Basics

- Data types
  - Numbers (int, long, float, complex)
  - Strings
  - Lists
  - Dictionaries
  - Booleans
  - Tuples
  - Sets
- Comparison Operators
- if, elif, else Statements
- for Loops
- while Loops
- range()
- list comprehension
- functions
- lambda expressions
- map and filter
- methods

Data types

Numbers

```
In [6]: 1 + 1
```

```
Out[6]: 2
```

```
In [7]: 1 * 3
```

```
Out[7]: 3
```

```
In [8]: 1 / 2
```

```
Out[8]: 0.5
```

```
In [9]: 2 ** 4
```

```
Out[9]: 16
```

```
In [10]: 4 % 2
```

```
Out[10]: 0
```

In [11]:
```
5 % 2
```

Out[11]: 1

In [12]:
```
(2 + 3) * (5 + 5)
```

Out[12]: 50

Variable Assignment

In [13]:
```
# Can not start with number or special characters
name_of_var = 2
```

In [14]:
```
x = 2
y = 3
```

In [15]:
```
z = x + y
```

In [16]:
```
z
```

Out[16]: 5

Strings

In [17]:
```
'single quotes'
```

Out[17]: 'single quotes'

In [18]:
```
"double quotes"
```

Out[18]: 'double quotes'

In [19]:
```
" wrap lot's of other quotes"
```

Out[19]: " wrap lot's of other quotes"

Printing

In [20]:
```
x = 'hello'
```

In [21]:
```
x
```

Out[21]: 'hello'

In [22]:
```
print(x)
```

```
hello
```

In [23]:
```
num = 12
name = 'Sam'
```

In [24]:
```
print('My number is: {one}, and my name is: {two}'.format(one=num,two=name))
```

```
My number is: 12, and my name is: Sam
```

In [25]:
```
print('My number is: {}, and my name is: {}'.format(num,name))
```

```
My number is: 12, and my name is: Sam
```

Lists

In [26]:
```
[1,2,3]
```

Out[26]: [1, 2, 3]

In [27]:
```
['hi',1,[1,2]]
```

Out[27]: ['hi', 1, [1, 2]]

In [28]:
```
my_list = ['a','b','c']
```

In [29]:
```
my_list.append('d')
```

In [30]:
```
my_list
```

Out[30]: ['a', 'b', 'c', 'd']

In [31]:
```
my_list[0]
```

Out[31]: 'a'

In [32]:
```
my_list[1]
```

Out[32]: 'b'

In [33]:
```
my_list[1:]
```

Out[33]: ['b', 'c', 'd']

In [34]:
```
my_list[:1]
```

Out[34]: ['a']

In [35]:
```
my_list[0] = 'NEW'
```

In [98]:
```
my_list
```

Out[98]: ['NEW', 'b', 'c', 'd']

In [99]:
```
nest = [1,2,3,[4,5,['target']]]
```

In [100]:
```
nest[3]
```

Out[100]: [4, 5, ['target']]

In [101]:
```
nest[3][2]
```

Out[101]: ['target']

In [102]:
```
nest[3][2][0]
```

Out[102]: 'target'

Dictionaries

In [37]:
```
d = {'key1':'item1','key2':'item2'}
```

In [38]:
```
d
```

Out[38]: {'key1': 'item1', 'key2': 'item2'}

In [39]:
```
d['key1']
```

Out[39]: 'item1'

Booleans

In [40]:
```
True
```

Out[40]: True

In [41]:
```
False
```

Out[41]: False

Tuples

In [42]:
```
t = (1,2,3)
```

In [43]:
```
t[0]
```

Out[43]: 1

In [44]:
```
t[0] = 'NEW'
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-44-97e4e33b36c2> in <module>()
----> 1 t[0] = 'NEW'

TypeError: 'tuple' object does not support item assignment
```

Sets

In [45]:
```
{1,2,3}
```

Out[45]: {1, 2, 3}

In [46]:
```
{1,2,3,1,2,1,2,3,3,3,3,2,2,2,1,1,2}
```

Out[46]: {1, 2, 3}

Comparison Operators

In [47]:
```
1 > 2
```

Out[47]: False

In [48]:
```
1 < 2
```

Out[48]: True

In [49]:
```
1 >= 1
```

Out[49]: True

In [50]:
```
1 <= 4
```

Out[50]: True

In [51]:
```
1 == 1
```

Out[51]: True

In [52]:
```
'hi' == 'bye'
```

Out[52]: False

Logic Operators

In [53]:
```
(1 > 2) and (2 < 3)
```

Out[53]: False

In [54]:
```
(1 > 2) or (2 < 3)
```

Out[54]: True

In [55]:
```
(1 == 2) or (2 == 3) or (4 == 4)
```

Out[55]: True

if,elif, else Statements

In [56]:
```
if 1 < 2:
    print('Yep!')
```

```
Yep!
```

In [57]:
```
if 1 < 2:
    print('yep!')
```

```
yep!
```

In [58]:
```
if 1 < 2:
    print('first')
else:
    print('last')
```

```
first
```

