Introduction to Programming(Python)

Content

Part - 1

- Programming Logic
- IDEs for Python
- Programming: 0 to Hero
- To Python
  - Variable: declaration and naming conventions
  - Strings
  - Data Types
  - Operators
- Commenting and Indentation
- Conditional statements
- Loops

Part - 2

- Data Structures
- Getting user input
- Some modules - Random, Math
- Functions (4 types)
- Best sites for python
- Simple programs
  - Roots of a quadratic equation
  - Finding Factorial
  - Reversing a number/string

Part 1

Programming Logic

Programming languages are a tool to **implement our logic to solve a problem,** python is just one of them

Logic is the way by which we are going to solve the problem, it is usually called as **algorithm** in technical terms

Reversing a number

Input → 427   Output → 724

```
num = 427
rev = 0

while num > 0

        mod = num % 10
        rev = 0 * 10 + mod
        num = num / 10

print(rev)
```

Python

Why Python?

Easy to learn, Can be used for wide range of applications, Large standard libraries & huge community support.

What is Python?

Python is a high-level, dynamically typed, interpreted programming language with a minimal syntax that reads like English.

What can be made using Python?

Python can make literally anything in IT, from printing a word to AI

Need to know more? DIY 😇

Getting Started with Python

1. Download python from python.org
2. Check **Add to path**, and Install python on your device
3. Install JupyterLab & Jupyter Notebook using PIP
    a. Open cmd on your computer, and
    b. Type ***pip install notebook***
4. Open Jupyter Notebook by typing ***jupyter notebook*** in cmd

Integrated Development Environment (IDE) for Python

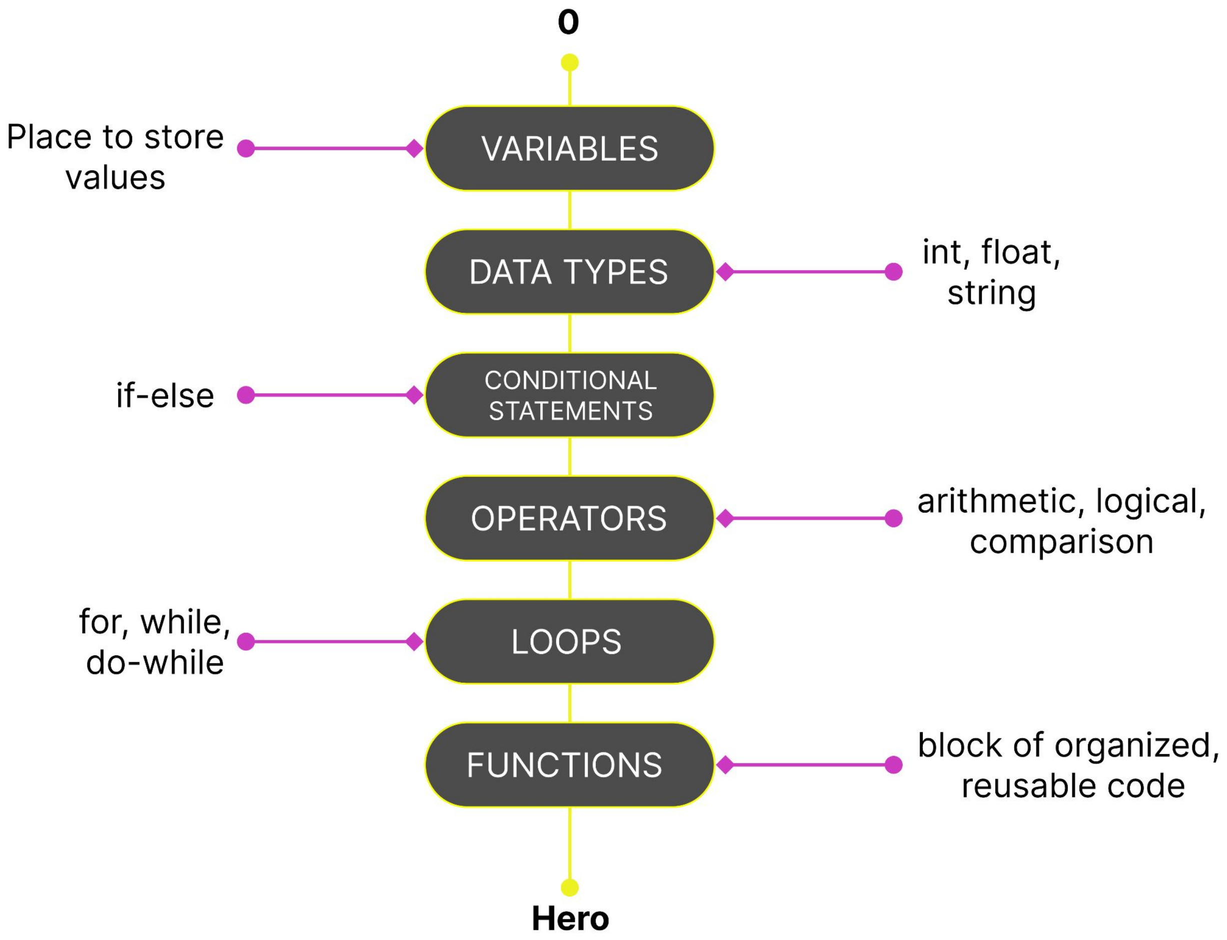
Jupyter Notebook

VS Code

Python IDLE

PyCharm

Programming: 0 to Hero



Python Variables

Python is a **Dynamically Typed language**

Variable declaration & assignment:

num = 10 → it is an integer

name = “John” → it is a string

Variables are case sensitive

Variables cannot be started with numbers, they cannot contain spaces and special characters.

x, z, FirstName, last_name, age - these are some valid variables in python

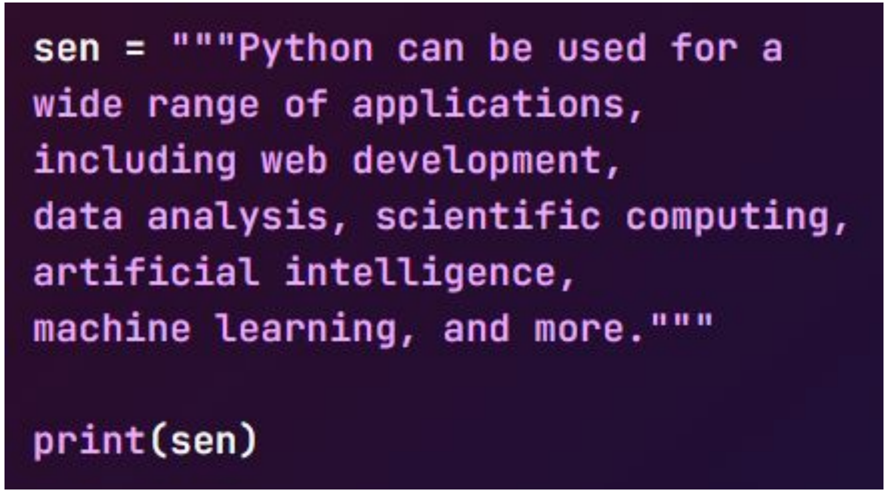
Find out the cases of these variables.

Python Strings

Strings in python are surrounded by either single quotation marks, or double quotation marks.

‘python' is the same as "python"

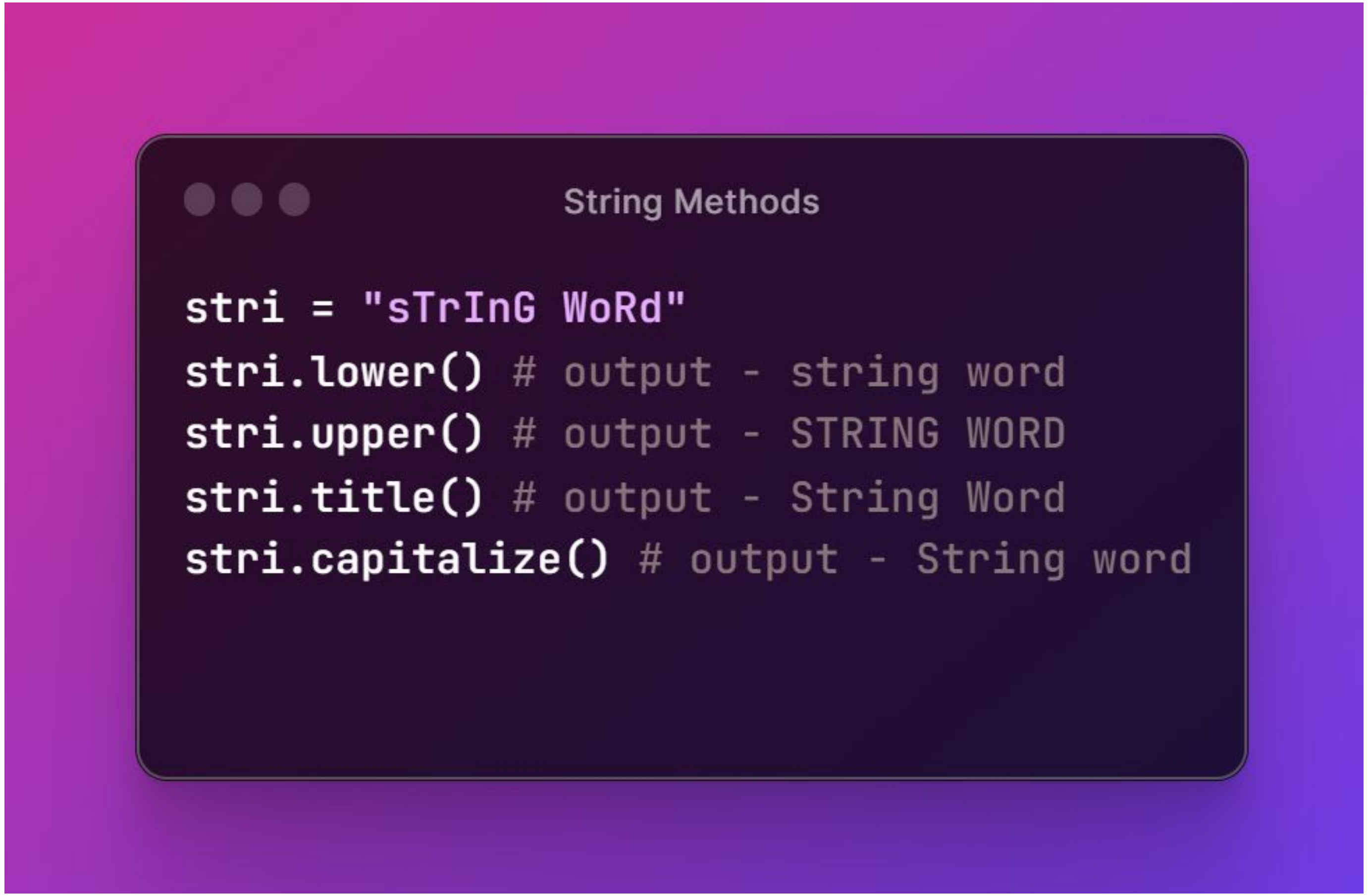
word = “programming” → example for string variable



```
sen = """Python can be used for a
wide range of applications,
including web development,
data analysis, scientific computing,
artificial intelligence,
machine learning, and more."""

print(sen)
```

