## environments

IDLE

REPL

Read Evaluate Print Loop

## Data types:

### Numbers:

int

float

complex numbers (6 + 5j)

### Strings:

“ “

‘ ‘

### bool

True

False

## 

## Operators

### arth :

+ add

- sub

\* mul

/ div

% modulo

// floor division

\*\* power of (exponent)

= assignment

BEDMAS

~~increment, decrement~~

### logical:

and

or

not

### relational (comparative)

< less than

> greater than

<= less than equal to

>= greater than equal to

== equal to

!= not equal to

### bitwise

& and

| or

^ ex-or

<< left shift

>> right shift

~ not

### identity:

is

is not

### membership:

in

not in

### comments:

### #

### False

False

0

[] ‘ ‘ { } ( ,)

None

### General Functions:

print()

type()

len()

input()

range()

all()

any()

sum()

sorted()

### conversion functions (cast):

int()

str()

float()

list()

tuple()

set()

### Strings:

place = “earth”

firm = “toshiba”

moon = "europa"

planet = 'mars'

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| e | u | r | o | p | a |
| 0 | 1 | 2 | 3 | 4 | 5 |
| -6 | -5 | -4 | -3 | -2 | -1 |

text

indexed

negative indexed

slicing (upper range not included)

step

immutable

functions:

upper

lower

index

rindex

replace

count

….

….

….

### docstrings:

triple double quotes

triple single quotes

multiple lines

programmers use it sometimes as multiline comments

## Data structures:

### List:

heterogeneous data

indexed

negative indexed

slicing (upper range not included)

step

mutable

functions:

count

reverse

append

extend

pop

….

….

….

### Tuple:

heterogeneous data

indexed

negative indexed

slicing (upper range not included)

step

immutable

functions:

count

index

### Set:

no duplicates

no order

heterogeneous data (immutable)

not indexed

mutable

| union

& intersection

^ symmetric difference

- difference

### Dict:

no order

keys:

immutable

no duplicates

values:

can be anything

indexed using keys

mutable

### misc\_1:

del

None

## functions:

def

return

default values

named arguments

variable number of arguments(\*)

global

#

variable number of keyword arguments (\*\*)

inner functions

nonlocal

### modules

1. import colour

colour.blue()

colour.yellow()

colour.green()

1. from colour import blue, yellow

blue()

yellow()

~~green()~~

1. from colour import \*

blue()

yellow()

green()

1. import colour as c

c.blue()

c.yellow()

c.green()

1. from colour import blue as b

b()

### special:

filter

map

lambda

generator

## Files:

open()

modes:

r read

w write (deletes contents of old file)

treats it like a new file

a append mode

r+ read & write

fa.read()

fa.readlines()

fa.readline()

fa.write()

fa.writable

fa.readable

fa.closed

os module

### escape sequences:

\n new line

\b backspace

\t tab

\v vertical tab

## 

## regex:

. placeholder (one character)

^ start of line

$ end of line

[ ] set of chars

[a-z] all lowercase

[0-9] all numbers

[a-e] [abcde]

[^ ] invert of the set

+ one or more occurrences

? zero or one occurrence

\* zero or more

( ) group

| or  
{ } number of characters

{2, 4} 2 to 4

{2} exactly 2 characters

{2,} 2 or bigger than 2

\b beginning\end of a word

\B not at the beginning or end of a word

\d any decimal digit

[0-9]

\D any non decimal digit

\s whitespace

(space, \n, \t,...)

\S non whitespace

\w alphanumeric character (includes underscore)

\W non alphanumeric character

## 

## Multiprocessing Multithreading

Process()

Thread()

### Time:

timeit.timeit()

start = datetime.datetime.now()

# your code

# your code

end = datetime.datetime.now()

end - start

## Exception Handling

try always followed by an except

no other code in between

multiple except(s) are allowed

except :

handles all exceptions

should always be at the end

can be nested

an inner exception can be handled outside