

1) We can have nested list in python.

$x = [1, 2, [3, 4, 5], 6, 7]$ .

print ~~x[2][i]~~ # 4

Python variables are simply references.

2) pass command does nothing

3) complex by python is just  $3 + 4j$ .

4) Collection in python.

str → String (unicode istring)

5)  $a = (1, 2, 3)$  → tuple → <sup>ordered</sup> immutable → hashable if all elements hashable

6)  $a = [1, 2, 3]$  → list ordered collection  
not hashable → mutable

7)  $a = \{1, 2, 'a'\}$  → set unordered collection unique values

8)  $a = \{1: 'one', 2: 'two', "three": 3\}$  → dict / hash table

Key has to be hashable i.e. immutable. Python compares to other objects.

M	T	W	T	F	S	Notes
2	3	4	5	6	7	
9	10	11	12	13	14	
16	17	18	19	20	21	
23	24	25	26	27	28	

Q) In python we use None (like null in java).  
None is less than any number.

Q) To check if variable is None

if x is None  
print(None).

Q) None is obj of dataType → None Type

Q) None does not imply empty list (not of like null)

Q) You can't create multiple None but can assign  
variable to it.  
i.e. None is Singleton object

Q) Functions that won't return anything returns  
None obj

Q) is used in Python to test objects identity  
while ~~=~~ is used to test if 2 ~~obj~~ are equal  
not.

Q) is used to test if 2 variants refer same obj  
ref.

Q) == is used to test if 2 variables are equal or  
↳ can also check condition list / ~~dict~~ / ~~set~~  
tuple

January

1st Week 003-362

03

SATURDAY

`reversed()` returns a reversed iterator of type `reversed`.

`a = reversed("abc")`

`a`  
0/first reversed. at `0x.....`

for name in a:

`print(name)`

`O/p - c`

`b`

`a`

`type(a) → reversed` reversed.

SUNDAY 04

1st Week 004-361

Unlike list & dict; String & tuple are  
also `None`, `True`, `False` are immutable.

So

`[ ] == []`    `{ } == { }`

`True`

`[ ] is [ ]`

`False`

`True`

`{ } is { }`

`False`

If `abs`, `int`, `long`,  
`float`, `complex`, `byte`,

`rozen` not are  
immutable.

but →

05

January

2nd Weeks 005-360

MONDAY

`' == == '` → True      `() == ()` → True  
`f '' in ''` → True.      also      `f () in ()` → True

19) To convert between data types  
Simple use.

`a = "123"`  
`b = int(a)`  
→ rem. `int(a)` not `(int)`.

`c = "123.23"`  
`d = float(c)`

~~`a = "hello"`~~  
~~`b = int(a)`~~ → `[ 'h', 'e', 'l', 'l', 'o' ]`  
~~`c = set(a)`~~ → `{ 'o', 'e', 'l', 'h' }`  
Since set is un

20) In python everything is obj (no primitive)

Int      is string too

21) `id` is used to check the identity of object.

Notes

`a = 123456789`

`a = [ ]`  
`id(a)`  
`15476372`

January

2nd Weeks 006-359

06

TUESDAY

method for list

but for list.

Unit 1 = {1, 2, 3}.

id(list 1)

307812801

Unit 1[3] = 4.

id(list 1)

Same,

~~input() is used to take user input~~

Don't use in 2<sup>2</sup>  
input() in 2<sup>2</sup>  
will evaluate  
one of remain

Use raw\_input() to get user input. (Python 2<sup>2</sup>)

input() " " , " " (Python 3<sup>x</sup>)

~~raw\_input("String")~~ (or ~~input()~~ input))

will display string & wait for  
user input.

~~raw\_input() returns the value~~

07

January

2nd Weeks 007-358

WEDNESDAY

22) To find ASCII value use  
 $\text{ord}('s') = \underline{\underline{115}}$

23) Negative indexing means counting from end of list

24) List methods

1) `append()` → add to list.

2) `.insert(index, element)` → insert before index

3) `.remove(item)` → removes

4) `.pop(index)` → removing at index  
default is last  $\underline{\underline{=}}$

25) Iterate → for  $\underline{\underline{a}}$  element in list:  
`print(item)`

26) Iterate through dict:  
 $a = \{1: 1, 2: 2, 3: 3\}$

for k in a.keys():

`print("{} is {}".format(k, a[k]))`

Notes

Important

S	M	T
1	2	3
8	9	12
15	16	19
22	23	26

January

2nd Weeks 008-357

08

THURSDAY

"{} is awesome" .format(" Marshm")  
↓

"Marshm is awesome"

Set is very useful as.

it is a collection of elements.  
with no duplicates & no  
insertion order but  
sorted order.

pow(2,4) = 2<sup>4</sup>

help(map) → Very Tuf help(dict)

→ displays what map does

→ returns  
all members  
of class

To import a module  
simply say ~~one~~ import modulename  
import function of module.  
from modulename import functionName.

09

January

2nd Weeks 009-356

FRIDAY

33) To detect a module by directory if it would have \_\_init\_\_.py in it.

34) and; or; → logical express.

35)  $5 \text{ in } [1, 3, 5]$  → returns true

↳ check if list / str / tuple - in contains a value.

36) there is no else if in python if it is elif

37)  $\frac{2}{3} \rightarrow 0$

$\frac{2}{3.0}$  or  $2.0/3 \rightarrow \underline{\underline{.66}}$

from operator import truediv

truediv(2,3) → .66

## UNICODE & ASCII

ASCII - 7 bits

UNICODE → +

UTF-8 → 1-4 bytes

Notes

Hexadecimal maps to exactly 4 binary digits

S	M	T
1	2	3
8	9	10
15	16	17
22	23	24

January

2nd Weeks 010-355

10  
SATURDAY

To sort in python say ~~sort & (d)~~

Concept of leaked variable in list comprehension of for loops.

$x = [x + 'hello' for x in 'AEIOU']$

Vowels = 'E'

for x in 'AEIOU'

vowel.append(x)

print(x)  
~~# → 'V'~~

$x = 'helloworld'$

Vowels = 'E' for x in 'AEIOU' SUNDAY 11  
2nd Weeks 0011-354

print(x)  
= 'V' → Python 2.2  
= helloworld → ' 3'

ord('a') → 97 | int('1000001', 2) = 85

chr(97) = 'a'

format(15, '08x') → 0000000F

T	F	S
5	6	7
12	13	14
19	20	21
26	27	28

Lox width X base

→ binary or octal

12

MONDAY

January

3rd Weeks 012-353

2015

to odd & hex no.

$a = \text{format}(14, "08x") \rightarrow \underline{\text{return char}}$

$b = \text{format}(13, "08x")$

$c = \text{format}(\text{int}(a, 16) + \text{int}(b, 16), "08x")$

$\downarrow$   
return decimal)

or

$a = \text{hex}(14)$

$b = \text{hex}(13)$

$c = \text{format}(\text{int}(a, 16) + \text{int}(b, 16))$

Notes

S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28