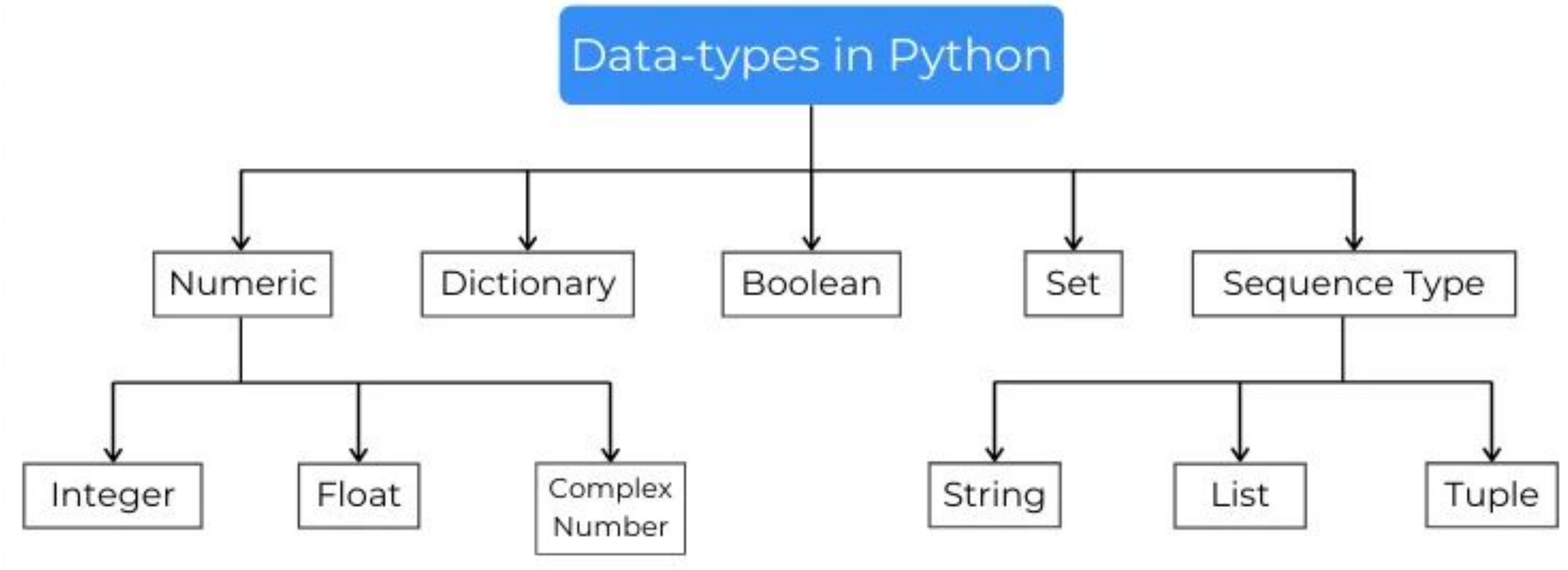
A Story can be written in python using triple quotes, which is read as string

Python String Methods



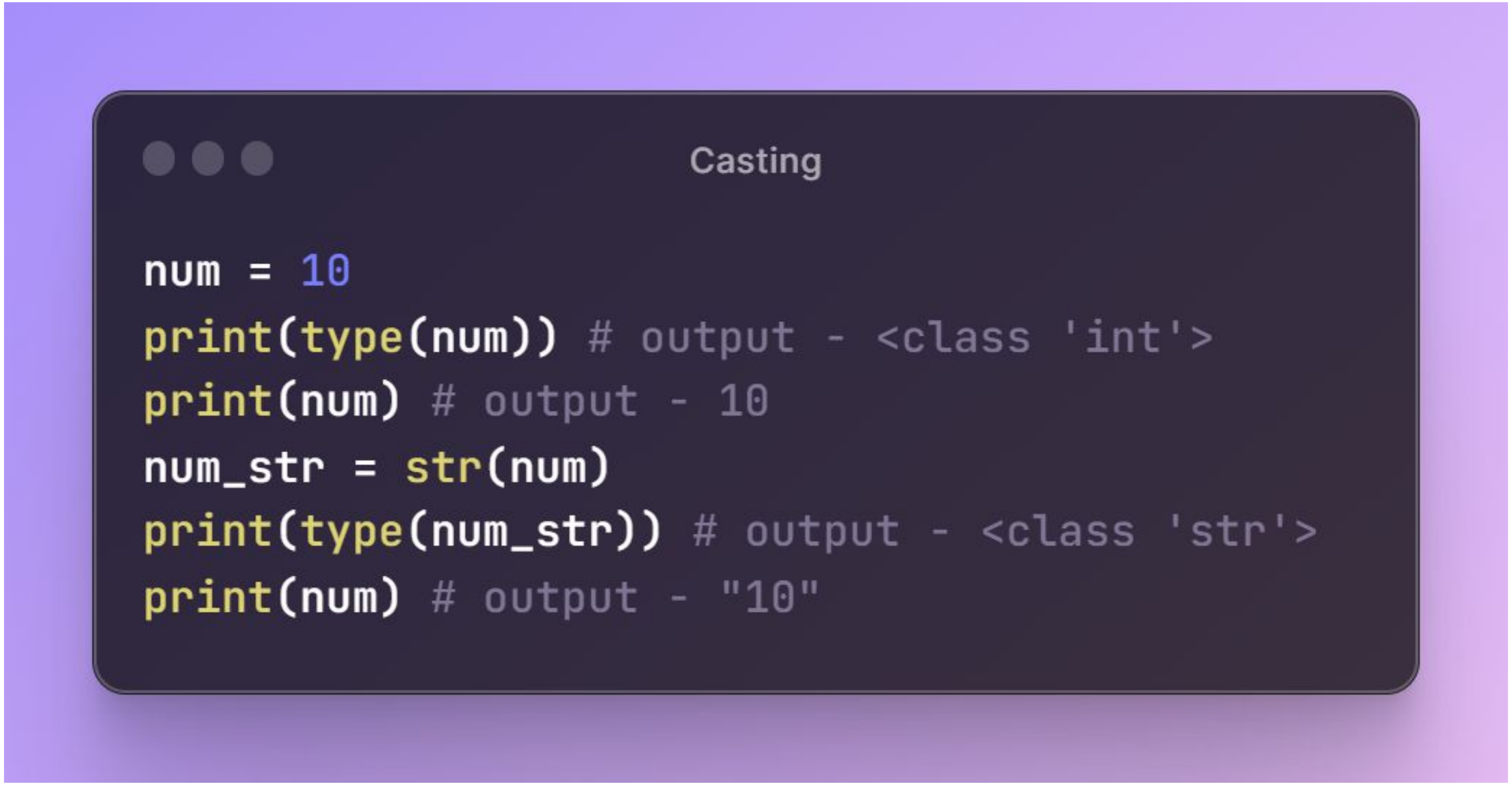
```
stri = "sTrInG WoRd"
stri.lower() # output - string word
stri.upper() # output - STRING WORD
stri.title() # output - String Word
stri.capitalize() # output - String word
```

