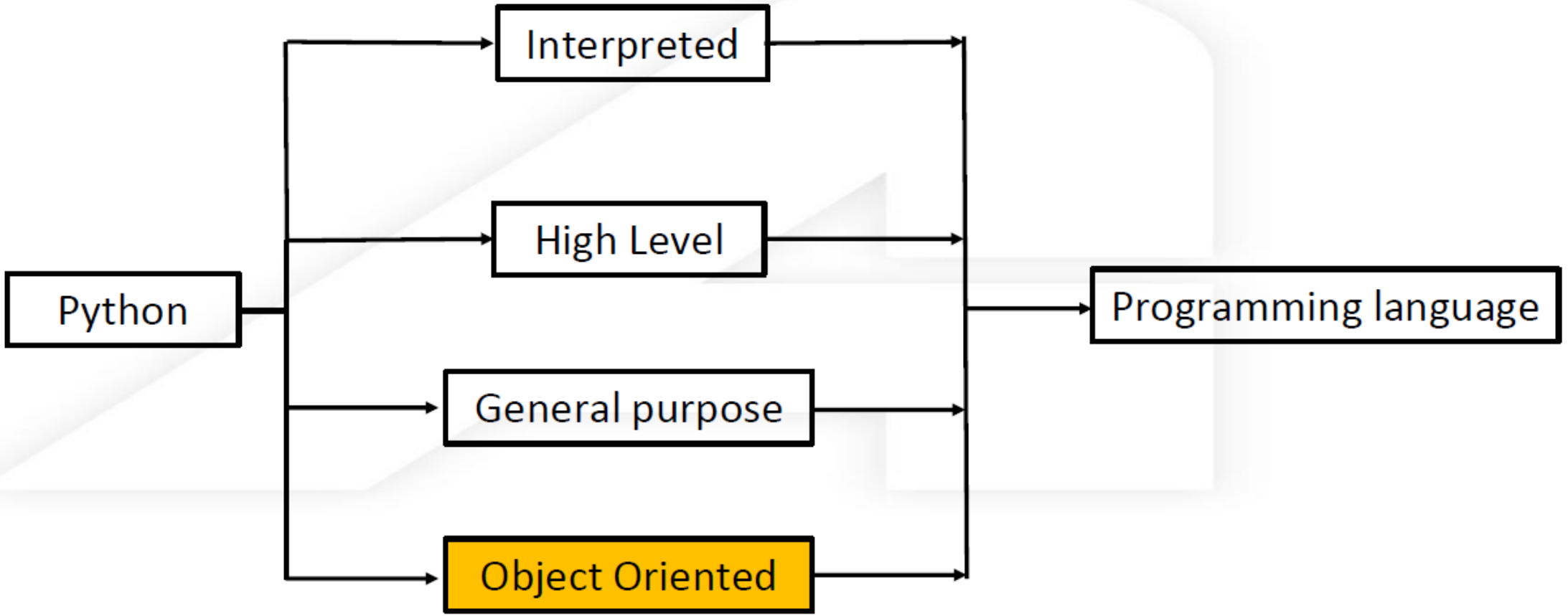
What is Python ?



What is Python ?



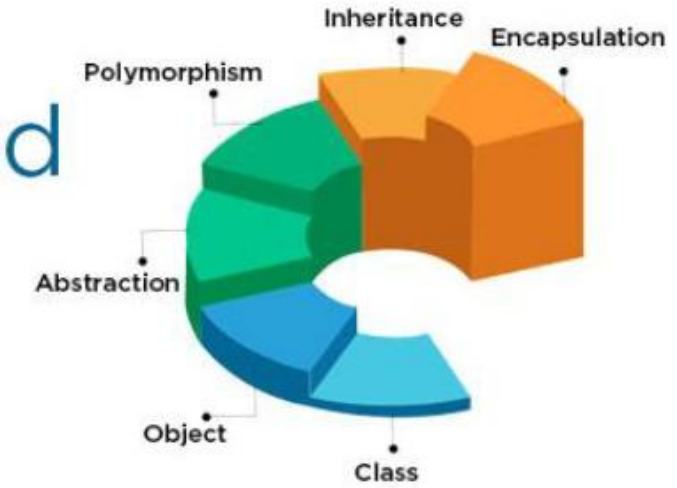
What is Python ?



Object-oriented programming (OOP)

Python is a great programming language that supports Object-oriented programming (OOP), OOP is a way of computer programming using the idea of “objects” to represents data and methods.

