

Builtin Functions

In this lecture we will look at functions. Functions are reusable segments of code. Functions can be either builtin functions or user defined. We have already seen many builtin functions. For example:

```
int(), str(), float(), print(), input(), etc.
```

In this notebook, we will take a look at a few more of the built-in functions:

In [4]:

```
'''
The max() function finds the maximum value within the argument provided. The argument can be any
sequence.
However, the elements in the sequence must be either all numeric or all string. A combination of
the two will result
in an error.
'''
print(max('hello world'))
print(max([3,7,9,1,3]))
print(max((5, 8, 10, 15, 23, 1, 9.0)))
print(max(['hello', 'why', 'are']))
print(max((5, 8, 10, 15, 23, 1, 'this'))) # This will result in an error.
```

```
w
9
23
why
```

```
-----
TypeError                                 Traceback (most recent call last)
<ipython-input-4-56f333a3666d> in <module>
      8 print(max((5, 8, 10, 15, 23, 1, 9.0)))
      9 print(max(['hello', 'why', 'are']))
--> 10 print(max((5, 8, 10, 15, 23, 1, 'this'))) # This will result in an error.

TypeError: '>' not supported between instances of 'str' and 'int'
```

In [6]:

```
'''
The sum() returns the total of the elements in a sequence.
'''
print(sum([3,7,9,1,3]))
```

```
23
```

In [7]:

```
'''
The sum() function accepts an iterable object and an optional 'start' argument. This start value
is added to the total.
The default value of start is 0.
'''
print(sum((3,7,9,1,3), 2))
```

```
25
```

In [8]:

```
'''
Note that the input argument of the sum() function must consist of numeric elements. Otherwise yo
u will get an error.
'''
sum(['a','b','c'])
```

```
-----
TypeError                                Traceback (most recent call last)
<ipython-input-8-1c3c6edab232> in <module>
      2 Note that the input argument of the sum() function must consist of numeric elements.
      Otherwise you will get an error.
      3 '''
----> 4 sum(['a','b','c'])

TypeError: unsupported operand type(s) for +: 'int' and 'str'
```

In [9]:

```
'''
round() is another builtin function that we have already seen.
'''
print(round(4578.843, -2))
```

4600.0

In [11]:

```
'''
Functions can be nested one inside another.
For example:
'''
my_str = input('Enter an integer ')
my_int = int(my_str)
my_abs = abs(my_int)
print(my_abs)

'''
can be written in one single statement as follows:
'''
print(abs(int(input('Enter an integer'))))
```

Enter an integer 25
25
Enter an integer25
25

In [12]:

```
'''
The type() builtin function returns the data type of the argument.
'''
print(type('abcdef'))
```

<class 'str'>

One more function is the `range()` function. What does it do?:

In [1]:

```
r = range(5,26,1)
print(list(r))
print(sum(r))
```

[5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25]
315