Python Data Types



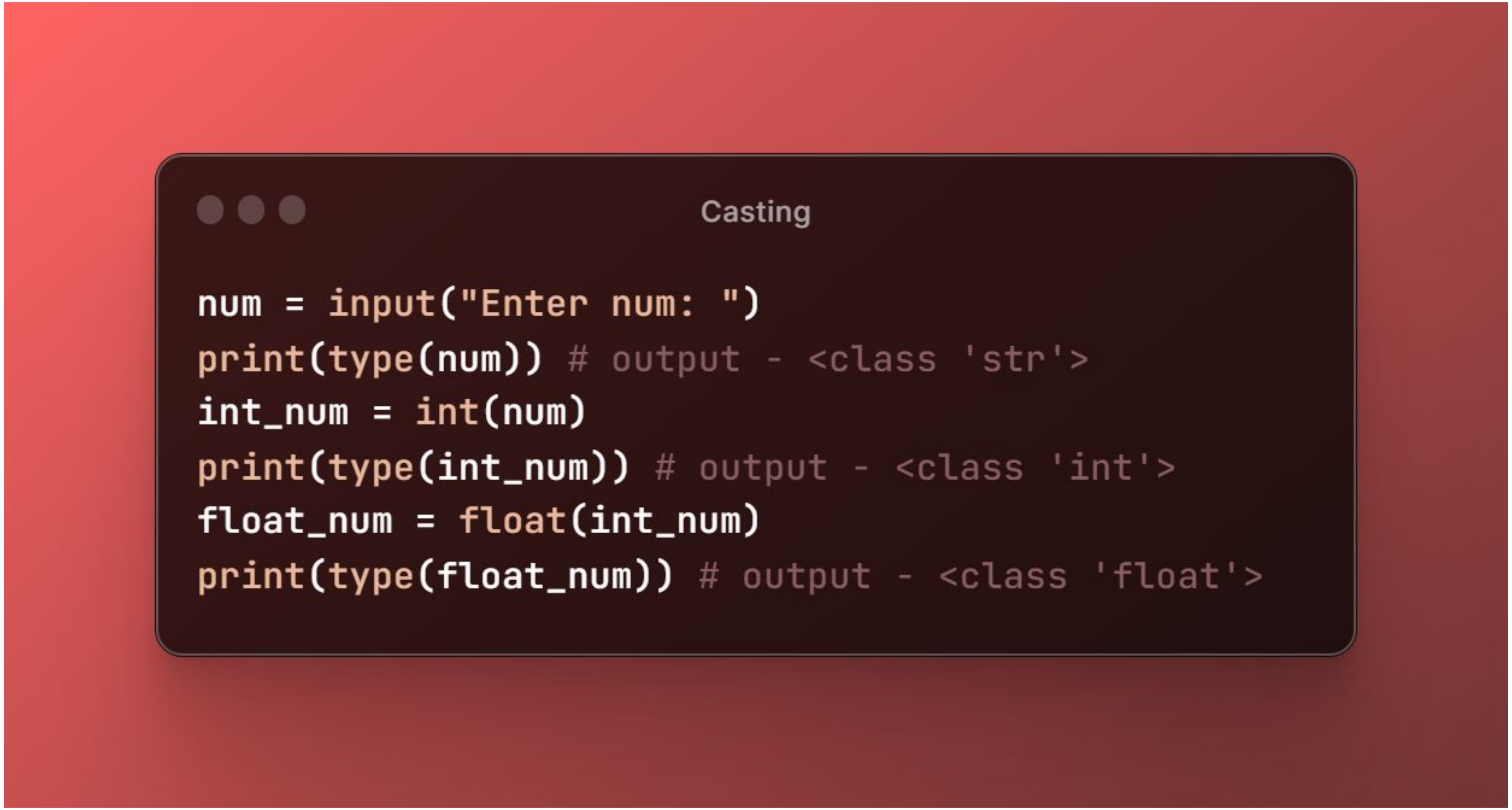
Use **type()** to see the data type of the variable
Casting → Changing a variable to another data type

Python Casting



```python
num = 10
print(type(num)) # output - <class 'int'>
print(num) # output - 10
num_str = str(num)
print(type(num_str)) # output - <class 'str'>
print(num) # output - "10"
```

Python Casting



```python
num = input("Enter num: ")
print(type(num)) # output - <class 'str'>
int_num = int(num)
print(type(int_num)) # output - <class 'int'>
float_num = float(int_num)
print(type(float_num)) # output - <class 'float'>
```

Types Of Operators In Python

Type	Operators
Arithmetic Operators	+,-,/,%,//,**
Comparison Operators	>,<,==,!=,>=,<=
Logical Operators	and,or,not
Bitwise Operators	&,|,~,^,>>,<<
Assignment Operators	=,+=,-=,/=
Identity Operators	is, is not
Membership Operators	in, not in

‘Equal to’ and ‘Double Equal to’

Equal to is used for assigning a value to a variable

a = 10

We have assigned the value ‘10’ to ‘a’

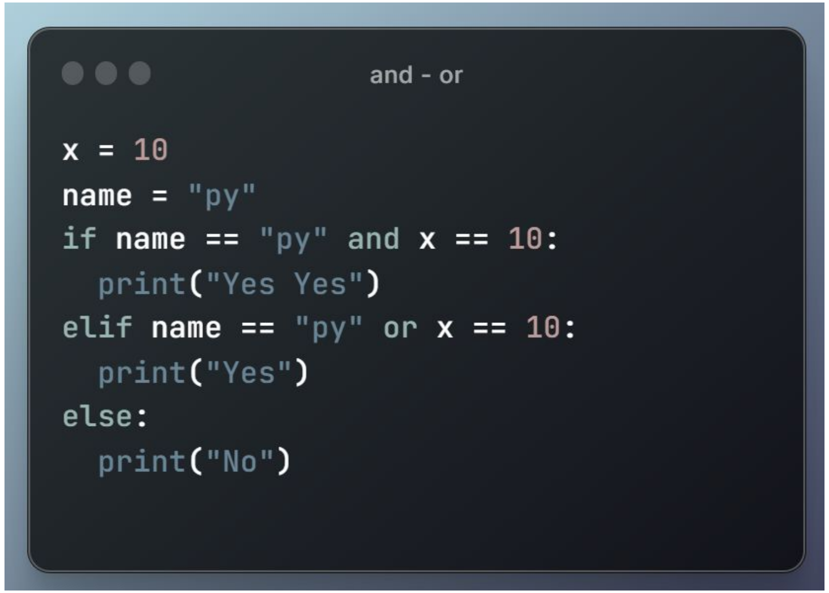
Double Equal to is used for checking value in a variable

a == 10

Checking if value in ‘a’ is 10

and AND or

In other programming languages we use symbols like &&, || for logical operations, but in python we use the words **and, or** for logical operations

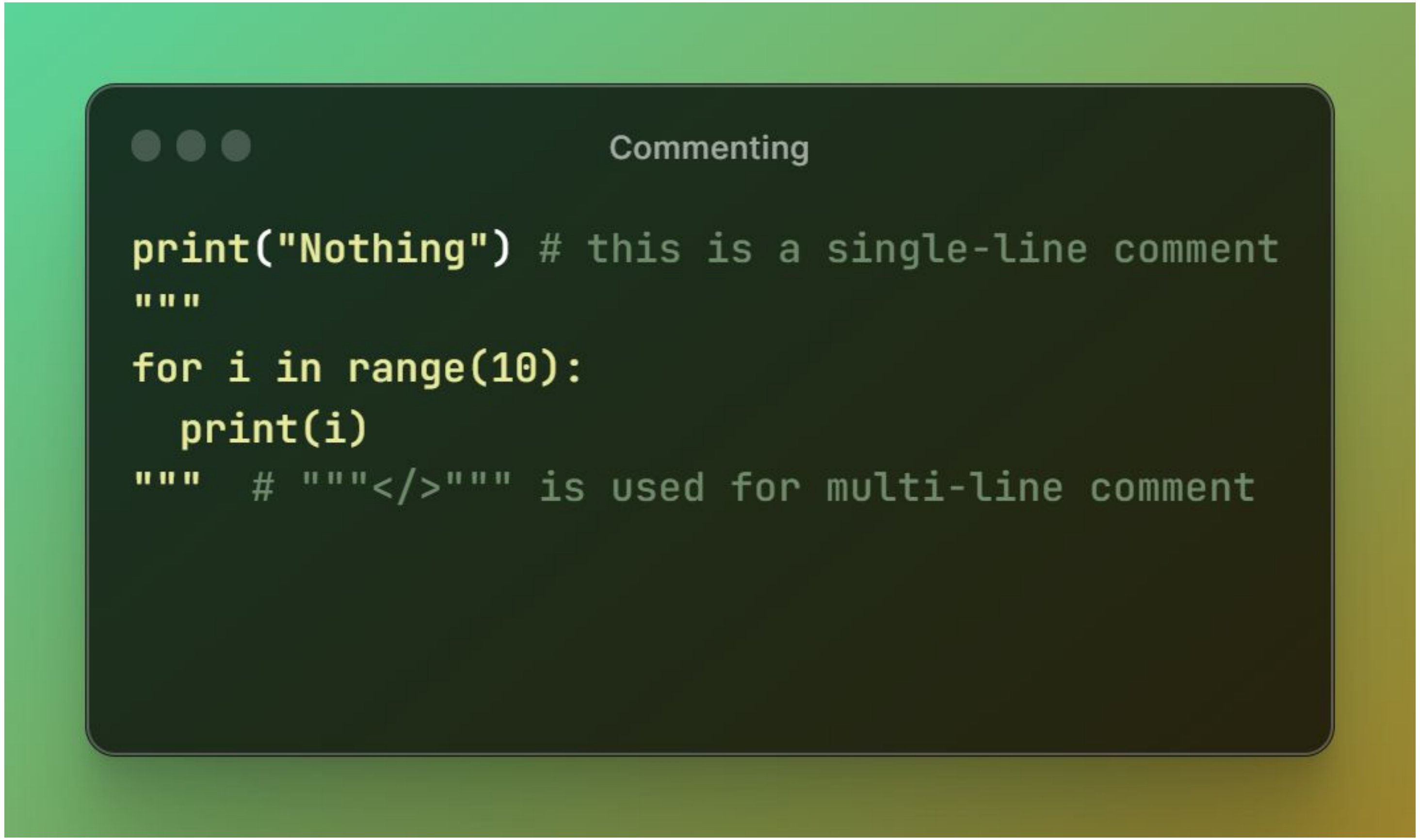


```python
x = 10
name = "py"
if name == "py" and x == 10:
  print("Yes Yes")
elif name == "py" or x == 10:
  print("Yes")
else:
  print("No")
```

