REPL

Read Eval Print Loop

IDE

Integrated Development environment

Some IDEs:

PyCharm

IDLE (the one that I just launched)

Jupyter (the one that all of you have it open)

Spyder

Visual Studio

Eclipse

etc.....

domains where python is used:

DS AI ML

web Frameworks (django, flask)

DevOps

Testing

Automation

Embedded Systems

Telecom

Types of python files we come across:

.py (python)

.ipynb (iPython Notebooks)

## 

## Numbers:

int

float

complex numbers (6 + 7i)

## bool:

True

False

What can be False:

False (bool)

None

empty sequences → ‘ ‘, [ ], { }

value = 0

negative numbers are not false

## 

## Operators:

Arith:

+ add

- subtraction

\* multiply

/ division

% modulo (remainder)

// floor division

\*\* exponent

## Logical:

and

or

not

## Comparative :

< less than

> greater than

<= less than equal to

>= greater equal to

== equal to

!= not equal to

## membership operators:

in

not in

## identity Operators:

is

is not

## Memory:

python is luxurious in terms of memory

is optimised for speed first, memory later

user friendly

user can’t optimise memory much

NEVER use memory address or id in programming

|  |  |  |  |
| --- | --- | --- | --- |
| 10202 | earth | stra, strc |  |
| 10212 | world | strb |  |
|  |  |  |  |

memory is not such a constraint anymore

## 

## Strings:

double quotes or single quotes

indexed

negative index

sliced

upper range not included

does not give out-of-bound error

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| a | l | s | t | o | m |
| 0 | 1 | 2 | 3 | 4 | 5 |
| -6 | -5 | -4 | -3 | -2 | -1 |

immutable

## Lists:

square brackets [ ]

indexed

negative index

sliced

upper range not included

does not give out-of-bound error

heterogenous mixture

Nesting to any extent is allowed

mutable

lot of functions

## Tuples:

square brackets ( )

indexed

negative index

sliced

upper range not included

does not give out-of-bound error

heterogenous mixture

Nesting to any extent is allowed

immutable

none of the modification functions will work

very few functions

## Sets:

curly braces { }

no duplicates allowed

unordered

can not be indexed

can not be sliced

set as a whole is mutable

can add or remove members

members of a set should be immutable

## Dictionaries:

curly braces { }

unordered

can be indexed (key)

can not be sliced

dictionaries by themselves are mutable

keys:

immutable

no duplicates

values:

mutable

duplicates allowed

## Docstring

string that can extend multiple lines

generally used for documentation

also used for multi line comments

functions: first line of function

modules: first line of modules

class: first line of class

## 

## Flow Control:

no brackets for code

indentation decides block of code

if else elif

while

for break continue

range()

enumerate()

pass:

does nothing

place holder, come back later & extend the code

no difference to the logic or algorithm

else with for:

else is

else executes if break does not execute

else does not execute if the for loop breaks

## Functions:

print()

type()

id()

len()

sorted()

returns None by default

default values

named values (arguments)

variable num of arguments

variable num of key word arguments

## 

## Modules:

1. import apser

apser.blue()

apser.red()

apser.magenta()

1. import blue, magenta from apser

from apser import blue, magenta

blue()

magenta()

r~~ed()~~

1. import apser as basha

basha.blue()

basha.red()

basha.magenta()

1. import blue from apser as b

b()

1. from apser import \*

blue()

red()

magenta()

imported modules run once in our application

interpreter does not import repeatedly

to re-import use importlib.reload()

## Class:

everything is an object