Object Oriented Programming with Python

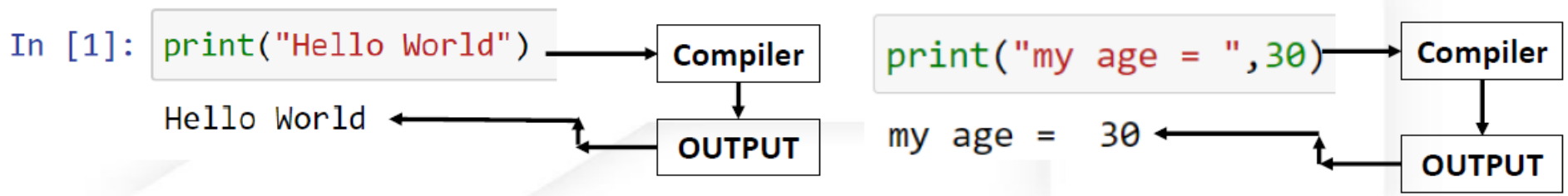


Python Basics

Print() Function

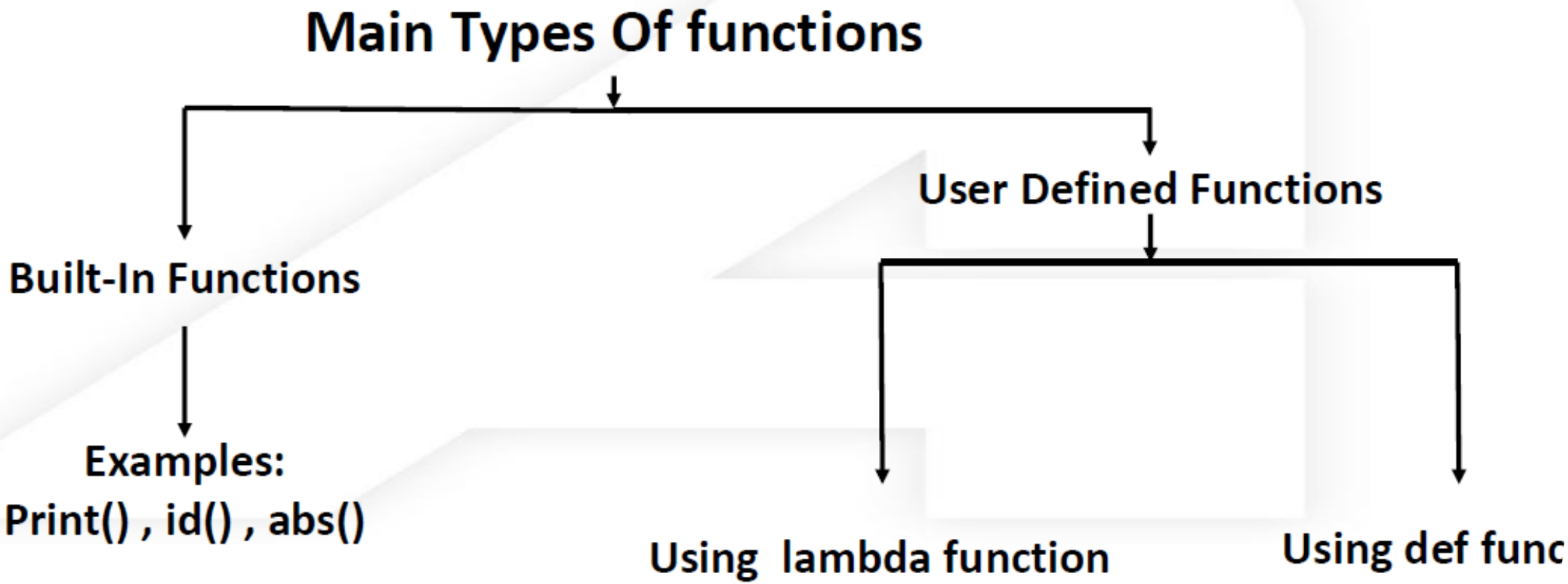
The print() function in Python is a **built-in function** that displays information on the screen.

What is functions



What is function

Functions in Python are blocks of reusable code that perform specific tasks.



Comment:

Comments in Python are text annotations within the code that are ignored during execution, providing explanations or notes for better understanding.