i = i + 1, not i++ or ++i

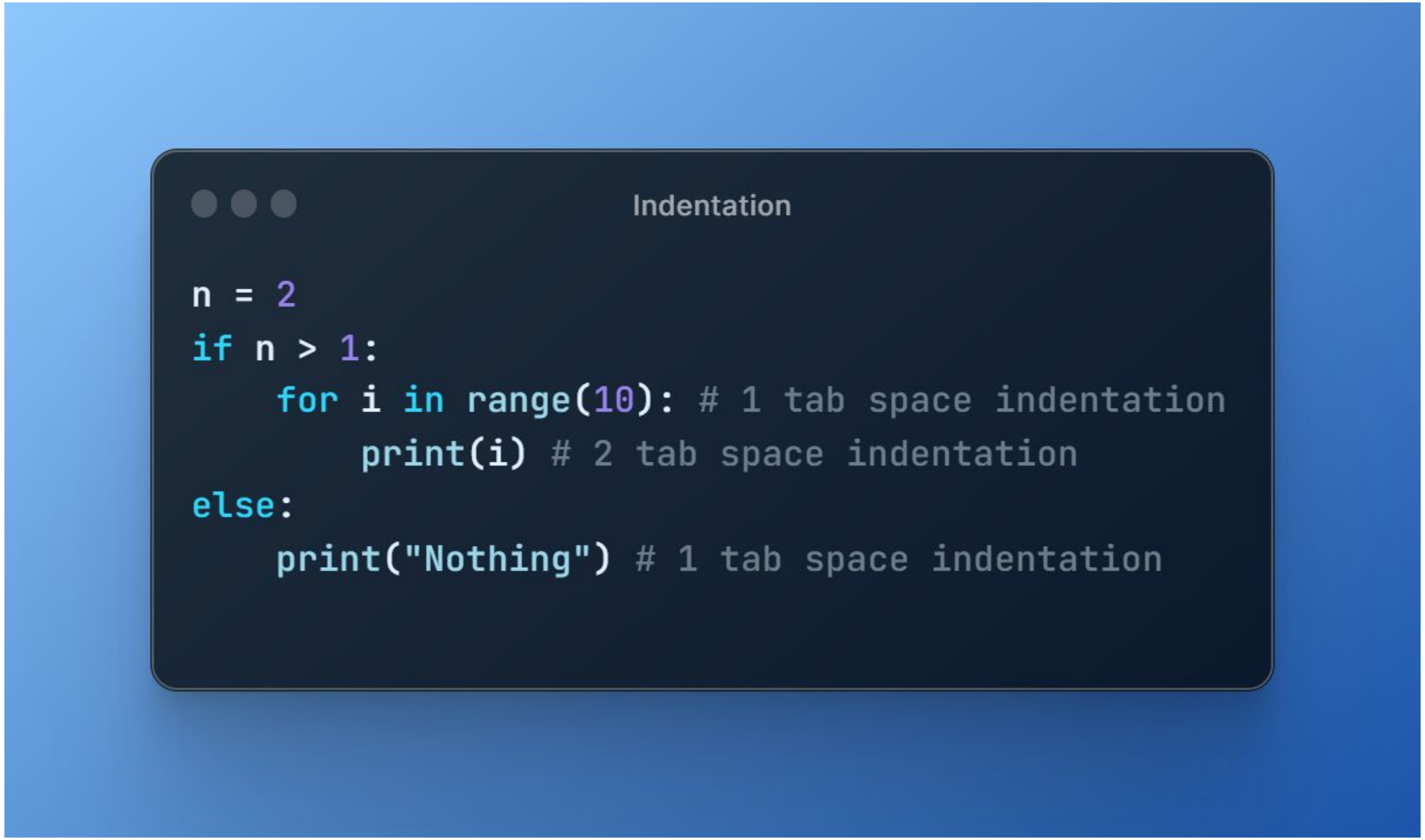
In python we increment the value of a variable using i = i + 1

Python Commenting



```
print("Nothing") # this is a single-line comment
"""
for i in range(10):
  print(i)
"""  # """</>""" is used for multi-line comment
```

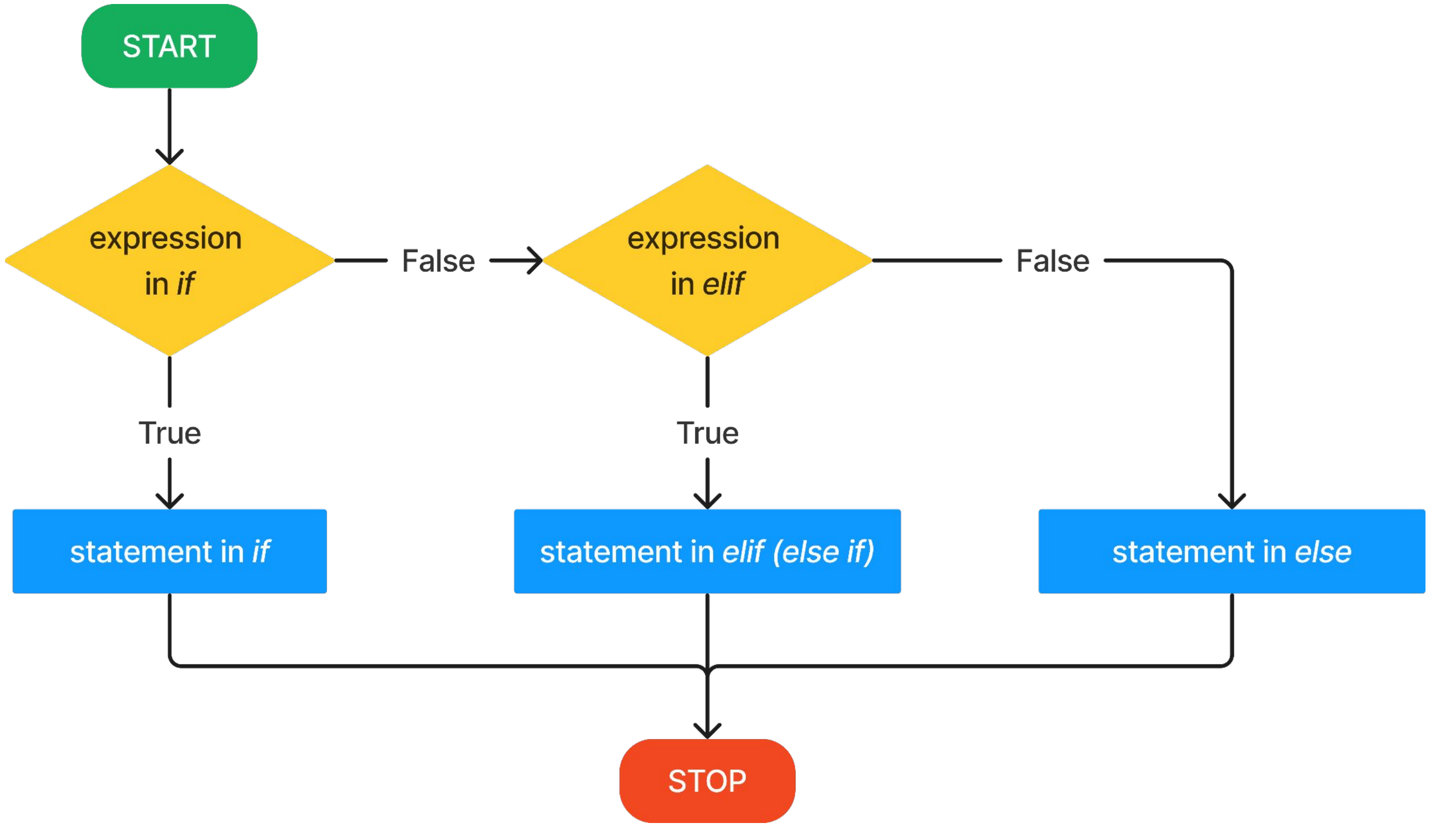
Python Indentation



```python
n = 2
if n > 1:
    for i in range(10): # 1 tab space indentation
        print(i) # 2 tab space indentation
else:
    print("Nothing") # 1 tab space indentation
```

