Date __/__/____ Page No.:____

Day-2, # tip calculator!

Today concepts - Data Types
Numbers
Operations
Type conversion
f-strings

⇒ Data types

String ⇒ as we know also called concatenation

- Print ("Hello")
  print ("123")
  print ("123" + "123")
  output ⇒ 123 +23

- everything you put b/w double quotes is string ("" ↓ "") put anything

Integer = Whole no, without decimal

123, print (123)

- we use int to identifies user type

subscripting

ex - "Hello"

hum easily koi bhi word ko pick kr sakte by using index

⇒ H E L L O → string
0 1 2 3 4 → index

if i put like that code
print ("Hello"[0])

output → H

Bonus → Index should be like that

H E L L O
-5 -4 -3 -2 -1

ABHINAV

Date __/__/____ Page No.:____

Large Integers, as we know, to determine correct/understandable we use ↓

print (123_456_789)

Output → 123456789

Float = Floating point Number, using by decimal number.

It should be like that

print (3.1456)

Output ⇒ 3.1456

Boolean → only two possible terms.

- True
- False

⇒ Print (True)
print (false)

help to identifies the possible terms

Bonus → True or false start with capital word, and don't have any " ", _, ', , these type of symbols

ABHINAV

Date __/__/____ Page No.:____

→ Type Error ↓

as we know

- len("Hello") Find out the length of a string
- But for integer

len(123)
↳ wrong

as len-function don't work with integers

Bonus → , it's basically saying that in a ideal situation, a python file should only one blank line

ex 1 → particular
2 - blank -
3 - blank -
4 - blank →
} Wrong

at the end it should be like that ↓

1 → particular code end
2 → - blank -
} correct

→ Type checking → To check which data type we are using ex →

output

print(type("Hello")) → <class 'str'>
print(type(123)) → <class 'int'>
print(type(1.23)) → <class 'float'>
print(type(True)) → <class 'bool'>

ABHINAV

Date __/__/____ Page No.:____

→ Type conversion → It simple like to convert data type according to our need.

But their are some rule

as you can see → print(int("123"))

output → 123 , type checking

↳ <class 'int'>

- print(int("123") + int("456"))

output → 579 # convert str into int

- print(int("ABC") + int("456"))

Output → Show , value error as "ABC" can't be able to convert into str.

We simply using these conversion ↓

Problem → Print("No of letter:" + len(input("Enter your name"))

Solve this problem.

- int()
- float()
- str()
- bool()

→ ~~print~~ Name = input("Enter your name:")

length = len(Name)

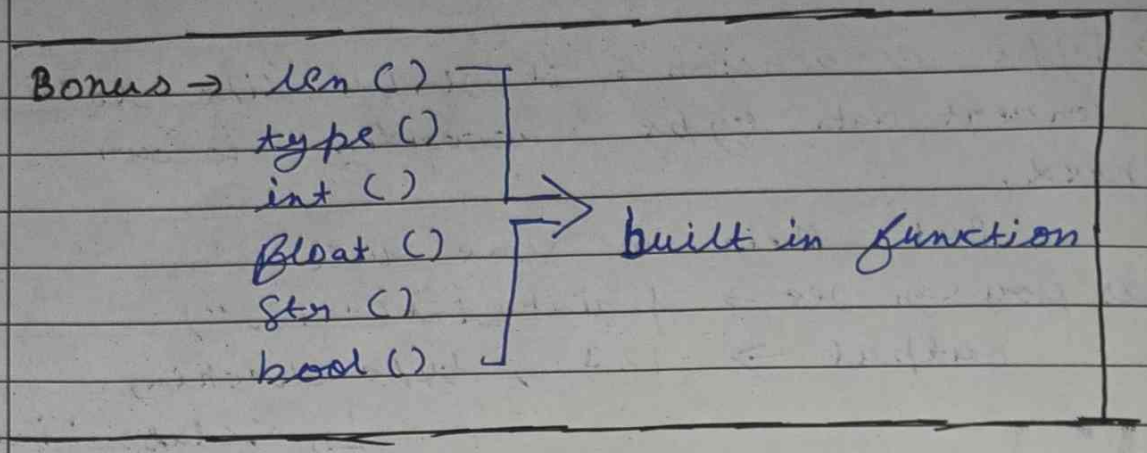
print("No of letter:" + str(length))

output → no of letter: 5

if user take Aryan as name..

ABHINAV

Date __/__/____ Page No.:____



→ Mathematical operations → it should be basic mathematical operator, +, -, *, /, // and **

Ex - print (123 + 456) # Addition
print (123 - 4) # Substraction
print (3 * 2) # multiplication
print (5 / 3) # Division show decimal
print (5 // 3) # not show value after dec.
print (2 ** 3) # 3 is power of 2
2^3, Exponent

PEMDAS, is rule for python mathematic operation

P - Parentheses - ( )
E - Exponent - **
M - Multiplication - *
D - Division - /
A - Addition - +
S - Substraction - -

It simply show the priority from P to S.

ABHINAV

Date ___/___/_____ Page No.:_____

Problem - print (3 * 3 + 3 / 3 - 3), solve according to PEMDAS rule.

Solution - 7

bcz, First took - 3 * 3 = 9
Next = 3 / 3 = 1
Then = 9 + 1 - 3
So, = 10 - 3
final = 7.0

Problem - WAC to built BMI calculator

Height = 1.65
Weight = 84

bmi = Weight / (Height ** 2)

print (bmi)

Output - 30.853.....

Bonus - if you do like this

print (int(bmi))
Output - 30 # it don't show values after decimal

ABHINAV

Date __/__/____ Page No: _____

→ Number Manipulation

Flooring a Number - You can floor a number or remove all decimal places using int() # built in function

Ex -

print (int (3.74879))

Output → 3

Rounding a Number - It help to do round or simplifies the decimal issues by using round() # built in function

Ex -

print (round (3.745.6)

Output → 4 # as it round 3.7 → 4
if it 3.3 → 3
according value after decimal (<5 → below
>5 → upper)

· print (round (3.7456, 2)
↳ (value after decimal you want)

Output → 30.85

Date ___/___/_____ Page No.:_____

Assignment Operators → it simple like this →

+=, -=, /=, *=, %

as you take
```
score = 0
score += 1   # 0+1
Print ( score)
```
Output - 1

as you take
```
score = 2
score -= 1   # 2-1
Print (score)
```
Output - 1

it already depends on you

ABHINAV

Page No.

Date / /

→ f - Strings → It help us for painful job like changing data type conversion can according to data type, it help us for these thing to resolve.

- It simple insert value into ~~a expression~~ a string.

Ex → score = 0
height = 1.5
winning = True.

Print (f "Your score is = {score}, height = {height}.
winning is = {winning})

f - string