#Day 02

len() → a predefined function to known the length of data passed in.

→ Datatypes.

1. Strings → "Hello"

To create strings we have to add double/single opening and closing quoatition

- Because of string we can pull out each alphabet or each item through indexing of strings.

[ ]

Here goes the index which I have to retrieve

The index starts from 0.

The method is also known as subscripting.

2. Number Datatype

I. Integer. → 123

This datatype is whole number not decimal. (positive & negative both are allowed)

- Sometime we have to use large number and that is harder to read so generally the idea is to put , with the number for much easier readibility

In python we can do that by replacing , with _ (underscore)

eg, 123, 456, 789

↓

123_456_789

And, python automatically convert those to commas.

II. Float

This is also known as floating point number.

eg. 123.456.

3. Boolean

There is only two option in this datatype

I. True II. False.

Both True and False starts with capital letter and don't have any type of quoation or other stuffs.

- This is generally used in conditional statement.

When we pass Integer to len() function, it gives us error (Type error).

len() function is a prefined function where it only works with string datatype, so it didn't proccess with Integer datatype.

So what can we do.

well we can use type conversion function.

firstly,

type( ) → Its a function that return the type of data that was paned into paranthese.

Str( ) → Its convert the variable to string datatype.

new_int = str(Integer).

Now, new_int is actually a string and NOW, len() function will not give any error with the variable passed, as we typecasted (converted the datatype) the integer.

```
a = 123.

type(a)
print(type(a)) ⟶ <class 'int'>

a = str(123)
type(a)
print(type(a)) ⟶ <class 'str'>

a = float(123)
type(a) ⟶ <class 'float'>
print(a) ⟶ 123.0
```

float()

This is a type conversion function that convert number (Integers) to floating number.

eg. 143 ⟶ 143.0 (value will be same)

float also works with string datatype

float("100.01") ⟶ 100.01

Then, we can do the arithmetic calculation as we required.

```
>>> print(70 + float("100.01"))
```

↳ This will be converted to floating number

→ will add both

→ will print the value.

⟶ 170.01 (output).

Coding Exercise.

For detailed question, refer to Github Repository.

- Example input. → 39
  Example output → 3+9 = 12.

Type a two digit number:

Solutions:-

```
input = input("Type a two digit number: ")
number = input.
first_number = number[0]
Second_number = number[1]
print(first_number + second_number)
```

The indexing will not work as we are not dealing with Str now. For that we can use Str() function to convert to string.

This will work, so we can optimise the code little bit

```
① input = input("Type a two digit number: ")
② number = Str(input)
③ first_number = number[0]
④ Second_number = number[1]
⑤ print(first_number + second_number)
```

However we are not assigning any datatype at the time of taking input and in python it is by default set to String datatype,

So basically we do not need to convert it again to string at line 2.

But, it raise error at line 5.

So, we need to modify them into Str.

At line 5.

```
print(int(first_number) + int(second_number))
```

int() function convert datatype to integer, if passed a float number, the decimal point and the numbers after that will be removed.

Mathematical operator.

- Addition → +
- Subtraction → -
- Multiplication → *
- Division → / (forward slash)
  - ↳ always return a float datatype.
- Exponent or raise to power → **

  eg. $2**2 = 2^2 = 4$.

- ~~e~~

Here also we have to follow, PEMDAS. LR

↳ It stand for it goes from left to Right.

Parenthese — ( )
Exponents — **
Multiplication — *
Division — /
Addition — +
Subtraction. — -

PEMDAS. →

However, *, / & +, - are equally placed while executing the operation.

eg.- 3*3 + 3/3 - 3

→ 9 + 1 - 3 (*, / both are done in same time)

→ 10 - 3

→ 7.0

↳ It return float data type.

** recommended debugging. for better understanding.

Let get 3.0 as result, we have modify the order

→ print ((3*(3 + 3))/ 3 - 3)

↓

(3* 6) / 3 - 3

→ 18/3 - 3

→ 6 - 3

→ 3.0 (Cool!!)

Coding Exercise 2.2

Calculate the BMI.

$$BMI = \frac{\text{weight Kg}}{\text{height}^2 (m^2)}$$

Solution:

```
height = input ("enter your height in m: ")
weight = input ("enter your weight in Kg: ")
BMI = weight/(height**2)
print (BMI)
```

>> Typeerror: unsupported operand type(s) for ** or pow()
'str' and 'int'.

So, we have to convert the height and weight to 'int.

```
BMI = int(weight) / int(height)** 2
print (BMI)
```

↑ for height float may be more suitable datatype.

⟹

```
BMI = int(weight) / float(height)** 2
print ('int(BMI))
```

↑ we added int() to avoid floating point.

In python, we can round a big float number alternatively to converting to string.

we use → round() → function.

So, whatever value we want to round we passed it in paranthese.

like

8/3 ⟶ 2.666665

round (8/3) ⟶ 3

#rounded.

round function also takes another parameter which decide the place precious upto which point the round should be done.

like.

round (8/3, 2) → set to 2 place precison

→ another optional parameter

⟹ 2.67. → 2 decimal point.

Also, if we want the division return integer by default inplace of using division (/) we can use floor division (//)

(8//3) ⟶ 2.

type (8//3) ⟶ <class 'int'>

F String.

We have seen that we can't find different datatype in single operation, all must be str.

Beside converting others to String, we can use F string.

Course : Python

Date : 12.02

Day : 02

```
>>> print("Hey Today is" + str(Date) + "& [redacted] Day" +
        str(02) + "Doing" + str([redacted] course))
```

→ Painful embedding!!

Using F String.

```
print(f"Hey Today is {Date} & Day {Day} Doing {course}")
```

→ Hey Today is (12.02) & Day (02) Doing (python)

Coding Exercise 2.3

For Question refer to Repository.

~~Solution~~:

1 year = 365 Days

1 year = 52 weeks

1 year = 12 months.

final-age = 90

Type checking
Type casting.

age → 20

remaing-age = 90 - 20 → 70

Days = 365 * 70
weeks = 52 * 70
Months = 12 * 70

} Data → print it using f string.

Solution:-

```
age = input ("What is your current age? ")
age = int (age) # converting str to int.
Max_age = 90
Rem_age = Max_age - age # calculating the Remaing year.
Rem_days = Rem_age * 365
Rem_weeks = Rem_age * 52
Rem_months = Rem_age * 12
print(f" You have {Rem_days} days, {Rem_weeks} weeks,
and {Rem_months} months left.)
```

(Cool!!)

Day 02 project.

Tip calculator:-

```
print(" welcome to the tip calculator.")
bill = input(" what was the total bill?")
bill = float(bill)
percentage = input(" what percentage tip would you like to
give?)
percentage = float(percentage).
Amount = bill + (percentage * bill)/100
no_people = input("How many people to split the bill?")
no_people = int(no_people)
Amount_each = (Amount/no_people) # round(Amount/no_people,2)
print(f"Each person should pay: {Amount_each}")
```