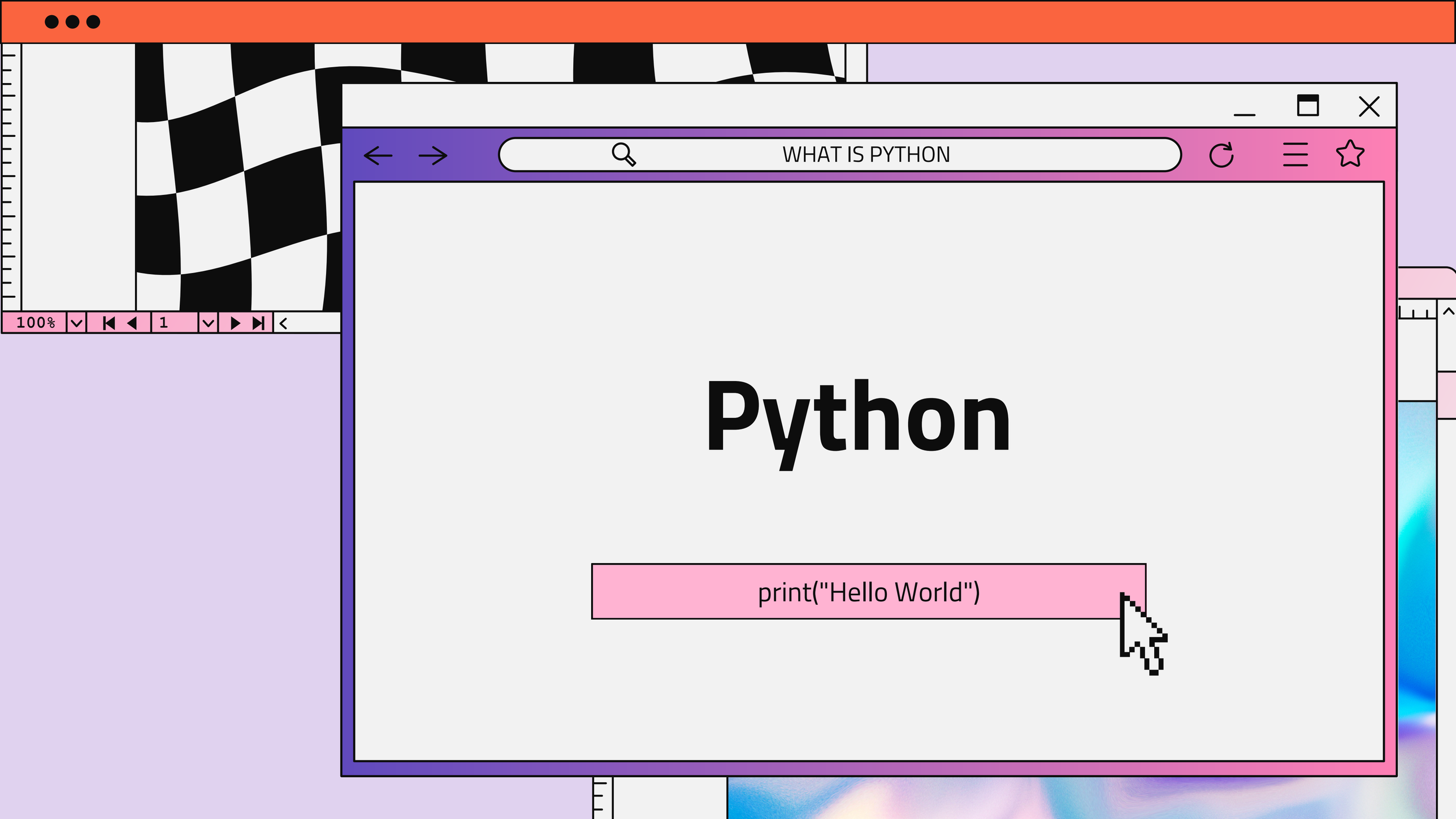
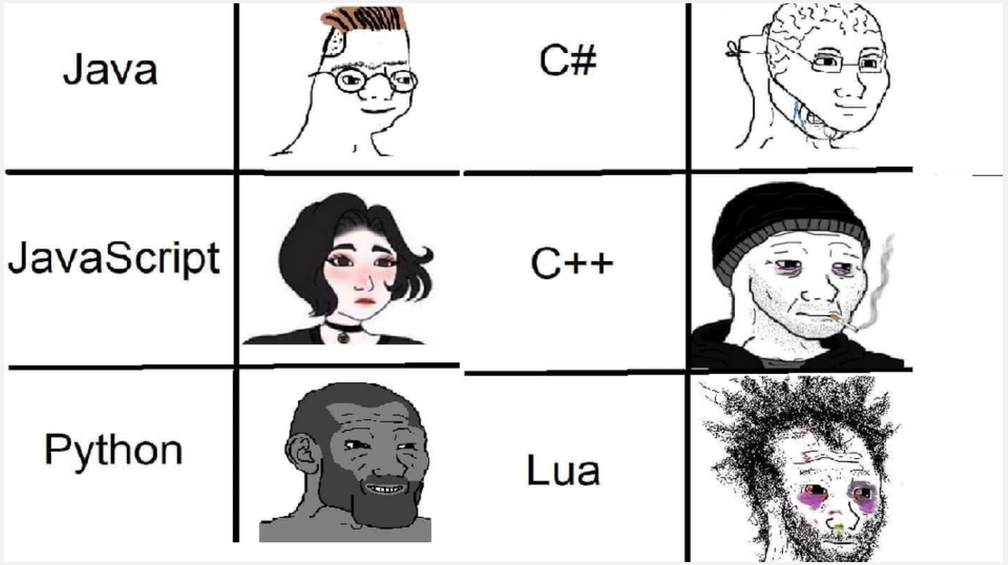