You can insert single-line comments using '#' and multi-line comments using triple single quotes (' ' ').

comment

```
print("hello world")
print("amit learning")
```

```
hello world
amit learning
```

```
print("hello world")
#print("amit learning")
```

```
hello world
```
→ OUTPUT

```
'''
print("hello world")
print("hello world")
print("hello world")
print("hello world")
print("hello world")
print("hello world")
'''
print("amit learning")
```

```
amit learning
```
→ OUTPUT

Try To Solve

What should be the expected output of the provided code?

```
'''
print("amit")
print("learning")
print("python")
'''

print("machine learning")

#print("deep learning")
#print("data science")
#print("python")
```

Try To Solve

What should be the expected output of the provided code?

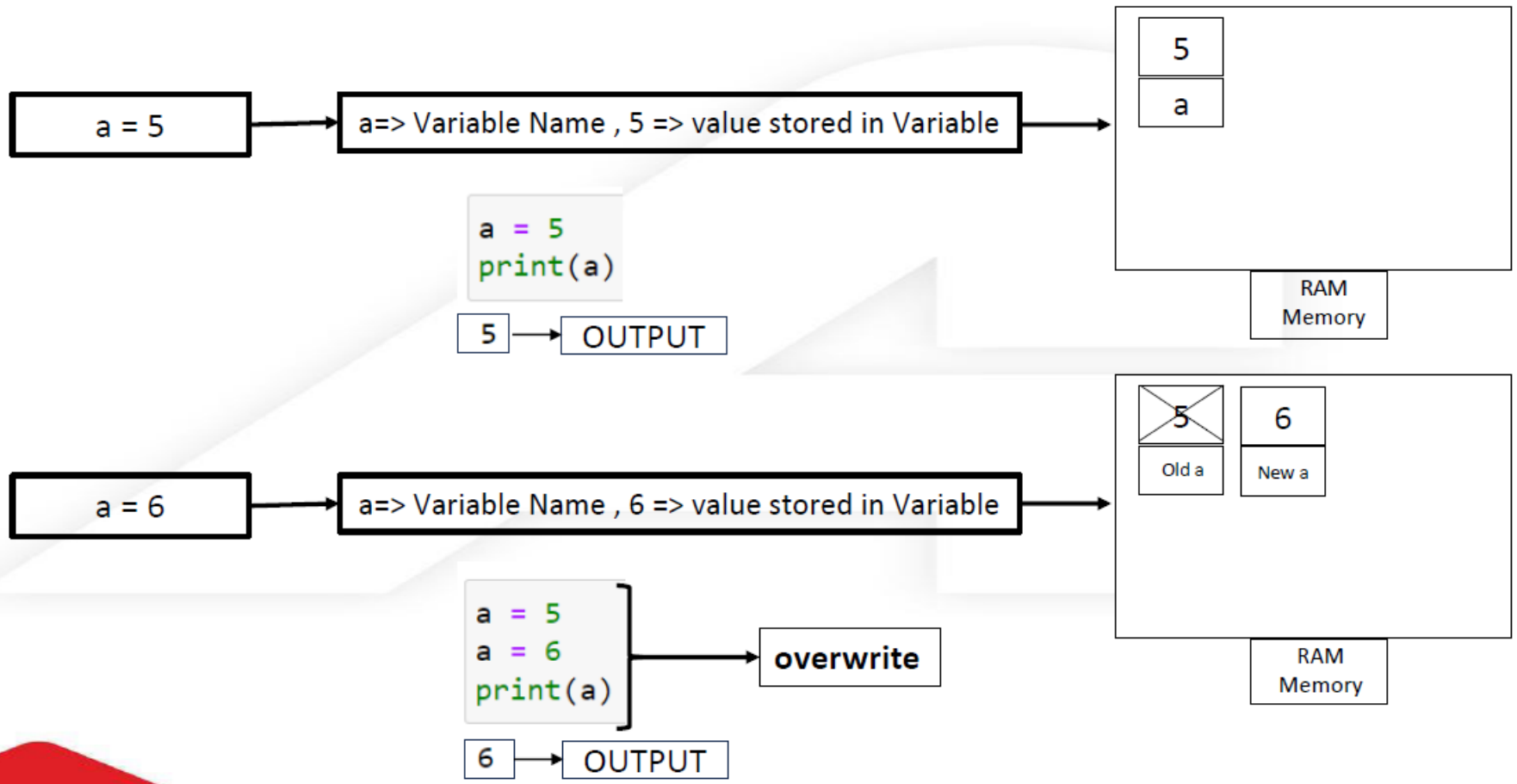
```
'''
print("amit")
print("learning")
print("python")
'''

print("machine learning")

#print("deep learning")
#print("data science")
#print("python")
```

