Built-in function

How to close Python interpreter?

- exit()
- quit()

In [ ]:

```
# print("hello world")
```

variable

```
- Just a label
- It is just a reference to certain memory location
```

input -> functions -> output

```
void-function-name(<input>):
    # logic will only do certain operations in input


return-function-name(<input>):
    # logic to perform certain operations on input and will give some output
```

Types of functions

- void function
  - void functions just perform the task
  - By default return value is `None`

    Examples,

    ```
    print
    help
    ```

- return function return functions return certain value

  Example,

  ```
  input()
  type()
  int()
  float()
  str()
  bool()
  ```

In [2]:

```
foo = print("hello")
print(foo)
```

```
hello
None
```

`input` function

- input function takes a string as `input`
- the same string will be printed as message to the end user
- whatever value user enter, gets converted into data type `str`

In [14]:

```
foo = input("Please enter your name: ") # function takes input from user
foo
```

```
Please enter your name: prashant
```

Out[14]:

```
'prashant'
```

In [5]:

```
bar = input("Please enter your profession : ")
```

```
Please enter your profession engineer
```

In [6]:

```
print(bar)
```

```
engineer
```

In [8]:

```
my_name = input("please enter your name : ")
type(my_name)
```

```
please enter your name : Rohit
```

Out[8]:

```
str
```

- programming is all about manipulating data
- data sources

    - databases
    - internet
    - file
    - user

what are basic data types in Python?

What are basic data types in Python?

- int (1, 2, 3, 4, 5, 6,...)
- float (20.3, 0.6,....)
- bool (True, False)
- str ("prashant", "python")
- None

- String is a collection of characters
- string is anything enclosed in double quotes/ single quotes / tripple quotes

In [9]:

```
help("keywords")
```

```
Here is a list of the Python keywords.  Enter any keyword to get more
help.

False               class               from                or
None                continue            global              pass
True                def                 if                  raise
and                 del                 import              return
as                  elif                in                  try
assert              else                is                  while
async               except              lambda              with
await               finally             nonlocal            yield
break               for                 not
```

type casting

- converting value from one data type into another

Built-in functions to change data types

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

In [1]:

```
# calculator to perform addition

first_number = input("Please enter first number : ")
first_number = int(first_number) # converting `str` value into `int`

second_number = input("Please enter second number: ")
second_number = int(second_number) # converting `str` value into `int`

result = first_number + second_number
print(result)
```

```
Please enter first number : 10
Please enter second number: 20
30
```

In [17]:

```
int("prashant")
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call
 last)
Cell In [17], line 1
----> 1 int("prashant")

ValueError: invalid literal for int() with base 10: 'prashant'
```

In [18]:

```
int("1000")
```

Out[18]:

```
1000
```

In [21]:

```
30.50 + 10.50
```

Out[21]:

```
41.0
```

In [22]:

```
30.40 + 12
```

Out[22]:

```
42.4
```

In [24]:

```
int(30.50)

# RULES -

# float can be converted into integer but precision will be lost
# string representing float cannot be converted into int
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call
 last)
Cell In [24], line 1
----> 1 int("30.50")

ValueError: invalid literal for int() with base 10: '30.50'
```

operators

- \- (substraction)
- \+ (addition)
- * (multiplication)
- / (division)
- % (modulo)
- // (floor division) (converts `float` output into `int` output)

In [ ]:

```
print()
input()
int()
float()
str()
bool()
exit()
quit()
help()
type()
```

In [ ]:

```
### Daily class

Mon - Fri

8:30 PM to 10:00 PM


9:00 PM to 10:30 PM  ????
```