WHAT IS PYTHON
Python
print("Hello World")
100%
1

Overview

Tip: less is more

How: It is a high-level programming language that has user-friendly syntax. It is known to have fewer lines of code in an easy and simple form



Untitled - TextEdit

File Edit View Help

New Tab

Introduction To Python

2 - Macrostuff Board

t Format View Help

12

Guido van Rossum

- Python is a widely used general-purpose, high level programming language.
- Created by Guido van Rossum in 1991 and further develope by the Python Software Foundation.
- Emphasizes code readability.
- Syntax allows programmers to express concepts in fewer lines of code.
- work quickly and integrate systems more efficiently
-

Language features

- Interpreted
- Platform Independent
- Free and Open Source
- High-level Language
- Simple
- Embeddable
- Robust
- Rich Library Support

Data type

data types are actually classes and variables are instance (object) of these classes.

Numberic

- Integers
- Float
- Complex Number

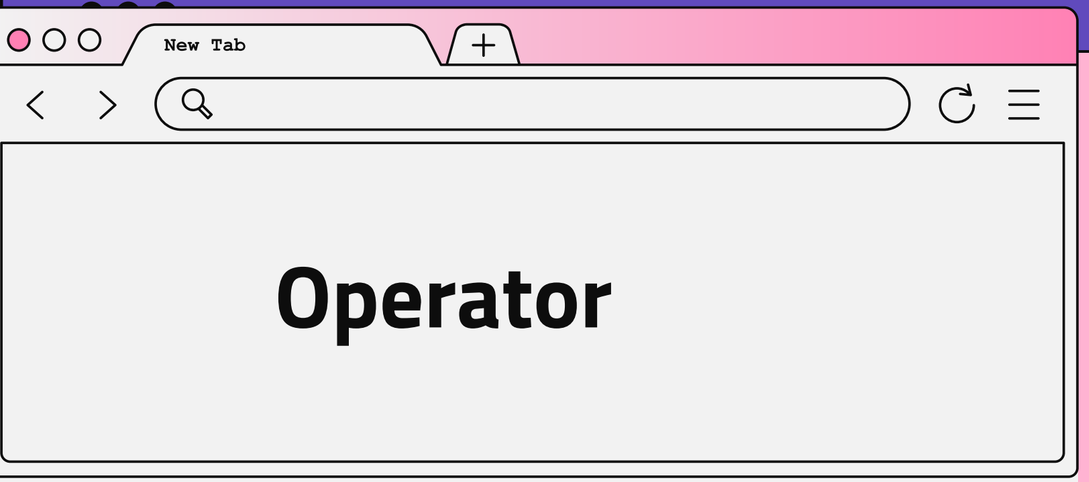
Sequence Type

- String
- List
- Tuple

Boolean

Set

Dictionary



Operator

The Equality operator (==) is a comparison operator in Python that compare values of both the operands and checks for value equality. Whereas the 'is' operator is the identity operator that checks whether both the operands refer to the same object or not (present in the same memory location)

