Lecture 4 - Functions

- Builtin Functions
- User-defined functions, and why use them?
- Writing your first function
- Variables' Scope & functions
- Other functions - Python Standard Library

What is a function?

```
x = input() ##input is a function
print('hello world') ## hello world is a function
type(x) ## type is a function
num = int("1000") ## converts string to integer
num = float("7.12") ## converts string to float
```

Python documentation

- You can find useful bulitin functions to use
- Builtin functions for Python 2: https://docs.python.org/2/library/functions.html (https://docs.python.org/2/library/functions.html)
- Some programming terminology:
  - When we use a function e.g. type(100), we refer to this as **function call** and we refer to input 100 to be **argument**
  - a function can have more than one argument

		Built-in Functions		
abs()	divmod()	input()	open()	staticmethod()
all()	enumerate()	int()	ord()	str()
any()	eval()	isinstance()	pow()	sum()
basestring()	execfile()	issubclass()	print()	super()
bin()	file()	iter()	property()	tuple()
bool()	filter()	len()	range()	type()
bytearray()	float()	list()	raw_input()	unichr()
callable()	format()	locals()	reduce()	unicode()
chr()	frozenset()	long()	reload()	vars()
classmethod()	getattr()	map()	repr()	xrange()
cmp()	globals()	max()	reversed()	zip()
compile()	hasattr()	memoryview()	round()	__import__()
complex()	hash()	min()	set()	
delattr()	help()	next()	setattr()	
dict()	hex()	object()	slice()	
dir()	id()	oct()	sorted()	

In [14]:

```
##useful numeric operations

x = abs(-15)
y = pow(5,2) ##same as 5**2
z = round(12.5)
print 'x:', x, 'y:', y, 'z:',z
```

```
x: 15 y: 25 z: 13.0
```

In [ ]:

```
p
```

In [43]:

```
help(abs)
```

```
Help on built-in function abs in module __builtin__:

abs(...)
    abs(number) -> number

    Return the absolute value of the argument.
```

In [1]:

```
### Useful builtin array functions in python

num_arr = [10,2,3,4]
max_number = max(num_arr)
min_number = min(num_arr)
no_elements = len(num_arr)
total = sum(num_arr)
print 'Array\'s maximum:', max_number, ', minimum:', min_number, ', sum:',total,
 ', No. of elements:', no_elements
```

```
Array's maximum: 10 , minimum: 2 , sum: 19 , No. of elements: 4
```

What is wrong with this code?

In [39]:

```
radius1 = 10
radius2  = 5
pi = 3.14


area1 = pi * radius1 * radius1
area2 = pi * radius2 * radius2
print 'Areas:', area1, area2

circumference1 = 2 * pi * radius1
circumference2 = 2 * pi * radius2
print 'Circumference:', circumference1, circumference2
```

```
Areas: 314.0 78.5
Circumference: 62.8 31.4
```

User-defined functions

- Aside from the builtin functions, you can define your own functions.
- Why is that useful?
  - More structured code => better readability
  - Reusing programs
  - abstraction => allows you to focus on performing one task at a time
- Syntax:

```
def func_name(argument1, argument2,...):
    # function operators
    return output # optional
```

In [45]:

```
pi = 3.14


def calculate_area(radius):
    area = pi * radius * radius
    return area


radius1 = 10
radius2  = 5

area1 = calculate_area(radius1)
area2 =  calculate_area(radius2)

print 'Areas:', area1, area2
```

```
Areas: 314.0 78.5
```

In [44]:

```
def calculate_area(radius):
    ''' number -> number
        Return area of circle given the radius as input.
    '''
    area = pi * radius * radius
    return area


help(calculate_area)
```

```
Help on function calculate_area in module __main__:

calculate_area(radius)
    number -> number
    Return area of circle given the radius as input.
```

Example 1

- In website movie-fans, each user gives a rating to a movie (0-5 stars). The overall rating of movie is considered to be the average of all the ratings from all users.

Part 1

```
- Write a function calculate_rating that calculates the  overall movie rat
ing:
    Example:
        >>> inception_ratings = [4,5,3,5,2,5,5,4,3,5]
        >>> calculate_rating(inception_ratings)
        >>> 4.1
```

Part 2

```
- Write a function get_better_movie that given two movie ratings, returns
 the movie with best overall rating
    Example:
        >>> inception_ratings = [4,5,3,5,2,5,5,4,3,5]
        >>> departed_ratings = [5,5,5,5,5,3,2,4,5,4]
        >>> get_better_movie(inception_ratings, departed_ratings)
        >>> "Second Movie"
```

In [3]:

```
### Part 1
def calculate_rating(movie_ratings):
    overall_rating = sum(movie_ratings)/float(len(movie_ratings))
    return overall_rating

def get_better_movie(rating_list1, rating_list2):
    overall_rating1 = calculate_rating(rating_list1)
    overall_rating2 = calculate_rating(rating_list2)
    if(overall_rating1 == overall_rating2):
        return 'Both have same overall rating'
    elif overall_rating1 > overall_rating2:
        return 'Movie 1'
    else:
        return 'Movie 2'


inception_ratings = [4,5,3,5,2,5,5,4,3,5]
print(calculate_rating(inception_ratings))

departed_ratings = [5,5,5,5,5,3,2,4,5,4]
print(get_better_movie(inception_ratings, departed_ratings))
```

```
4.1
Movie 2
```

Using functions from another file:

Other functions - Python Standard Library

- More functions are available here: https://docs.python.org/2/library/index.html (https://docs.python.org/2/library/index.html)
- Functions that are not built in needs to be imported
- Examples:
  - Math functions : https://docs.python.org/2/library/math.html (https://docs.python.org/2/library/math.html)
  - Web browser : https://docs.python.org/2/library/webbrowser.html (https://docs.python.org/2/library/webbrowser.html)

In [35]:

```
import math
c = math.ceil(1.5)
print(c)
```

```
2.0
```

In [1]:

```
import webbrowser
webbrowser.open("https://www.youtube.com")
```

Out[1]:

True

Variables' scope & functions

```
def dummy_func():
    x = 10
    print("Value inside function:",x)
x = 20
dummy_func()
print("Value outside function:",x)
```

- Variables defined in a function are not visible from outside

Homework

Example 1

- In Twitter, users can write tweets up to 140 characters long. While user is writing a tweet , twitter informs him of no. of characters left, when he exceeds the 140, it starts counting in negative.
- Write a function remaining_chars(tweet) that prints such a number to user
- Example

```
>>> remaining_chars("Good morning twitter")
>>> 120
>>> remaining_chars("")
>>> 140
>>> remaining_chars("I have more respect for a man who lets me kn
ow where he stands, even if he's wrong, than the one who comes up l
ike an angel and is nothing but a devil.")
>>> -11
```

- Hint: function **len** can be used to return no. of characters in a string

```
>>> len("hi")
>>> 2
```