

Built in Functions

Built-in functions are developed by creators of Python and are we can use them without defining them. we can just call them using the function name and with argument (if they accept any).

`len()` is built-in function that returns the number of elements present in data structure.

In []:

```
help(len)
```

In []:

```
my_list = ['a', 'f', 1, 3, 4]
len(my_list)
```

`max()` is built-in function that returns the maximum element from the given data structure

In []:

```
help(max)
```

In []:

```
my_list = [1, 2, 3, 4, 5]
max(my_list)
```

`min()` is built-in function that returns the minimum element from the given data structure

In []:

```
my_list = [1, 2, 3, 4, 5]
min(my_list)
```

`sum()` is built-in function that returns the sum of elements in the given data structure

In []:

```
my_list = [1, 2, 3, 4, 5]
sum(my_list)
```

`type()` - returns the type of variable or object

In []:

```
x = 5.6
print("type of x is ", type(x))

x = 4
print("type of x is ", type(x))

x = "string"
print("type of x is ", type(x))
```

sorted() arranges the list of elements in specified order.

In []:

```
my_list = [1, 4, -3, -4, 0, 5 ]
print(sorted(my_list))

my_tuple = (1, 4, -3, -4, 0, 5 )
print(sorted(my_tuple))
```

reversed() returns the list of elements in reversed order

In []:

```
my_list = [1, 4, -3, -4, 0, 5 ]
reversed_list = list(reversed(my_list))
print(reversed_list)

my_tuple = (1, 4, -3, -4, 0, 5 )
reversed_tuple = list(reversed(my_tuple))
print(reversed_tuple)
```

Type conversion functions

Python provides us built-in functions that helps to convert one variable to the other variable type.

For example,

int()

- converts float, Boolean or string into integer

float()

- converts int, Boolean or string into float

str()

- converts float, Boolean or integer into string

bool()

- converts float, string or integer into boolean

In []:

```
int('123')  #converts string '123' to number 123
```

In []:

```
float(123) #converts integer 123 to number 123.0
```

In []:

```
str(123.45) #converts string 123.45 to string "123.45"
```

In []:

```
bool(1)
```

Calling functions from built-in modules

Random number functions

Functions are part of modules as well. Random module is available as part of core Python language. One just need to import random module in order to use the functions defined in it. Once imported, the functions can be called in as usual i.e. with or without passing arguments to it.

In []:

```
import random  
  
help(random)  
  
print(random.random() )
```

In []:

```
print(random.randint(5, 10)) #every time generates the random number between the range i.e  
print(random.randint(5, 10))  
print(random.randint(5, 10))  
print(random.randint(5, 10))
```

In []:

```
for i in range(5):  
    print(random.randint(1, 100))
```

Math functions

Math package is available by default. One can import it in the program and can call the functions defined in it.

In []:

```
import math

print(math.floor(1.4))
print(math.ceil(1.4))
print(math.isnan(2))
print(math.sqrt(5))
```

Exercise

Q1. Explore different built in functions available on string data structure for

- determining whether a string is alphanumeric or not - finding out index of a substring - finding out whether string is identifier or not

In []:

```
#Try it here
```

Q2. Explore following Python built in functions.

- abs
- dir
- eval
- format
- range
- round

In []:

```
#Try it here
```

Q3. Explore the functions in built-in module calendar.

In []:

```
#Try it here
```