## Python refresher

REPL

read eval print loop

Cpython

jython (java virtual machine)

data science

ML

AI

cloud computing

devops

automation

testing

embedded

web (django, flag)

~~cyber security~~

## data

### data types

### int

### float

### complex

### bool

True

False

### str

double or single quotes

index

negative index

slicing

upper range not included

no exceptions

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| c | o | r | o | n | a |
| 0 | 1 | 2 | 3 | 4 | 5 |
| -6 | -5 | -4 | -3 | -2 | -1 |

immutable

functions: upper, lower

title, capitalize,

replace

find, count

isalpha, isdigit, is alnum

### docstrings:

multilines

triple quotes

""" """

''' '''

\_\_doc\_\_

### comments:

# single line comments

""" """ **used** like multiline comments

### keywords:

False await else import pass

None break except in raise

True class finally is return

and for lambda try

as def from while nonlocal

assert del global not with

async elif if or yield

continue

## functions:

### general

print

len

type

### cast:

int

float

str

list()

tuple()

set()

range()

sorted()

max()

min()

any()

all()

map()

filter()

### user defined

def

can pass 0 to n number of values

can return 0 to n number of values

no function overloading

single definition (latest) exists

default values

named arguments

variable num of arguments: args

variable n um of keyword arguments: kwargs

global

recursion

nesting

### 

## operators:

### arth:

+

-

\*

/

% modulo

=

// floor division

\*\* power (exponent)

### logical:

and

or

not

### bitwise:

<<

>>

^

&

|

~

### relational (comparison)

<

>

<=

>=

==

!=

### identity

is

is not

### membership

in

not in

### 

### False

0

0.0

None

False

''

{}

[ ]

## Data structures

### list:

sequences

of any type

index

negative index

slicing

mutable

functions:

index, count

append, insert

pop, remove, clear

sort, reverse

### tuples:

sequences

of any type

index

negative index

slicing

immutable

functions:

index, count

### set:

sequences

no duplicates

unordered

immutable members

by itself set is mutable

no nest

### dict:

key: value pair

unordered

key:

immutable

no duplicates

values:  
 anything

mutables

nesting

by themselves dict is mutable

functions:

get

update

values, keys

items

pop

## control statements

### if else

if

elif

else

pass

while

for

else

## modules

1. import colours

colours.blue()

colours.green()

colours.yellow()

1. from colours import blue, green

blue()

green()

~~yellow()~~

1. from colours import \*

blue()

green()

yellow()

1. import colours as c

c.blue()

c.green()

c.yellow()

1. from colours import blue as b

b()

for reloading:

import importlib

importlib.reload(colours)

## files:

open

close

read()

read(n)

readline

readlines

write

seek

tell

readable()

writable()

closed

modes:

r read

w write (always a new file)

a append

r+ read & write

rb read in binary mode

## OOP

class

self

\_\_init\_\_

inheritance

encapsulation

\_\_ (double underscores)

polymorphism

operator overloading

magic methods

C++ operator , function

java function

python operator

\_\_init\_\_

\_\_del\_\_

\_\_repr\_\_

\_\_call\_\_

\_\_add\_\_

@classmethod

@staticmethod

## regex:

search pattern

https://regex101.com

metacharacters:

^ line starts with

$ lin ends with

[ ] set of characters

[^ ] invert of [ ]

. single character

\* zero or more occurrences of the pattern left to it

+ one or more occurrences of the pattern left to it

? zero or one occurrences of the pattern left to it

{x, y} atleast x times, max y times

{2, 4}

( ) group sub patterns

\d digits

\D non digits

\s whitespace

\t \n \r space \v

\S non whitespace

\w alphanumeric characters including underscores

[a-zA-Z0-9\_]

re

match

findall

split

sub

subn

## exception handling

try

except

finally

else

raise

1. exception occurs & is handled

except

finally

code continues

1. exception never occurs

else

finally

code continues

1. exception occurs, & is not handled

finally

code crashes

Common exceptions:

Exception Base class

AssertionError Raised when an assert statement fails.

EOFError Raised when the input() function hits end-of-file condition.

FloatingPointError Raised when a floating point operation fails.

GeneratorExit Raise when a generator's close() method is called.

ImportError Raised when the imported module is not found.

IndexError Raised when the index of a sequence is out of range.

KeyError Raised when a key is not found in a dictionary.

KeyboardInterrupt Raised when the user hits the interrupt key (Ctrl+C)

Also raised when you click on "interrupt" in Jupyter

MemoryError Raised when an operation runs out of memory.

NameError Raised when a variable is not found in local or global scope.

OSError Raised when system operation causes system related error.

OverflowError Raised when the result of an arithmetic operation is too large to be

represented.

RuntimeError Raised when an error does not fall under any other category.

StopIteration Raised by next() function to indicate that there is no further item to be

returned by iterator.

SyntaxError Raised by parser when syntax error is encountered.

IndentationError Raised when there is incorrect indentation.

TabError Raised when indentation consists of inconsistent tabs and spaces.

SystemError Raised when interpreter detects internal error.

TypeError Raised when a function or operation is applied to an object of incorrect

type.

UnboundLocalError Raised when a reference is made to a local variable in a function or

method, but no value has been bound to that variable.

ValueError Raised when a function gets an argument of correct type but improper

value.

ZeroDivisionError Raised when the second operand of division or modulo operation is zero.

more exceptions at:

<https://docs.python.org/3/library/exceptions.html>

# concurrency:

### threading:

import threading

ta = threading.Thread()

ta.start()

ta.join()

l = threading.Lock()

l.acquire()

l.release()

### multiprocessing

pa = multiprocessing.Process()

pa.start()

pa.join()

## anonymous functions

### lambdas

lambda

## generator

yield

## pickle:

convert into a byte stream

object serialization

serialization:

marshal

pickle

json

xml

## json

python json

dict object

list, tuple array

str string

int, float number

True true

False false

None null

load load a json file

dump to write into a json

loads load from a json object

dumps dump into a json object

## xml

extensible markup language

## yaml:

yet another markup language

yai’nt another markup language

parked topics:

~~list comprehensions~~

bitwise operations

~~XML~~

~~YAM~~L

REST API

date & time

globals() locals()

python + C

python + DLL

modules & packages:

package

update path

fetch from online

\_\_main\_\_

~~logging & extracting exceptions~~

debugging

time.time()

timeit