Answer:

- machine learning

Python Variables

Python Variables

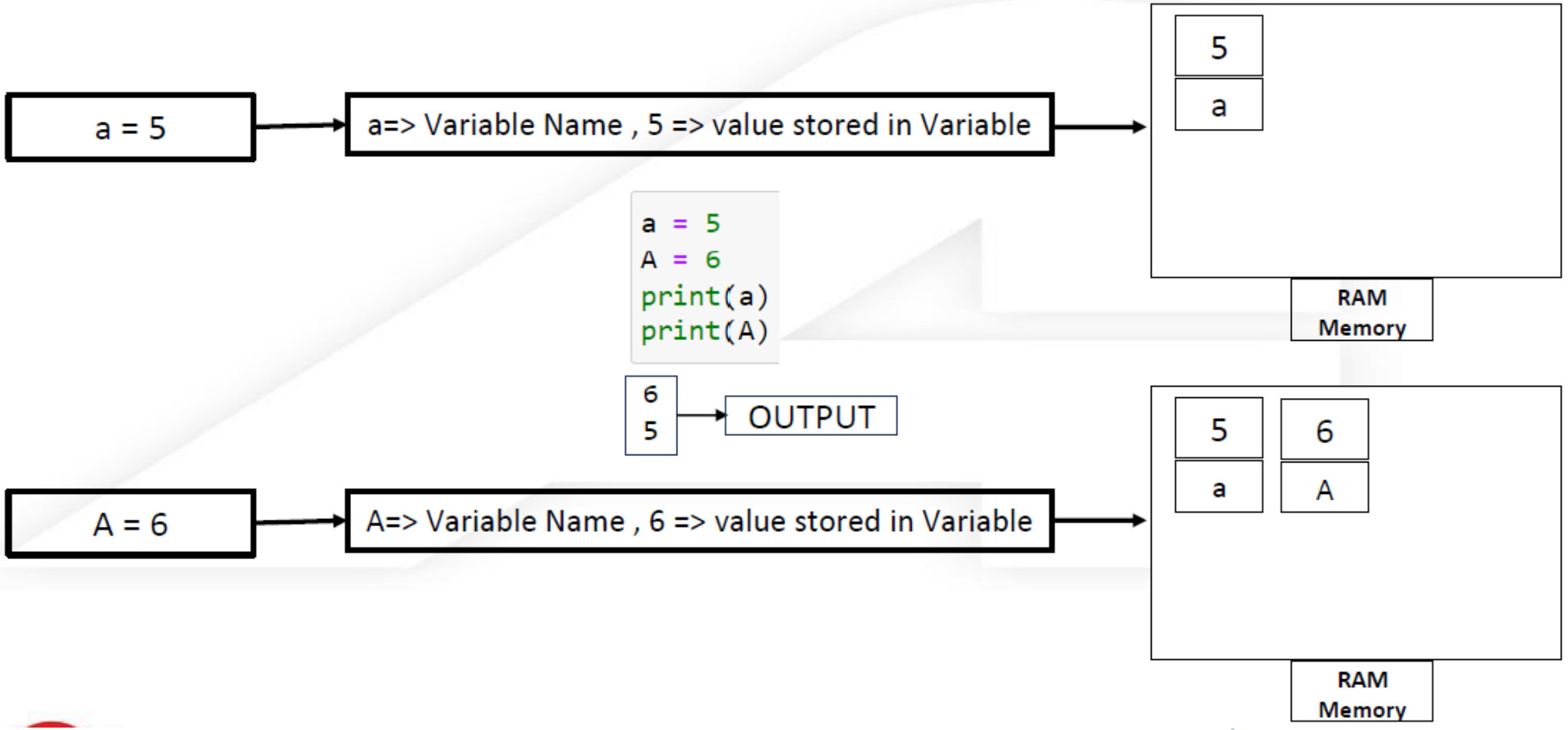


PYTHON IDENTIFIERS

1. Python identifiers are user-defined names for variables, functions, classes, or objects in code.
2. Guidelines for creating identifiers include using letters (uppercase and lowercase), numbers, and underscores.
3. Special characters and operators are not allowed in identifiers.
4. Identifiers should not begin with a number, and certain keywords are reserved and cannot be used as standalone identifiers.
5. Meaningful names for identifiers are encouraged.
6. **Python is case-sensitive,** distinguishing between uppercase and lowercase identifiers.
7. Avoid using 'l', 'I', or 'O' as single-character variable names due to potential font-related confusion.

