

name_1 = sunny

name_2 = buntty

print (f' Hello {name_1} & {name_2}')

→ Hello sunny buntty

print ('Hello there { }, { }'.format(name_1, name_2))

↳ Hello there ~~name_1~~ sunny buntty

print ('Hello there %s and %s' % (name_1, name_2))

TRUE | BOOLEAN | FALSE

bool(True)

bool(False)

Print (bool(None))

(bool(True))

(bool(False))

(bool(0))

(bool(1))

(bool(0))

→ Don't overwrite keywords in the .py file.

OPERATOR PRECEDENCE

②

- () ——— ① ——— Braces
- ** ——— ② ——— To the Power
- * / ——— ③ ——— Mul and Div
- + - ——— ④ ——— Add & Sub

bin() Complex

These are Extra data type

Complex (real, [imag]) → Complex Number
(real + imag * 1j)

imag = 0

Control + shift + T

print(bin(5)) ——— 0b101

print(int('0b101', 2)) — 5

Variables

- Snake - Case
- start with lowercase or underscore only, not uppercase
- Letters, numbers, underscores
- Case sensitive

Expressions Vs Statements

a, b, c = 1, 2, 3

print(a) 1

print(b) 2

print(c) 3

// recursively ads, a=1, b=2, c=3

Augmented Assignment Operator

some_value = 5

some_value = +2 Incrementing Value

pt(some_value) = 7 //

some_value = some_value * 2

Immutable

selfish = '8121'

selfish[0] = '9'

print(selfish) error

Traceback → TypeError 'str' object does not support item assignment.

selfish + '8'

O/P
print → 8121(8) → 8 is added

Boolean

True or False.

bool(0) → False

bool(1) → True.

List

Unlike strings & other d.T, lists are mutable sequences in python.

my_list = [1, 2, '3', True, "Alpha"]

→ we can denote them by

~~Tuple~~
[<List name> = []]
SYNTAX

len(my_list)

↳ All the functions & Methods are applicable on lists.

↳ [] are also used for matrix & 2D lists

`print (my-list[3])`

list
Slicing is also available and called [start: end]

`my-list * 2`

↳ The list get doubled.

i.e The contents of list are added (Appended)

`my-list + [100]`

↳ Doesn't Mutate original list Creates a new one.

`my-list.insert`

- `extend([100, 200])`
- `append(100)`

1) `join`

Copying

`basket = ['apples', 'fruits', 'Oranges']`

`new-basket = basket.copy()`

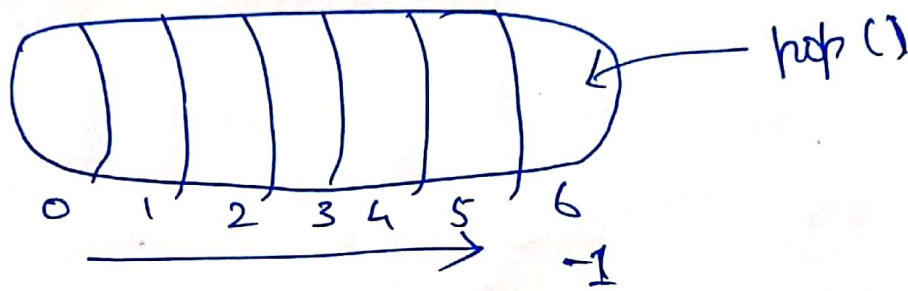
`new-basket 2 = basket[:]`

// so, these thing Directly Copies the whole list
in order using Own functions we use building
functions/methods.

we can also remove from the list by using pop method.

`[1, 2, 3]. pop()`

default index of in pop method is -1 (last item)



`[1, 2, 3]. pop(1)`
0 1 2
Index is Mentioned
2 is removed.

- remove
- clear

`del [1, 2, 3][0]`

`del` is a predefined keyword in python

ordering

`[1, 2, 5, 3]. sort()` → Mutate

`[1, 2, 5, 3]. sort(reverse=True)` → Mutate

`[1, 2, 5, 3]. reverse` → Mutate

`sorted([1, 2, 5, 3])` → reversed (New list)