Python Basics - Day 03

Please note, this is not meant to be a comprehensive overview of Python or programming in general.

This notebook is just a code reference for the videos, no written explanations here

This notebook will just go through the basic topics in order:

- Data types
  - Numbers
  - Strings
  - Printing
  - 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 [4]:
```
1 / 2.0
```

Out[4]:
```
0.5
```

In [5]:
```
2 ** 4
```

Out[5]:
```
16
```

In [6]:
```
4 % 2
```

Out[6]:
```
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 [4]:
```
print('hello')
```

```
hello
```

In [5]:
```
print("hello world")
```

```
hello world
```

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

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

In [1]:
```
"double quotes"
```

Out[1]:
```
'double quotes'
```

In [3]:
```
"wrap lot's of other quotes"
```

Out[3]:
```
"wrap lot's of other quotes"
```

Printing

In [47]:
```
x = 'hello'
```

In [48]:
```
x
```

Out[48]:
```
'hello'
```

In [49]:
```
print(x)
```

```
hello
```

In [55]:
```
print('enter name of the lab you are taking: ')
y=input()
print(x+' '+y)
```

```
enter name of the lab you are taking:
AI LAB
hello AI LAB
```

In [13]:
```
age = 31
name = 'amanda'
```

In [14]:
```
print('My name is:{} and my age is:{}'.format(name,age))
```

```
My name is:amanda and my age is:31
```

In [19]:
```
print('My number is: {age}, and my name is: {two}, my friend age: {age}'.format(age=age,
```

```
My number is: 31, and my name is: amanda, my friend age: 31
```

In [23]:
```
print('my friends name is {} and his age is {}'.format(name,age))
```

```
my friends name is amanda and his age is 31
```

In [26]:
```
s='Artificial Intelligence Lab'
```

In [28]:
```
s[0]
```

Out[28]:
```
'A'
```

In [29]:
```
s[:10]
```

Out[29]:
```
'Artificial'
```

In [30]:
```
s[0:]
```

Out[30]:
```
'Artificial Intelligence Lab'
```

In [31]:
```
s[11:]
```

Out[31]:
```
'Intelligence Lab'
```

Lists example

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

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

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

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

In [33]:
```
my_list = ['a','b','c']
```

In [34]:
```
my_list.append('d')
```

In [35]:
```
my_list
```

Out[35]:
```
['a', 'b', 'c', 'd']
```

In [36]:
```
my_list[0]
```

Out[36]:
```
'a'
```

In [37]:
```
my_list[1]
```

Out[37]:
```
'b'
```

In [41]:
```
my_list[1:3]
```

Out[41]:
```
['b', 'c']
```

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 [42]:
```
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 [56]:
```
d = {'key1':'item1','key2':'item2'}
```

In [57]:
```
d
```

Out[57]:
```
{'key1': 'item1', 'key2': 'item2'}
```

In [59]:
```
d['key1']
```

Out[59]:
```
'item1'
```

In [61]:
```
d['key2']
```

Out[61]:
```
'item2'
```

In [86]:
```
d={'k1':[1,2,3,5]}
```

In [87]:
```
d['k1']
```

Out[87]:
```
[1, 2, 3, 5]
```

In [89]:
```
d={'k1':{'innerkey':[1,2,3,4]}}
```

In [92]:
```
d['k1']['innerkey'][2]
```

Out[92]:
```
3
```

In [ ]:
```

```

Booleans

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

Out[40]:
```
True
```

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

Out[41]:
```
False
```

Tuples

- The list is dynamic, whereas the tuple has static characteristics.
- This means that lists can be modified whereas tuples cannot be modified,
- the tuple is faster than the list because of static in nature.

In [81]:
```
t = (1,2,3)
```

In [84]:
```
# We can access a range of items in a tuple by using the slicing operator colon :.
t[:2]
```

Out[84]:
```
(1, 2)
```

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

Out[43]:
```
1
```

In [80]:
```
t[0] = 'NEW'
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[80], line 1
----> 1 t[0] = 'NEW'

NameError: name 't' is not defined
```

Sets

A set is a collection of unique data. That is, elements of a set cannot be duplicate. For example,Suppose we want to store information about student IDs. Since student IDs cannot be duplicate, we can use a set.

