1] Python Data Type

2] loops

Data Type ⇒ type of data

identity of your data present in your variable

↓

where we store data

x , y , z

a = 10

w = 10.5

type(_) → what is the type of that data

In python, we don't have char

In python, our variables are unbounded (int)

14 data types 6 categories

Numeric	Boolean	Sequence	Set	Mapping	None
int ✓	bool ✓	str ✓	set	dict	None Type
float ✓		list	frozen set		
complex ✓		bytes			
		bytearray			
		tuple			
		range			

Properties of data types in python

1. Data Types are dynamic

⇓

Keep on changing

a variable can have diff. data type at different point in code

C++ / Java	vs	Python
int a		a
string		
float		

2. Size of datatype is also dynamic

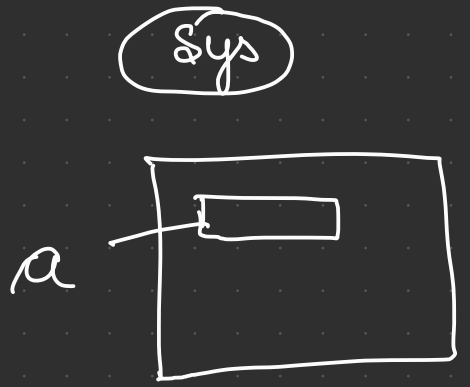
int a 2/(4) bytes compiler

a ⇒

5 or five

int a

getsizeof( )



3. data types are unbounded

a = 4 ways

min max

because size is dynamic

Numeric data Type

int ⇒ numbers but without fractional part

a=10 -10 4 ~~4.54~~

Float ⇒ Python supports fraction / floating points

. 2.15 .

3.14

Just like Java/C, floating point precision value is 16

Exponention

e^5 $e^x \Longrightarrow 10$

10^x

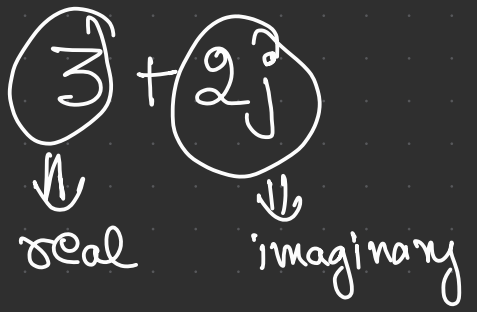
$e = 2 \sim 3$

$3.5\,e^{1} = 35$

$3.5\,e^{-1} = .35$

$5.14\,e^{-(-1)} = 51.4$

Complex



$j = \sqrt{-1}$

j/J can only be used.

i + j

m + n ✗

2 + 3j ✓

2 + j3 ✗

Boolean

T/F
1/0

true
false ✗

True
False ✓

Conditional statements ★★

In backend, True stored as 1
False stored as 0

True + True ⇒ 2
False + True = 1

String ★★

sequence of characters

no char type like C++/Jav

' ' " "

for strings in python

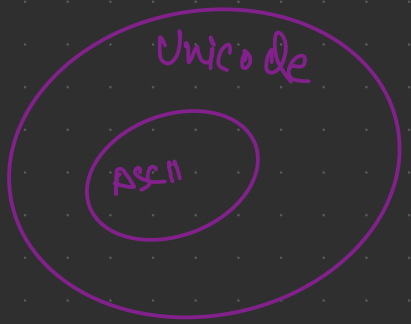
python recommends this

It understands just 0 & 1

⟹ ASCII / Unicodes

&var = 'm'

creates a memory → location | type | value



Unicode / ASCII just a table or conversion

' ' " "

mayanK's Code

let us "Code"

let's "Code"

imp. examples

triple Quotes

Backslash.

How to access/print Individual Characters

Str = 'P Y T H O N'

0 1 2 3 4 5

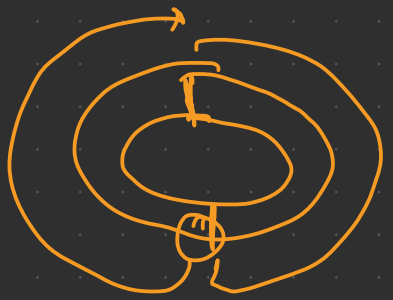
len = 6

Str[0] = P

Str[1] = Y

Str[5] = N

Negative Indexing



0	1	2	3	4	5
P	Y	T	H	O	N
-6	-5	-4	-3	-2	-1

str[-1] = N

str[-6] = P

str[-7] = out of bounds

$-len(str) \leq -6 < -1 \leq -1$

+ve 0 <= < len - 1

-ve - len <= < -1

If above or beyond, then out of range
(len-1) (-len)

Change -ve index to +ve index

-1 → (-1 + len)

Python

-1 and 5

-6 and 0

Type Conversion

process of converting type of one variable to another

1. Implicit
2. Explicit

Implicit

⇒ Python does it automatically

⇒ No programmer's involvement

a = 5 (int) b = 10.5 (float)

c = 5 + 10.5
int float

c will be float

Python always converts to larger data type so as to save the information

5. → 5

5.5 → 5

float can store all values of int + decimal

Explicit Type Conversion

cost/rax = 100.533 float

(int) 100

5 predefined function

1. int ( )
2. float ( )
3. complex ( )
4. Bool ( ) ⟶ only 0 is false
5. Str ( )

False values for bool() fn

- None
- False
- 0, 0.0, 0+0j → false
- {}, (), [] →