In [59]:
```
if 1 > 2:
    print('first')
else:
    print('last')
```

```
last
```

In [60]:
```
if 1 == 2:
    print('first')
elif 3 == 3:
    print('middle')
else:
    print('Last')
```

```
middle
```

for Loops

In [61]:
```
seq = [1,2,3,4,5]
```

In [62]:
```
for item in seq:
    print(item)
```

```
1
2
3
4
5
```

In [63]:
```
for item in seq:
    print('Yep')
```

```
Yep
Yep
Yep
Yep
Yep
```

In [64]:
```
for jelly in seq:
    print(jelly+jelly)
```

```
2
4
6
8
10
```

while Loops

In [65]:
```
i = 1
while i < 5:
    print('i is: {}'.format(i))
    i = i+1
```

```
i is: 1
i is: 2
i is: 3
i is: 4
```

range()

In [66]:
```
range(5)
```

Out[66]:
```
range(0, 5)
```

In [67]:
```
for i in range(5):
    print(i)
```

```
0
1
2
3
4
```

In [68]:
```
list(range(5))
```

Out[68]: [0, 1, 2, 3, 4]

list comprehension

In [69]:
```
x = [1,2,3,4]
```

In [70]:
```
out = []
for item in x:
    out.append(item**2)
print(out)
```

```
[1, 4, 9, 16]
```

In [71]:
```
[item**2 for item in x]
```

Out[71]: [1, 4, 9, 16]

functions

In [72]:
```
def my_func(param1='default'):
    """
    Docstring goes here.
    """
    print(param1)
```

In [73]:
```
my_func
```

Out[73]: <function __main__.my_func>

In [74]:
```
my_func()
```

```
default
```

In [75]:
```
my_func('new param')
```

```
new param
```

In [76]:
```
my_func(param1='new param')
```

```
new param
```

In [77]:
```
def square(x):
    return x**2
```

In [78]:
```
out = square(2)
```

In [79]:
```
print(out)
```

```
4
```

lambda expressions

In [80]:
```
def times2(var):
    return var*2
```

In [81]:
```
times2(2)
```

Out[81]:
```
4
```

In [82]:
```
lambda var: var*2
```

Out[82]:
```
<function __main__.<lambda>>
```

map and filter

In [83]:
```
seq = [1,2,3,4,5]
```

In [84]:
```
map(times2,seq)
```

Out[84]:
```
<map at 0x105316748>
```

In [85]:
```
list(map(times2,seq))
```

Out[85]:
```
[2, 4, 6, 8, 10]
```

In [86]:
```
list(map(lambda var: var*2,seq))
```

Out[86]:
```
[2, 4, 6, 8, 10]
```

In [87]:
```
filter(lambda item: item%2 == 0,seq)
```

Out[87]:
```
<filter at 0x105316ac8>
```

In [88]:
```
list(filter(lambda item: item%2 == 0,seq))
```

Out[88]: [2, 4]

methods

In [111]:
```
st = 'hello my name is Sam'
```

In [112]:
```
st.lower()
```

Out[112]: 'hello my name is sam'

In [113]:
```
st.upper()
```

Out[113]: 'HELLO MY NAME IS SAM'

In [103]:
```
st.split()
```

Out[103]: ['hello', 'my', 'name', 'is', 'Sam']

In [104]:
```
tweet = 'Go Sports! #Sports'
```

In [106]:
```
tweet.split('#')
```

Out[106]: ['Go Sports! ', 'Sports']

In [107]:
```
tweet.split('#')[1]
```

Out[107]: 'Sports'

In [92]:
```
d
```

Out[92]: {'key1': 'item1', 'key2': 'item2'}

In [93]:
```
d.keys()
```

Out[93]: dict_keys(['key2', 'key1'])

In [94]:
```
d.items()
```

Out[94]: dict_items([('key2', 'item2'), ('key1', 'item1')])

In [95]:
```
lst = [1,2,3]
```

In [96]:
```
lst.pop()
```

Out[96]: 3

In [108]:
```
lst
```

Out[108]: [1, 2]

In [109]:
```
'x' in [1,2,3]
```

Out[109]: False

In [110]:
```
'x' in ['x','y','z']
```

Out[110]: True

Great Job!