Class names	Variable names / Methods / Functions / Arguments / Globals	Constants
PascalCase	snake_case	FULLY CAPITALIZED

Python Variables



How To Print value of Variable Inside String

Format function

format function in strings helps you put variables into a text by using curly braces {} as placeholders. You can replace these placeholders with actual values when using the format function, making your text dynamic and customized.

format

```
print("my name is {0} , my age is {1}".format("amit",30))
print("my name is {} , my age is {}".format("amit",30))
print("my name is {x} , my age is {y}".format(x = "amit",y = 30))
```

```
my name is amit , my age is 30
my name is amit , my age is 30
my name is amit , my age is 30
```

```
name = "amit"
age = 30
print(f"my name is {name}, my age is {age}")
```

```
my name is amit, my age is 30
```

Try To Solve

Create a Python script to display personalized information. Utilize variables to store the user's name, age, and the name of the course they are learning. Dynamically generate a message that includes this information and print the result. Ensure clarity in your code implementation without relying on user input.

Expected Output:

My name is {your_name}, and I am {your_age} years old. I am engaged in learning {course_name}.

Try To Solve

Create a Python script to display personalized information. Utilize variables to store the user's name, age, and the name of the course they are learning. Dynamically generate a message that includes this information and print the result. Ensure clarity in your code implementation without relying on user input.

Answer:

```
your_name = "amit learning"
your_age = 25
course_name = "machine learning and AI diploma"

output_text = f"My name is {your_name}, and I am {your_age} years old. I am engaged in learning {course_name}."


print(output_text)
```

Python Data Types

Python Data Types

One variable : One Data

Integer	Float	String	Boolean
whole number without a decimal point.	numeric data type that represents real numbers and can include a decimal point.	sequence of characters, enclosed within single or double quotes, used to represent text data.	binary data type representing either True or False

One variable : Many Data

List	Tuple	Set	Dictionary
1. Data ordered 2. Changeable Data 3. Allow Duplicate Data 4. List can be represented by [ ] 5. Can be nested among all 6. Convert any datatype to list using List() function	1. Data ordered 2. Unchangeable Data 3. Allow Duplicate Data 4. Tuple can be represented by ( ) 5. Can be nested among all 6. Convert any datatype to tuple using tuple() function	1. Data unordered 2. Unchangeable Data 3. Not allow Duplicate Data 4. Tuple can be represented by { } 5. Can be nested among all 6. Convert any datatype to set using set() function	1. Data ordered 2. Changeable Data 3. Not allow Duplicate for keys 4. Dictionary can be represented by { } 5. Can be nested among all 6. Convert any datatype to dictionary using dict() function

Type() function:

used to determine the data type of a variable or value.

Examples:

integer

```
In [11]:
x = 5
print(type(x))
<class 'int'>
```

float

```
In [12]:
x = 5.5
print(type(x))
<class 'float'>
```

string

```
In [13]:
x = "amit learning"
print(type(x))
<class 'str'>
```

boolean

```
In [14]:
x = True
print(type(x))
<class 'bool'>
```

List

```
In [15]:
x = [1,2.5,3.1,4.25,"amit",True]
print(type(x))
<class 'list'>
```

Tuple

```
In [16]:
x = (1,2.5,3.1,4.25,"amit",True)
print(type(x))
<class 'tuple'>
```