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

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

In [62]:
```
{1,2,3,1,2,1,2,3,3,3,3,2,2,2,1,1,2}
```

Out[62]:
```
{1, 2, 3}
```

In [66]:
```
s=set([1,4,4,5,5,9,7,'mia','mia','a','a'])
```

In [67]:
```
s
```

Out[67]:
```
{1, 4, 5, 7, 9, 'a', 'mia'}
```

In [68]:
```
s.add(12)
```

In [69]:
```
s
```

Out[69]:
```
{1, 12, 4, 5, 7, 9, 'a', 'mia'}
```

In [72]:
```
s.add(55)
```

In [73]:
```
s
```

Out[73]:
```
{1, 12, 4, 5, 55, 7, 9, 'a', 'mia'}
```

In [74]:
```
s.add(12)
```

In [78]:
```
# remove set elemet using discard
s.discard(12)
s.discard('mia')
s
```

Out[78]:
```
{1, 4, 5, 55, 7, 9, 'a'}
```

In [79]:
```
len(s)
```

Out[79]:
```
7
```

In [ ]:
```

```

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 [93]:
```
'hi' == 'bye'
```

Out[93]:
```
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 [95]:
```
if 1 == 2:
    print('first')
elif 3 == 3:
    print('middle')
```

```
else:
    print('Last')
```

```
middle
```

In [108…
```
x= int(input())
if (x% 2)==0:
    print(x,' is even number')
else:
    print(x,' is  odd number')
```

```
14
14  is even number
```

In [109…
```
x= int(input())
if (x% 2)==0:
    print('{} is even number'.format(x))
else:
    print('{} is  odd number'.format(x))
```

```
24
24 is even number
```

for Loops

In [4]:
```
for i in range(5):
    print('Element {i} = {i}'.format(i=i+1))
```

```
Element 1 = 1
Element 2 = 2
Element 3 = 3
Element 4 = 4
Element 5 = 5
```

In [5]:
```
seq = [1,2,3,4,5,7,9]
```

In [6]:
```
for item in seq:
    print(item)
```

```
1
2
3
4
5
7
9
```

In [7]:
```
for item in seq:
    print('Yep')
```

```
Yep
Yep
Yep
Yep
Yep
Yep
Yep
```

In [8]:
```
for jelly in seq:
    print(jelly+jelly)
```

```
2
4
6
8
```

```
10
14
18
```

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 [34]:
```
def my_func(lab='default'):
    """
    Docstring goes here.
    multiline commnet.
    will make square of any value
```

```
    """
    print('hello '+lab)
```

In [35]:
```
my_func
```

Out[35]:
```
<function __main__.my_func(lab='default')>
```

In [36]:
```
my_func()
```

```
hello default
```

In [37]:
```
my_func('Artificial Intelligence Lab')
```

```
hello Artificial Intelligence Lab
```

In [38]:
```
my_func(lab='new param')
```

```
hello new param
```

In [39]:
```
def square(x):
    return x**2
```

In [40]:
```
output = square(3)
```

In [43]:
```
print(output)
```

```
9
```

In [46]:
```
square(4)
```

Out[46]:
```
16
```

lambda expressions

In [49]:
```
def times2(var):
    return var*2
```

In [50]:
```
times2(2)
```

Out[50]:
```
4
```

In [51]:
```
lambda var: var*2
```

Out[51]:
```
<function __main__.<lambda>(var)>
```

map and filter

In [52]:
```
seq = [1,2,3,4,5]
```

In [53]:
```
map(times2,seq)
```

Out[53]:
```
<map at 0x1c9aa014a00>
```

In [54]:
```
list(map(times2,seq))
```

Out[54]:
```
[2, 4, 6, 8, 10]
```

In [55]:
```
list(map(lambda var: var*2,seq))
```

Out[55]:
```
[2, 4, 6, 8, 10]
```

In [56]:
```
filter(lambda item: item%2 == 0,seq)
```

Out[56]:
```
<filter at 0x1c9aa02e890>
```

In [57]:
```
list(filter(lambda item: item%2 == 1,seq))
```

Out[57]:
```
[1, 3, 5]
```

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 [58]:
```
'x' in ['x','y','z']
```

Out[58]:
```
True
```

tuple unpacking

In [59]:
```
x=[(1,2),(3,4),(5,6)]
```

In [60]:
```
x
```

Out[60]:
```
[(1, 2), (3, 4), (5, 6)]
```

In [61]:
```
for item in x:
    print(item)
```

```
(1, 2)
(3, 4)
(5, 6)
```

In [64]:
```
for a,b in x:
    print(a)
```

```
1
3
5
```

Great Job!