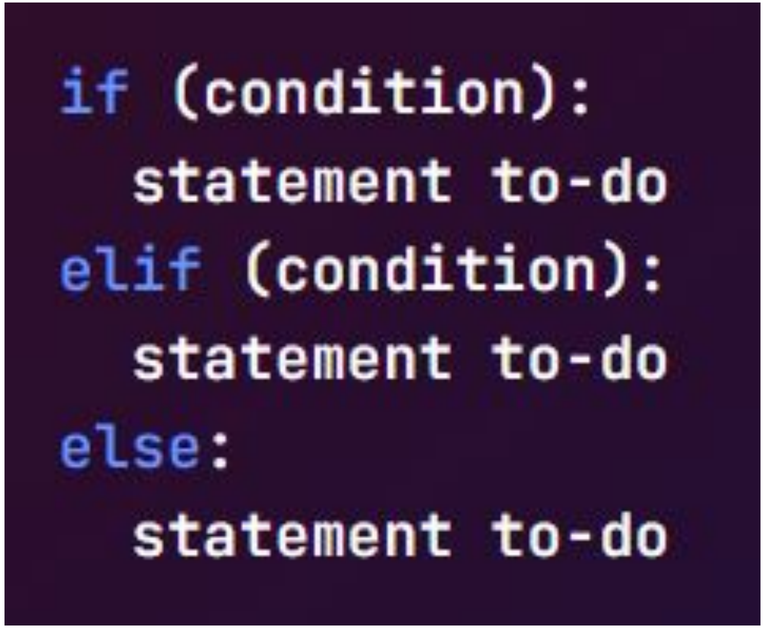
Python Conditional Statements

There is if-else, but **no switch case**



Python if-else

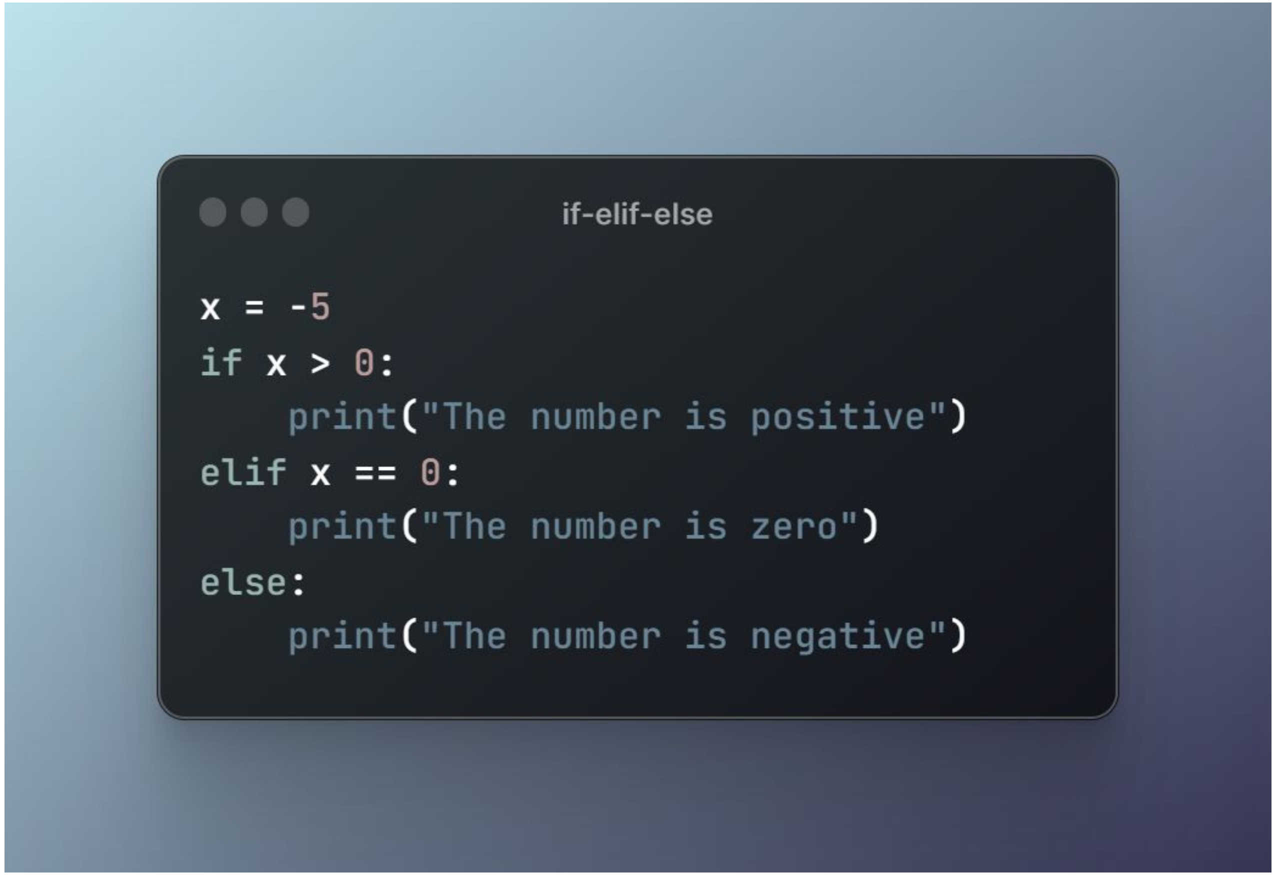
If-elif-else syntax in python



```
if (condition):
  statement to-do
elif (condition):
  statement to-do
else:
  statement to-do
```

In-addition, we could use
break
continue, and
pass
in python if-elif-else statements

Python if-else



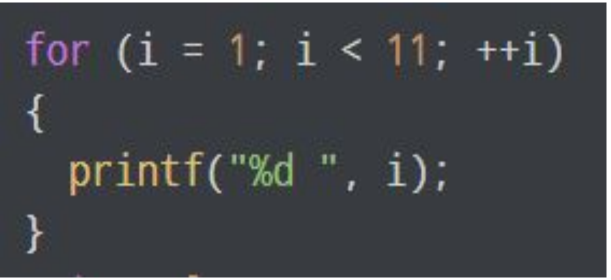
```python
x = -5
if x > 0:
    print("The number is positive")
elif x == 0:
    print("The number is zero")
else:
    print("The number is negative")
```

Programming: Loops

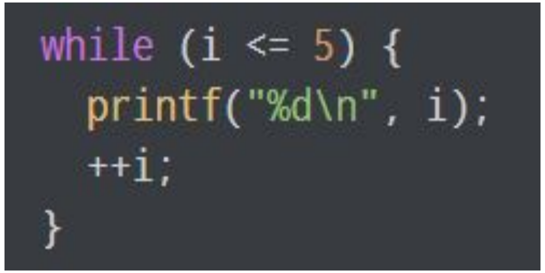
Loops are used to **repeat a statement n times**

Main loops in programming are **For loop, While loop and Do-While loop**

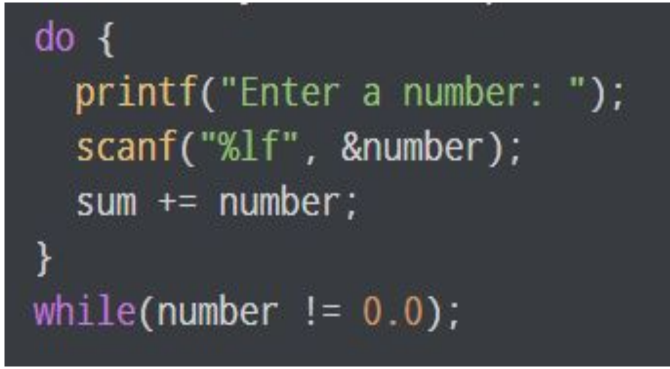
Python doesn't have do-while loop, as while loop with if condition does the same work



```
for (i = 1; i < 11; ++i)
{
  printf("%d ", i);
}
```



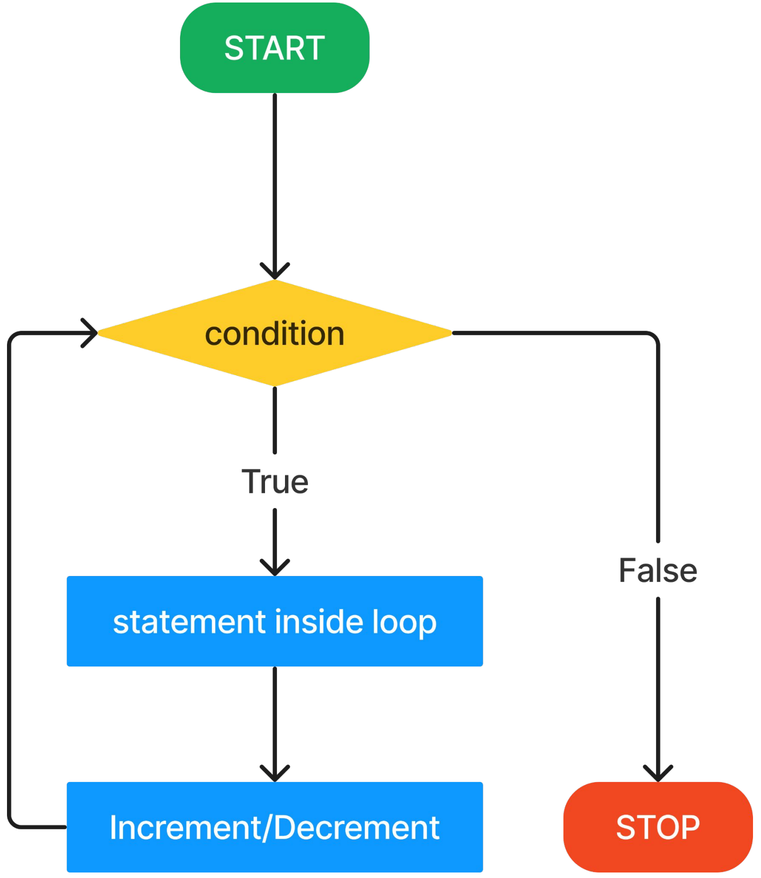
```
while (i <= 5) {
  printf("%d\n", i);
  ++i;
}
```