Set

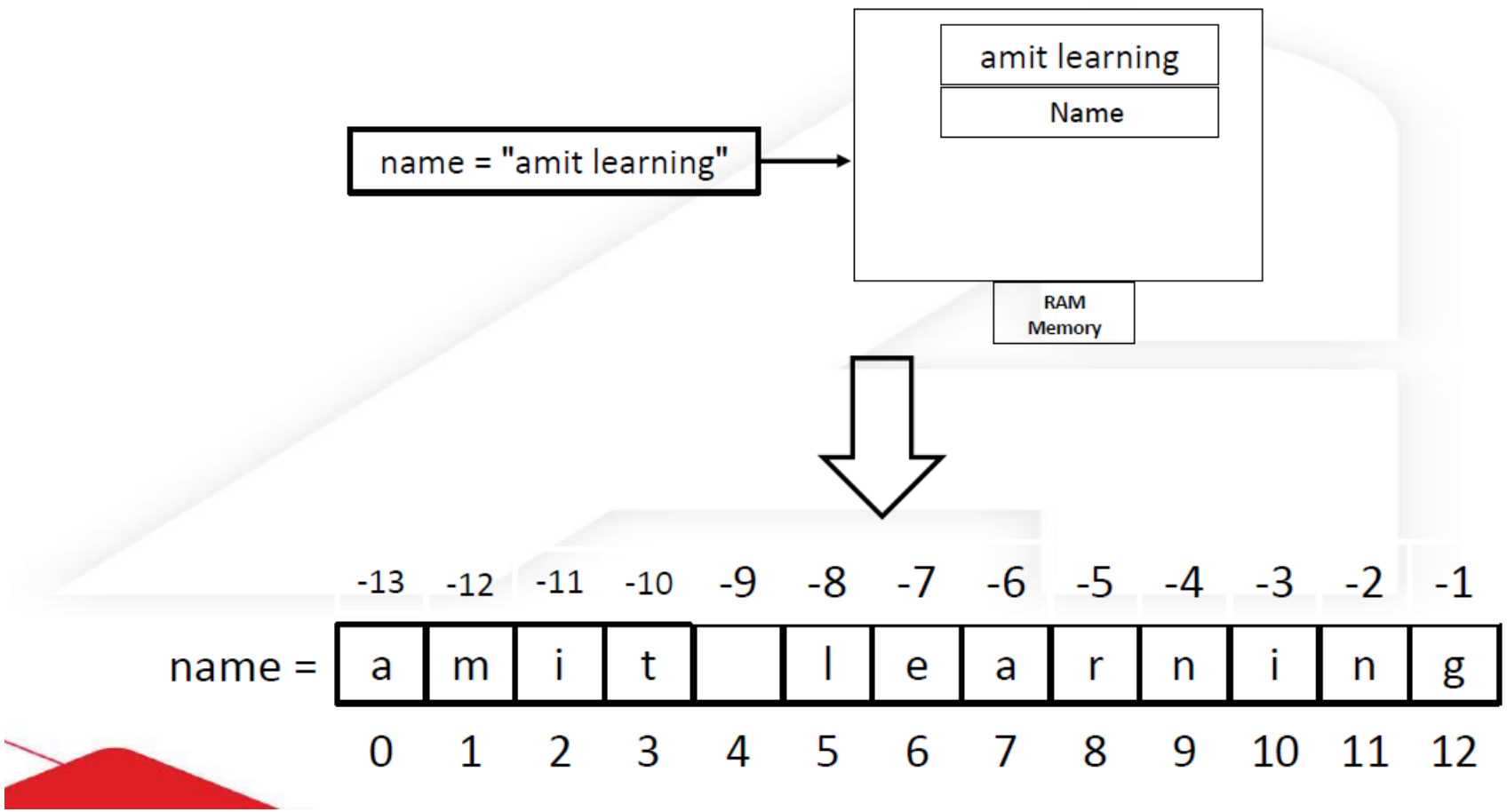
```
In [17]:
x = {1,2.5,3.1,4.25,"amit",True}
print(type(x))
<class 'set'>
```

Dictionary

```
In [18]:
x = {"key1":"value1","key2":"value2"}
print(type(x))
<class 'dict'>
```