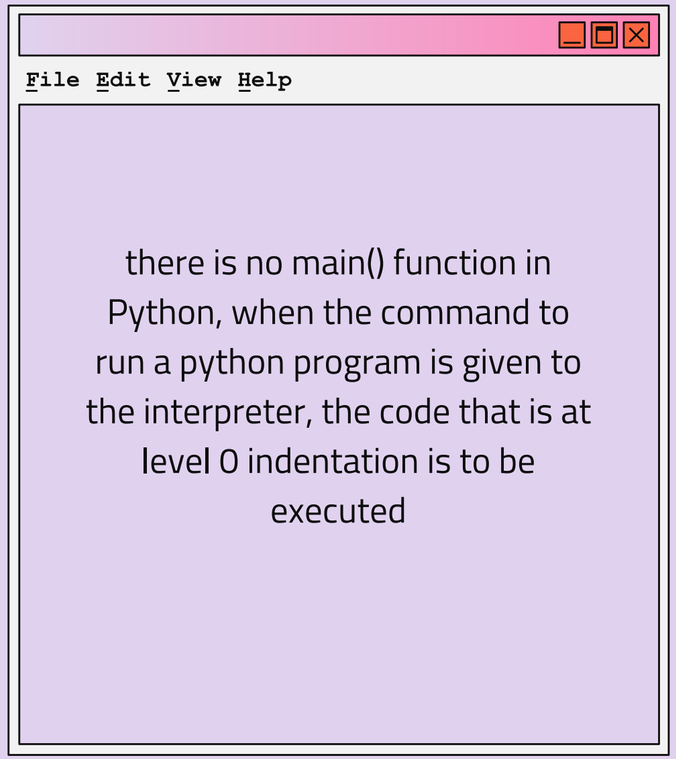
Operator	Description	Syntax
>	Greater than: True if the left operand is greater than the right	x > y
<	Less than: True if the left operand is less than the right	x < y
==	Equal to: True if both operands are equal	x == y
!=	Not equal to - True if operands are not equal	x != y
>=	Greater than or equal to True if the left operand is greater than or equal to the right	x >= y
<=	Less than or equal to True if the left operand is less than or equal to the right	x <= y
is	x is the same as y	x is y
is not	x is not the same as y	x is not y

No main function but it oke!

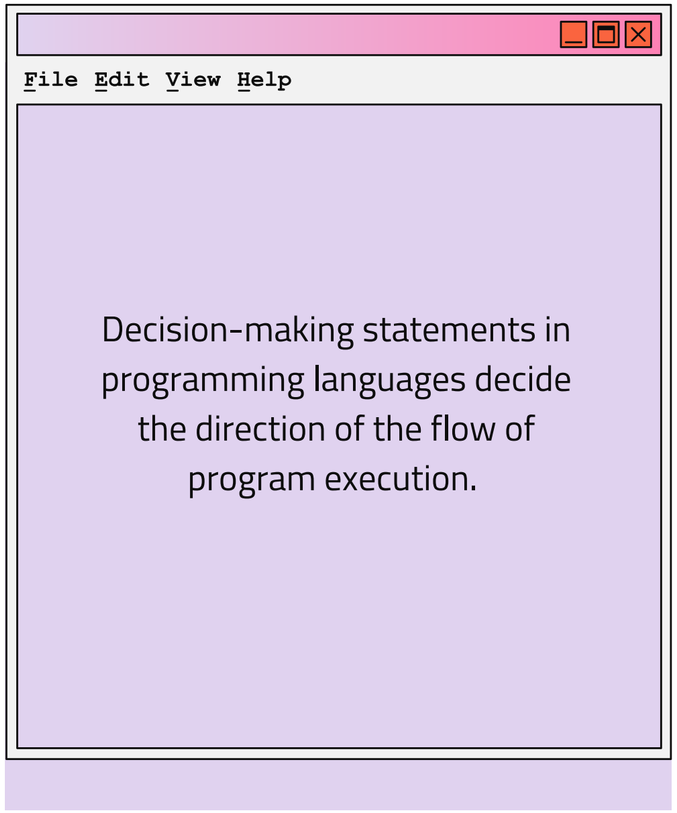
(JANUARY - MARCH 2019)

__name__
(A Special variable) in Python

- the source file is executed as the main program
- sets the __name__ variable to have a value "__main__".



Control statement



if statement

```
if (condition):
    # Executes this block if
    # condition is true
else:
    # Executes this block if
    # condition is false
```