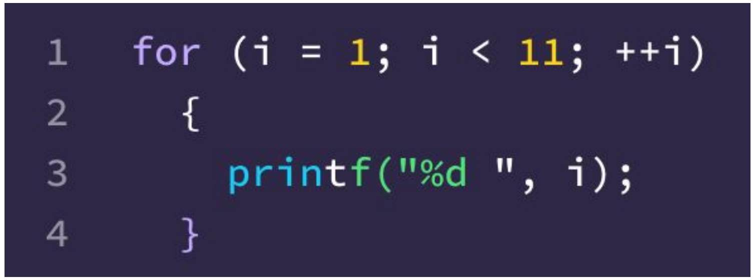
```
do {
  printf("Enter a number: ");
  scanf("%lf", &number);
  sum += number;
}
while(number != 0.0);
```

These are loops implemented in C

Python For Loop

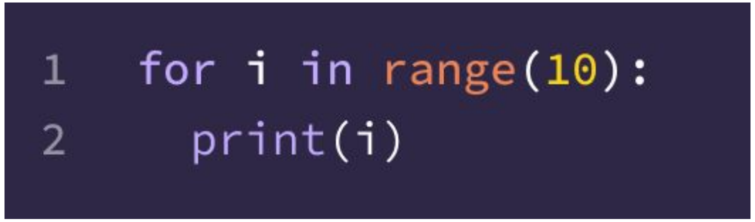


For loop in C



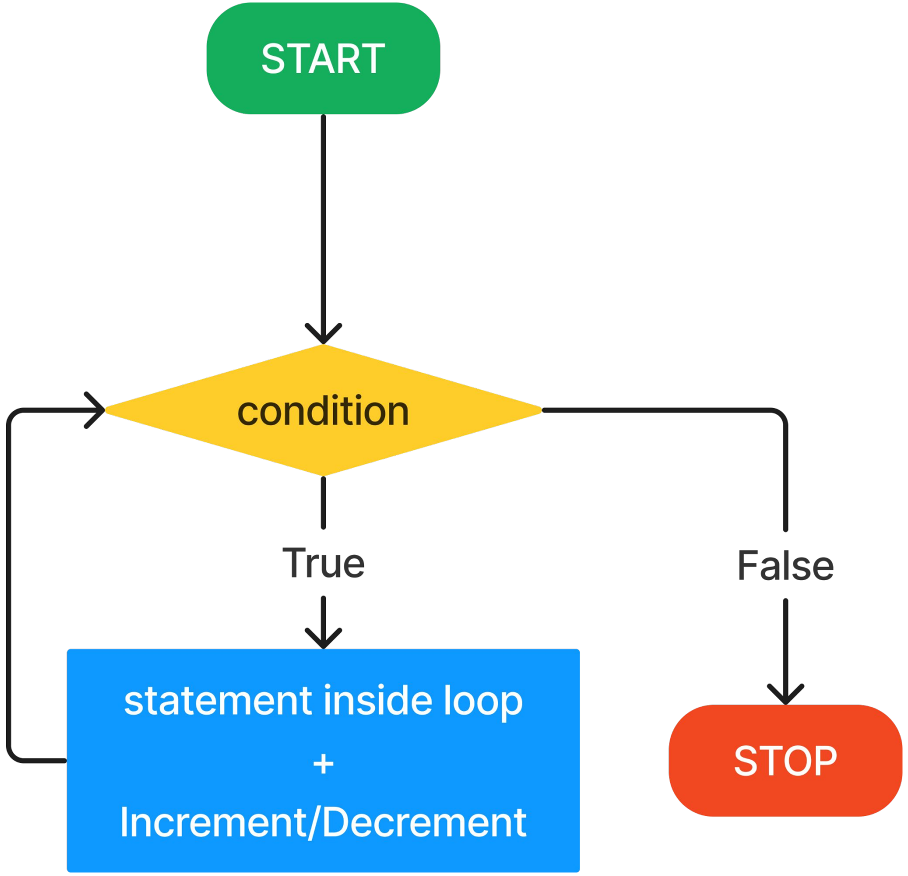
```c
for (i = 1; i < 11; ++i)
  {
    printf("%d ", i);
  }
```

For loop in Python

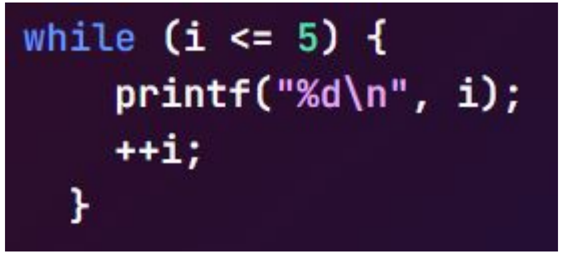


```python
for i in range(10):
  print(i)
```

Python While Loop

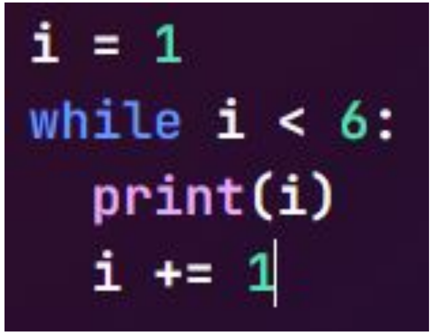


While loop in C



```c
while (i <= 5) {
    printf("%d\n", i);
    ++i;
  }
```

While loop in Python



```python
i = 1
while i < 6:
  print(i)
  i += 1
```

End of Part 1

Into Python Programming

Largest of 3 numbers

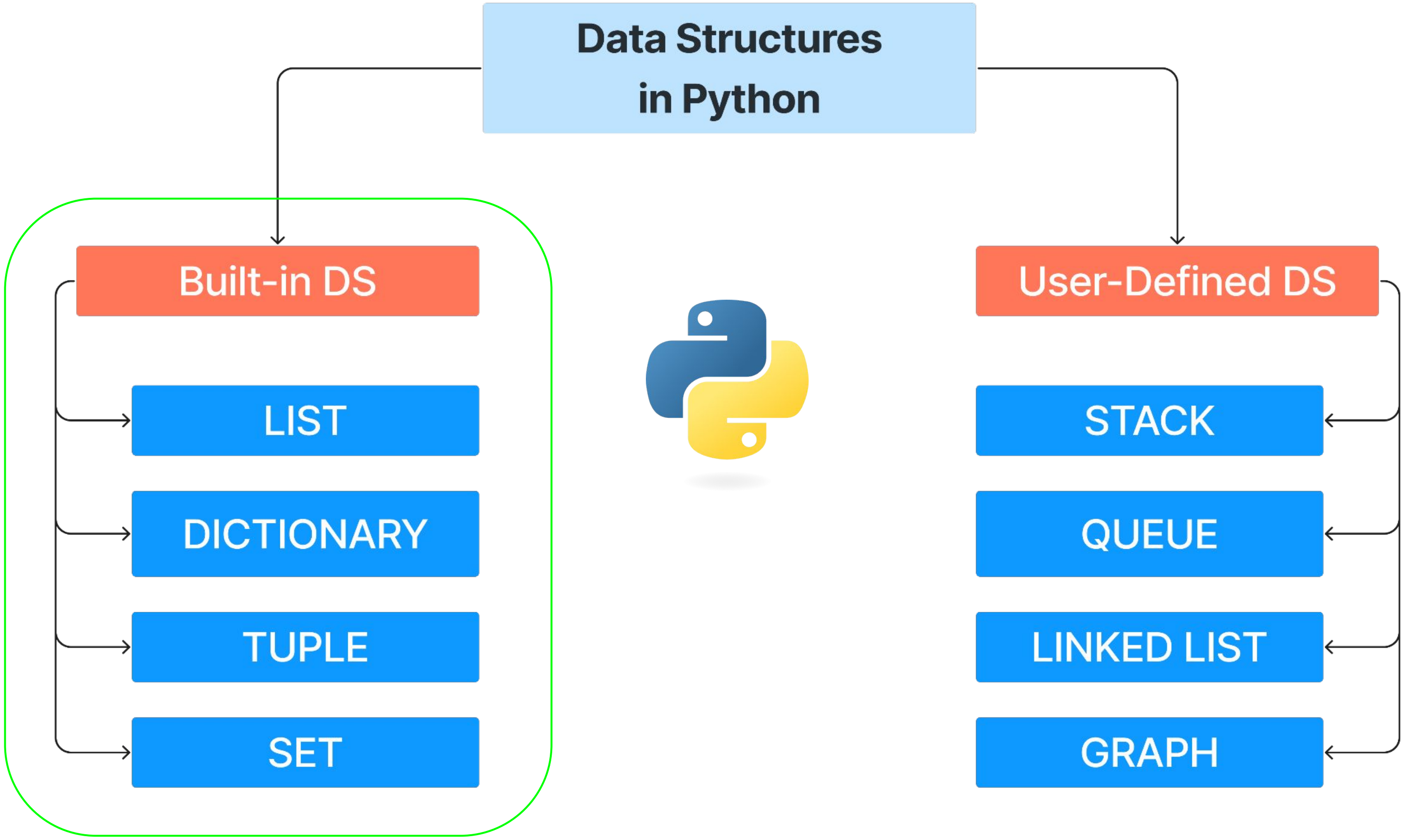
Live coding, me(with the help of students) or the students itself