String

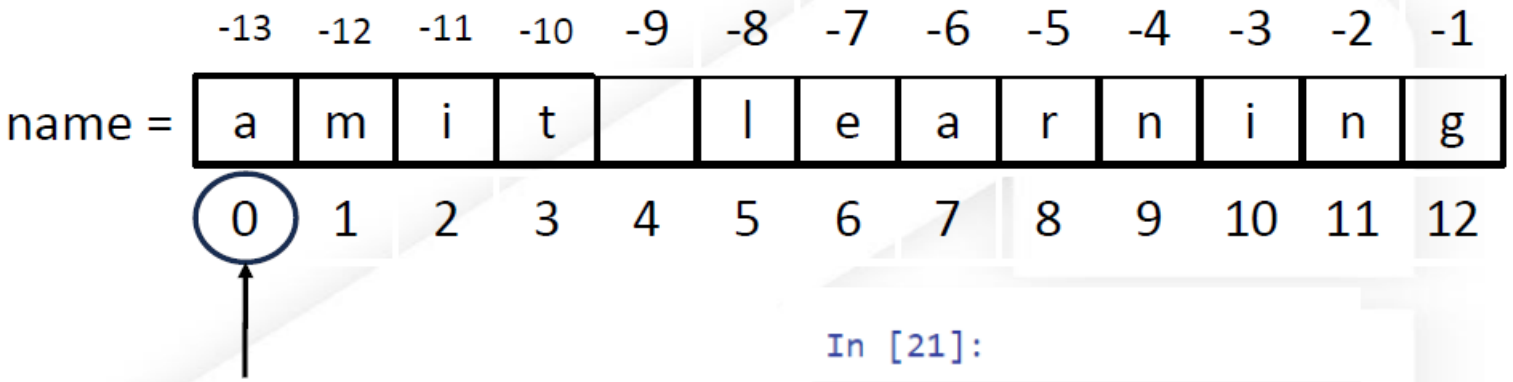
How String Stored in Memory



In [20]:

```
name = "amit learning"
print(name[0])
```

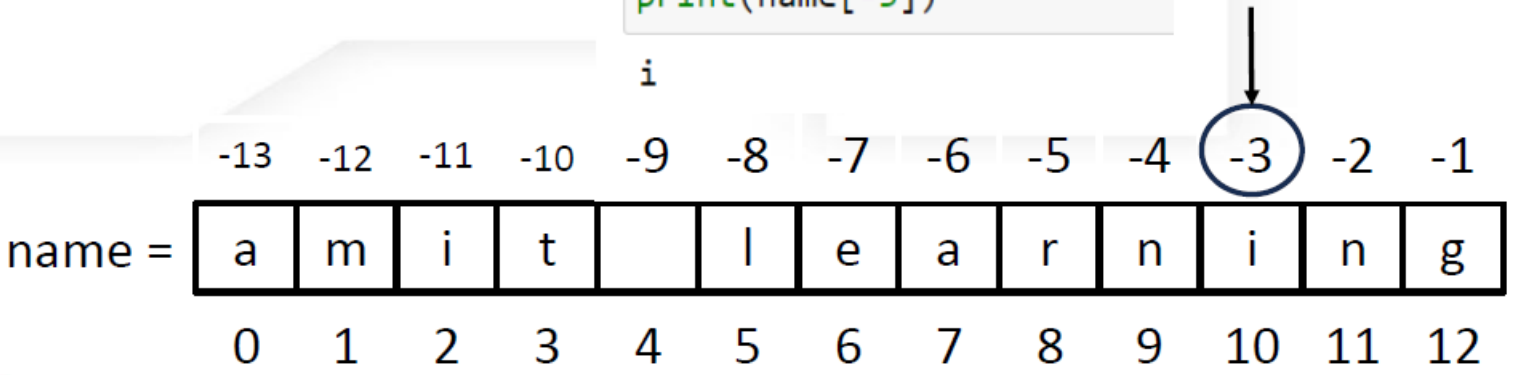
```
a
```



In [21]:

```
print(name[-3])
```

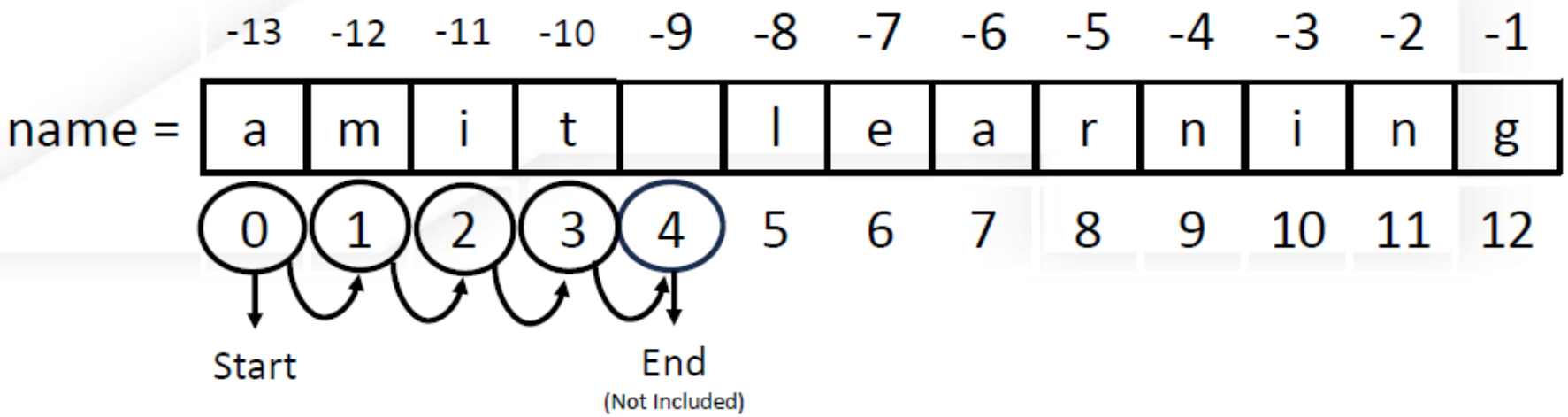
```
i
```



String Slicing

print(variable_name[start index: end index: step])

```
In [22]:
name = "amit learning"
print(name[0:4:1])

amit
```