match case statment

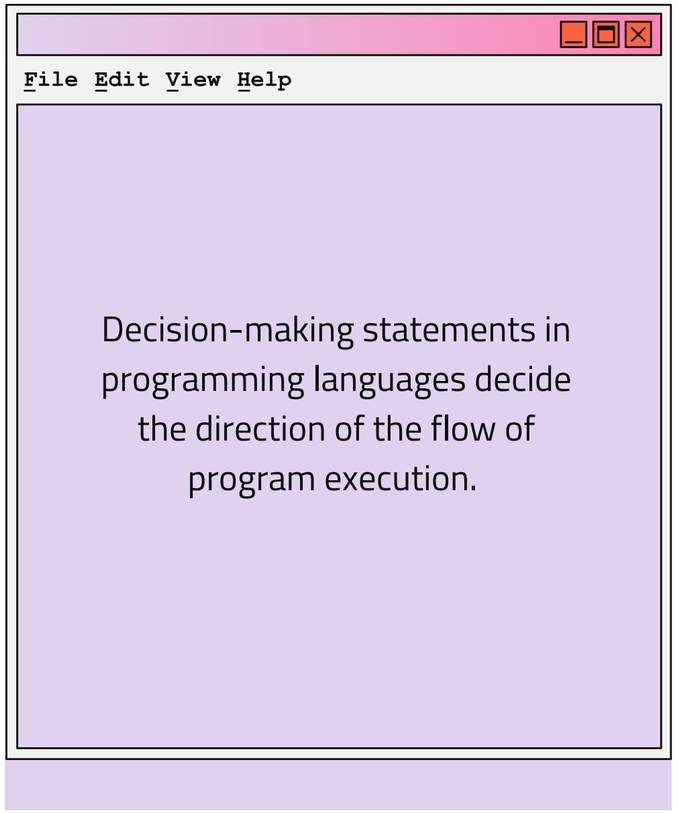
```
match parameter:

    case first  :
        do_something(first)

    case second :
          do_something(second)

    case third :
        do_something(third)
        ............
        ............
    case n :
        do_something(n)
    case _  :
          nothing_matched_function()
```

Control statement



for loop

```
for iterator_var in sequence:
    statements(s)
```

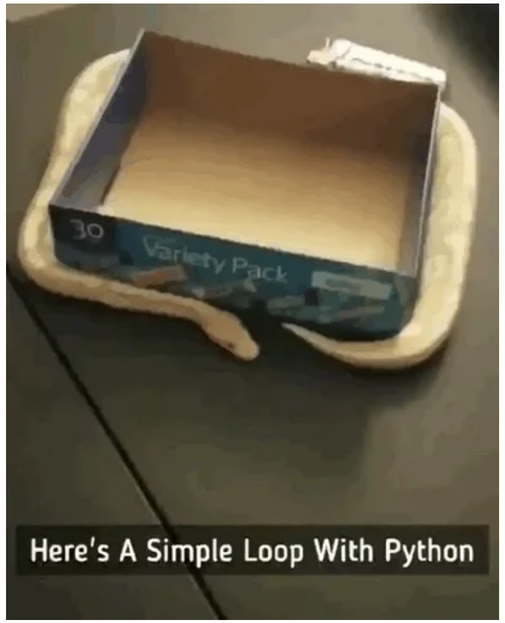
```
n = 4
for i in range(0, n):
    print(i)
```

```
l = ["geeks", "for", "geeks"]
for i in l:
    print(i)
```

```
cars = ["Aston" , "Audi", "McLaren "]
for x in enumerate(cars, start=1):
    print (x[0], x[1])
```

while loop

```
while condition:
    # execute these statements
else:
    # execute these statements
```



Function in python

```
def function_name(parameters):
        """docstring"""
        statement(s)
        return expression
```

Functions can be both built-in or user-defined. It helps the program to be concise, non-repetitive, and organized.

Variable length non-keywords argument

```
# Python program to illustrate
# *args for variable number of arguments
def myFun(*argv):
    for arg in argv:
        print(arg)


myFun('Hello', 'Welcome', 'to', 'GeeksforGeeks')
```

Variable length keyword arguments

```
# Python program to illustrate
# *kwargs for variable number of keyword arguments
def myFun(**kwargs):
    for key, value in kwargs.items():
        print("%s == %s" % (key, value))

# Driver code
myFun(first='Geeks', mid='for', last='Geeks')
```

Function in python

Anonymous functions

an anonymous function means that a function is without a name

```
# Python code to illustrate the cube of a number
# using lambda function


def cube(x): return x*x*x

cube_v2 = lambda x : x*x*x

print(cube(7))
print(cube_v2(7))
```

String

Strings are arrays of bytes representing Unicode characters

Creating a String in Python

Strings in Python can be created using single quotes or double quotes or even triple quotes.
Python3

Accessing characters in Python String

individual characters of a String can be accessed by using the method of Indexing

Deleting/Updating from a String

Updation or deletion of characters from a String is not allowed

Escape Sequencing in Python

Escape sequences start with a backslash

Formatting of Strings

the use of format() method which is a very versatile and powerful tool for formatting Strings

Old style formatting was done without the use of format method by using % operator Python3

String bonus

Strings are arrays of bytes representing
Unicode characters

String method

Pattern match

List,Tuple,Set, Diction

Document2 - Macrostuff Board

File Edit Format View Help

Cool New Font 12 B I U

Lists

Python Lists are just like dynamically sized arrays, declared in other languages (vector in C++ and ArrayList in Java).