Into Python Programming

Multiplication table of 5

Live coding, me(with the help of students) or the students itself

Part 2

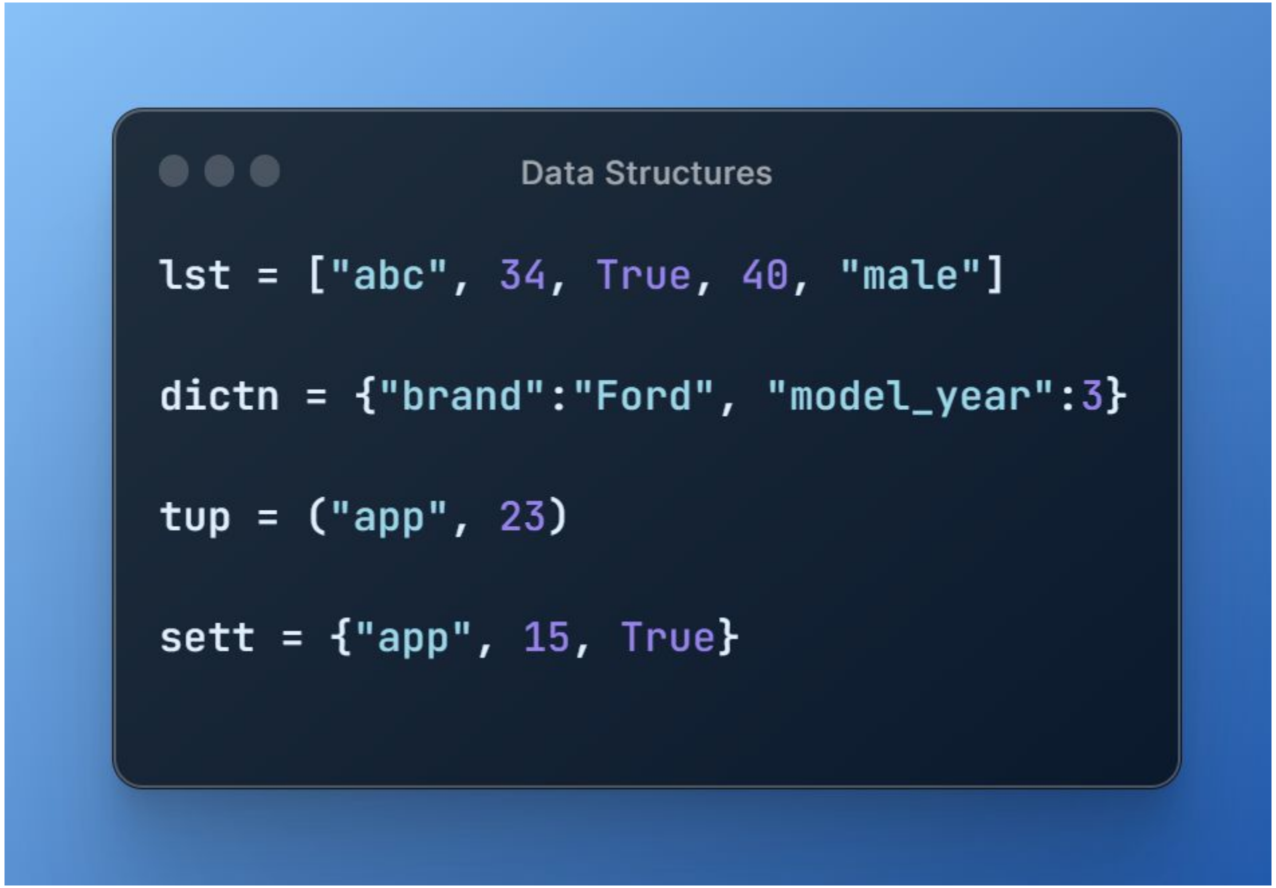


INDEXING IN PYTHON STARTS FROM **0**

Python Data Structures Properties

LIST				
[item1, item n]	ordered	changeable	allow duplicate values	indexed
DICTIONARY				
{ key : value }	ordered	changeable	**no** duplicate values	referred using key
TUPLE				
(item1, item n)	ordered	**un**changeable	allow duplicate values	indexed
SET				
{ item1, item n }	**un**ordered	**un**changeable	**no** duplicate values	**un**indexed

Python Data Structures



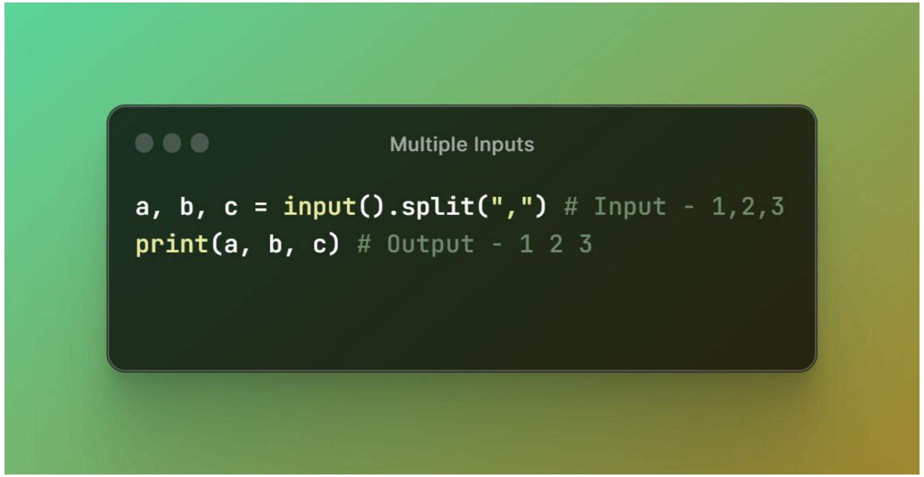
```
lst = ["abc", 34, True, 40, "male"]

dictn = {"brand":"Ford", "model_year":3}

tup = ("app", 23)

sett = {"app", 15, True}
```

User Input

You can take inputs from user using **input()**
You can write messages in input()
input() takes in input as string

Multiple Input in a single line

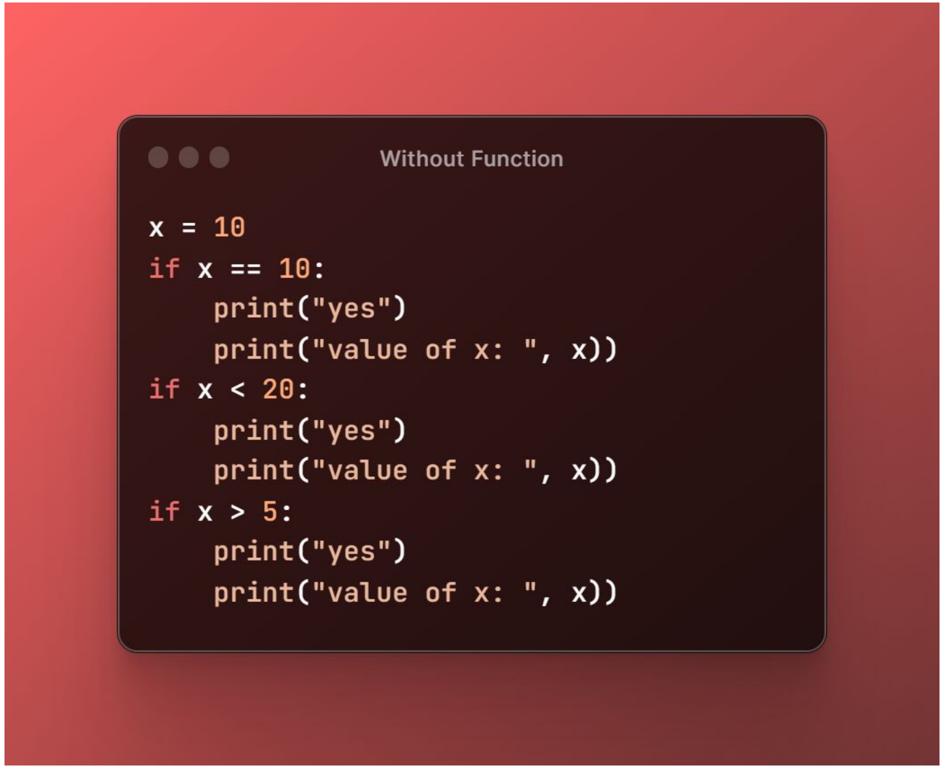
For this, you can use **split()**



```python
a, b, c = input().split(",") # Input - 1,2,3
print(a, b, c) # Output - 1 2 3
```

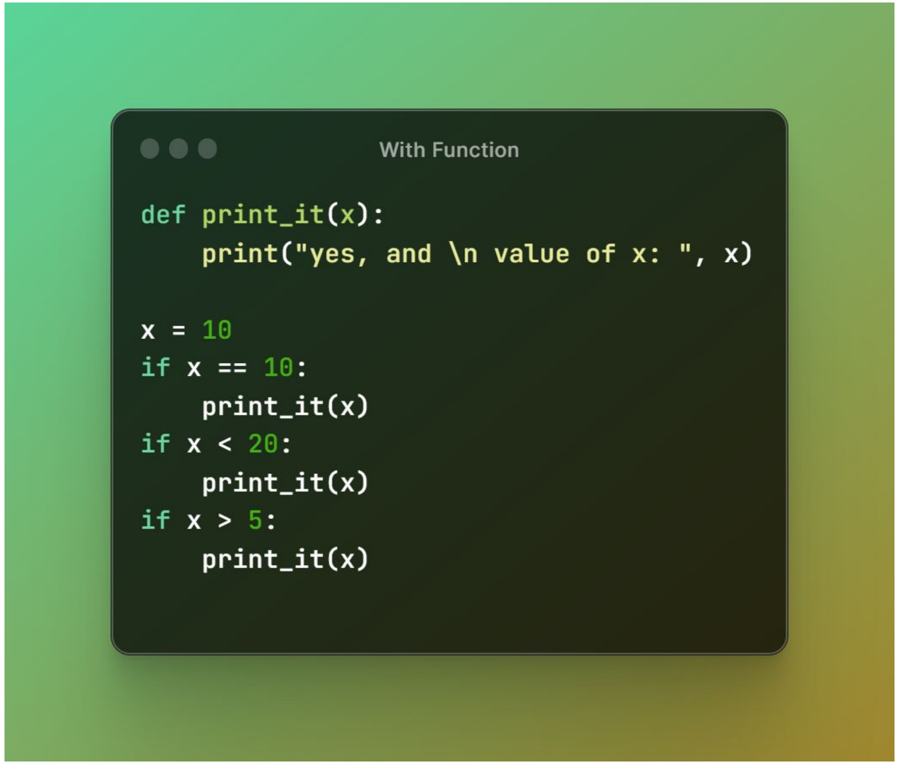
Python Functions

Function is a block of reusable code that performs a specific task



Without Function

```
x = 10
if x == 10:
    print("yes")
    print("value of x: ", x))
if x < 20:
    print("yes")
    print("value of x: ", x))
if x > 5:
    print("yes")
    print("value of x: ", x))
```



With Function

```
def print_it(x):
    print("yes, and \n value of x: ", x)

x = 10
if x == 10:
    print_it(x)
if x < 20:
    print_it(x)
if x > 5:
    print_it(x)
```

Which looks better, and easy to read?