Functions to Strings

upper function

```
name = "amit learning"
print(name.upper())
```

```
AMIT LEARNING
```

Upper Function

converts all the letters in a string to uppercase, making the entire string capitalized.

lower function

```
name = "AMIT LEARNING"
print(name.lower())
```

```
amit learning
```

lower Function

converts all the letters in a string to lowercase, transforming the entire string to its lowercase form.

replace function

```
name = "amit learning"
print(name.replace("amit","hello"))
```

```
hello learning
```

replace Function

used to replace a specified substring or character in a string with another substring, creating a modified version of the original string.

capitalize function

```
name = "amit learning"
print(name.capitalize())
```

```
Amit learning
```

capitalize Function

The capitalize() function in Python is used to capitalize the first letter of a string, converting the rest of the characters to lowercase if applicable.

Functions to Strings

Title function

```
name = "amit learning"
print(name.title())
```

```
Amit Learning
```

Title Function

used to convert the first character of each word in a string to uppercase,

swapcase function

```
name = "amit LEarning"
print(name.swapcase())
```

```
AMIT leARNING
```

Swapcase Function

used to swap the case of each character in a string, converting uppercase letters to lowercase and vice versa.

count function

```
name = "amit learning amit learning"
print(name.count("amit"))
```

```
2
```

Count Function

used to count the number of occurrences of a specified substring or character within a string

split function

```
name = "amit learning amit learning"
print(name.split())
```

```
['amit', 'learning', 'amit', 'learning']
```

split Function

used to split a string into a list of substrings based on a specified delimiter.

Functions to Strings

join function

```
words = ["This", "is", "a", "sentence."]
sentence = ' '.join(words)
print(sentence)
```

```
This is a sentence.
```

Join Function

used to concatenate a sequence of strings with a specified separator, creating a single string.

startwith function

```
name = "amit learning amit learning"
print(name.startswith("Hello"))
```

```
False
```

startwith Function

used to check if a string starts with a specified prefix.

endwith function

```
name = "amit learning amit learning"
print(name.endswith("learning"))
```

```
True
```

endwith Function

used to check if a string ends with a specified suffix.

Escape Characters

Escape characters in Python are special codes (starting with \) used to represent hard-to-type characters or control the formatting of strings

new line

```
print("Hello\nWorld")
```

```
Hello
World
```

tab

```
print("Hello\tWorld")
```

```
Hello	World
```

double qoute

```
print("This is a double quote: \"hello\" ")
```

```
This is a double quote: "hello"
```

backspace

```
print("Hello\b")
```

```
Hell
```

Try To Solve

What is the output of the following code?

```
> my_string = "0123456789"
> print(my_string[-2: -6: -2])
```

- 5432
- 8765
- 532
- 86

Try To Solve

What is the output of the following code?

```
> my_string = "0123456789"
> print(my_string[-2: -6: -2])
```

- 5432
- 8765
- 532
- 86

Assignment

Encoded Message:

`###!!@mocleW EPGTQ!!!6789`

Steps to Decode:

1. Extract the core part of the message: "`mocleW EPGTQ`".
2. Reverse the first word: "`mocleW`" becomes "`Welcome`".
3. Replace shifted vowels in the second word:
    - "`EPGTQ`": No vowels to change.
4. Final decoded message: "`Welcome PGTQ`".

Suppose you have this email address “Amit_ml@gmail.edu”

- Input Validation: Check if the input string contains exactly one "@" symbol and at least one "." after the "@" symbol. If it's not a valid email, return "Invalid email".

- Extract Username: Extract and return the part of the email before the "@" symbol.

- Extract Domain: Extract and return the domain (the part between "@" and the last ".").

- Check for Domain Ending: Check if the email ends with ".com". If it does, return "Commercial Domain". If it ends with ".edu", return "Educational Domain". Otherwise, return "Other Domain".

Thank you