Tuples

Tuple is a collection of Python objects much like a list. The sequence of values stored in a tuple can be of any type, and they are indexed by integers.

List,Tuple,Set, Diction

Document2 - Macrostuff Board

File Edit Format View Help

Cool New Font 12 B I U

Dictionary

Dictionary in Python is a collection of keys values, used to store data values like a map, which, unlike other data types which hold only a single value as an element.

Sets

set is an unordered collection of data type that is iterable, mutable and has no duplicate elements. The order of elements in a set is undefined though it may consist of various elements.

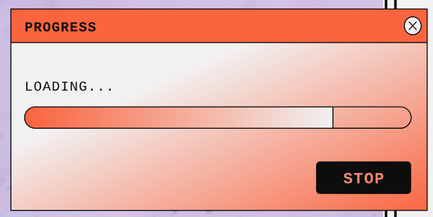
Exception & file handling

Untitled - TextEdit

File Edit View Help

Try & Except

Error in Python can be of two types i.e. Syntax errors and Exceptions. Errors are the problems in a program due to which the program will stop the execution. On the other hand, exceptions are raised when some internal events occur which changes the normal flow of the program.



preview.jpg

File Handling

Python treats files differently as text or binary and this is important. Each line of code includes a sequence of characters and they form a text file. Each line of a file is terminated with a special character, called the EOL or End of Line characters like comma {,} or newline character

Back to Agenda Page

OOP Python

Class & object

Class creates a user-defined data structure, which holds its own data members and member functions, which can be accessed and used by creating an instance of that class. A class is like a blueprint for an object.

Contructor, getter, setter

__init__ method
getters and setters are not the same as those in other object-oriented programming languages. Basically, the main purpose of using getters and setters in object-oriented programs is to ensure data encapsulation.
Private variables in python are not actually hidden fields like in other object oriented languages

Inheritentance

We can achieve this by using single underscore and double underscores. Python provides three types of access modifiers private, public, and protected.
- Public Member: Accessible anywhere from uotside oclass.
- Private Member: Accessible within the class
- Protected Member: Accessible within the class and its sub-classes

OOP Python

Polymorphism in Python

The word polymorphism means having many forms. In programming, polymorphism means the same function name (but different signatures) being used for different types.

Static variable

static variables in Python are declared inside a class and outside all the methods. This means that the static variable belongs to the class rather than any specific object of that class.

Classmethod & static method

1. A class method takes cls as the first parameter while a static method needs no specific parameters.
2. A class method can access or modify the class state while a static method can’t access or modify it.
3. In general, static methods know nothing about the class state. They are utility-type methods that take some parameters and work upon those parameters. On the other hand class methods must have class as a parameter.
4. We use @classmethod decorator in python to create a class method and we use @staticmethod decorator to create a static method in python.

Bonus python

Module

1. Single file contain code organized and help reuse code encapsulating related functionality into a separate file
2. break down program into smaller, manageable units, making it easier to understand, maintain, and reuse

Package

1. way to organize module into directory hierarchy.
2. Package one or more module
3. have special file is __init__.py to mark the directory as a Python package

Decorator

A Python decorator is a function that takes in a function and returns it by adding some functionality
- a decorator is a callable that returns a callable
- @ Symbol With Decorator : note function run before an function

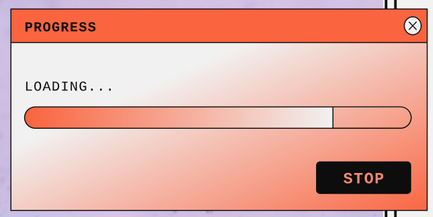
Python virtual

Untitled - TextEdit

File Edit View Help

Why we need

projects might require a different version of an external library than another one. If you have only one place to install packages, then you can't work with two different versions of the same library. This is one of the most common reasons for the recommendation to use a Python virtual environment.



preview.jpg

Who support python

1. venv: Built-in module in Python 3 for creating and managing virtual environments.
2. virtualenv: Third-party tool supporting virtual environments for both Python 2 and Python 3.
3. conda: Cross-platform package and environment manager with virtual environment support for Python and other languages.
4. Pipenv: Tool combining package and virtual environment management for Python.
5. Pyenv: Advanced Python version manager that also supports virtual environments.
6. Anaconda: Data science platform with a powerful environment manager for creating and managing virtual environments for Python.