Python Functions

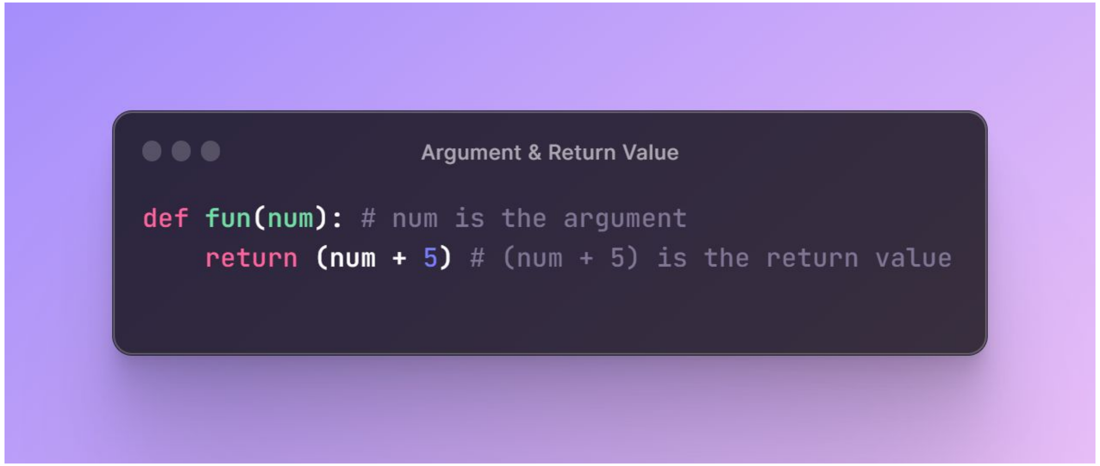
Two words: Argument & Return value

Argument is a value that is passed to a function when it is called

Return value is the value that a function sends back to the calling code after finishing all operations

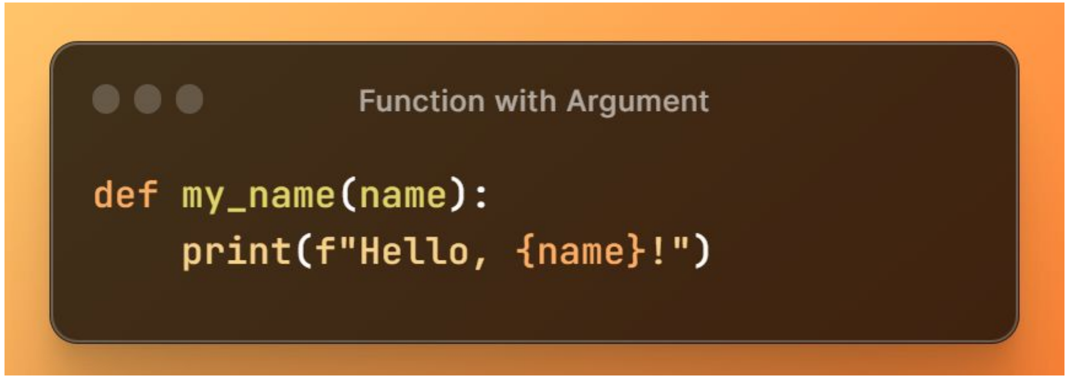
Types of functions:

Function with argument
Function without argument
Function with return value
Function without return value



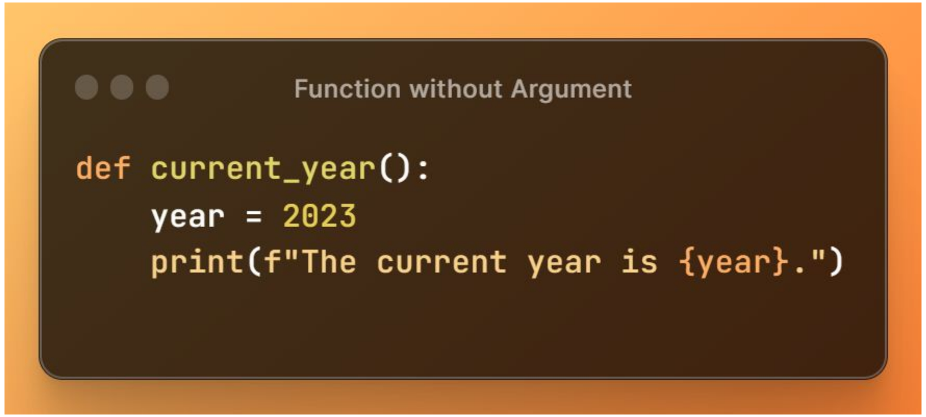
```python
def fun(num): # num is the argument
    return (num + 5) # (num + 5) is the return value
```

Function with Argument



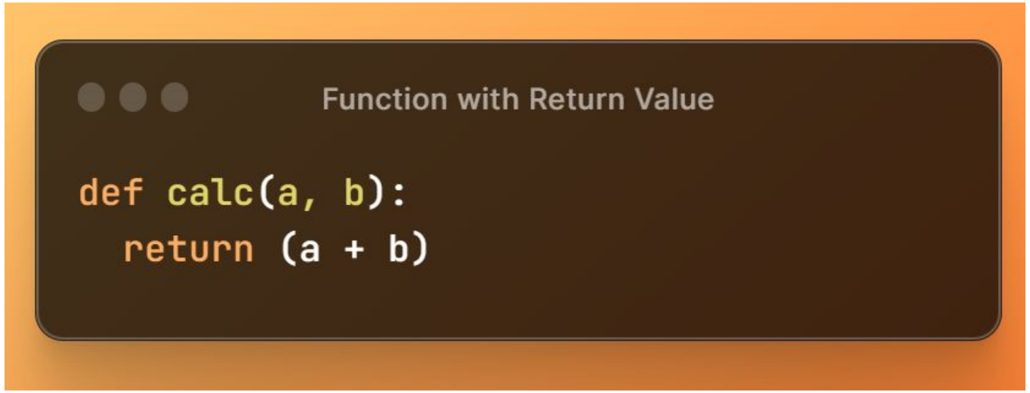
```python
def my_name(name):
    print(f"Hello, {name}!")
```

Function without Argument



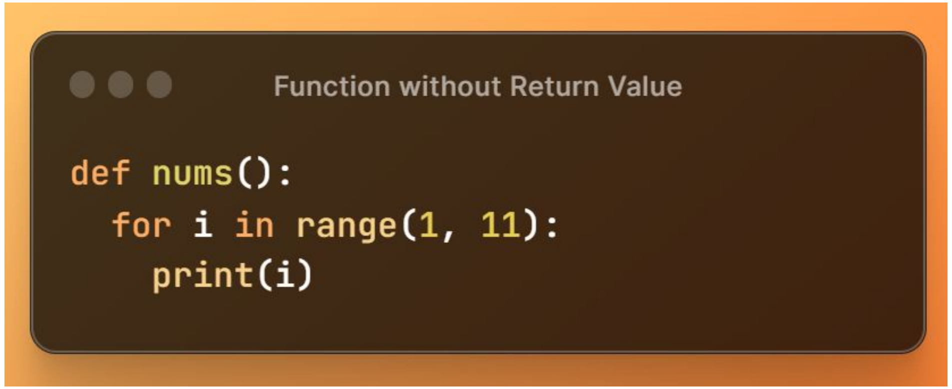
```python
def current_year():
    year = 2023
    print(f"The current year is {year}.")
```

Function with Return Value


Function with Return Value

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

Function without Return Value


Function without Return Value

```
def nums():
  for i in range(1, 11):
    print(i)
```

Best sites for Python

w3schools

GitHub

GeeksforGeeks

Google

Official Documentations

YouTube University

stackoverflow

Medium

Quora

javaTpoint

tutorialspoint
SIMPLY EASY LEARNING

Programiz

DEV

You Must Do

Read Official Documentations (Official Websites)

Get feedbacks (Communities, Friends)

Show you results/failures (LinkedIn)

See other people's code (GitHub)

Study coding standards (GitHub)

Collaborate (GitHub)

End of Part 2

Python Programming Exercise

Quadratic Equation

Live coding, me(with the help of students) or the students itself

Python Programming Exercise

Factorial of a number

Live coding, me(with the help of students) or the students itself