Python_Course_1

April 20, 2024

0.1 Data types

0.1.1 Numbers

[1]:
```
1 + 1
```

[1]:
```
2
```

[2]:
```
1 * 3
```

[2]:
```
3
```

[3]:
```
1 / 2
```

[3]:
```
0.5
```

[4]:
```
2 ** 4
```

[4]:
```
16
```

[5]:
```
4 % 2
```

[5]:
```
0
```

[6]:
```
5 % 2
```

[6]:
```
1
```

[7]:
```
(2 + 3) * (5 + 5)
```

[7]:
```
50
```

0.1.2 Variable Assignment

[8]:
```
# Can not start with number or special characters
name_of_var = 2
```

[9]:
```
x = 2
y = 3
```

[10]:
```
z = x + y
```

[11]:
```
z
```

[11]: 5

0.1.3 Strings

[12]:
```
'single quotes'
```

[12]: 'single quotes'

[13]:
```
"double quotes"
```

[13]: 'double quotes'

[14]:
```
" wrap lot's of other quotes"
```

[14]: " wrap lot's of other quotes"

0.1.4 Printing

[17]:
```
x = 'hello world'
```

[18]:
```
x
```

[18]: 'hello world'

[19]:
```
print(x)
```

```
hello world
```

[20]:
```
num = 12
name = 'Sam'
```

[21]:
```
print('My number is: {one}, and my name is: {two}'.format(one=num,two=name))
```

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

[22]:
```
print('My number is: {}, and my name is: {}'.format(num,name))
```

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

0.1.5 Lists

[23]:
```
[1,2,3]
```

[23]:
```
[1, 2, 3]
```

[24]:
```
['hi',1,[1,2]]
```

[24]:
```
['hi', 1, [1, 2]]
```

[25]:
```
my_list = ['a','b','c']
```

[26]:
```
my_list.append('d')
```

[27]:
```
my_list
```

[27]:
```
['a', 'b', 'c', 'd']
```

[28]:
```
my_list[0]
```

[28]:
```
'a'
```

[29]:
```
my_list[1]
```

[29]:
```
'b'
```

[30]:
```
my_list[1:]
```

[30]:
```
['b', 'c', 'd']
```

[31]:
```
my_list[:1]
```

[31]:
```
['a']
```

[32]:
```
my_list[0] = 'NEW'
```

[33]:
```
my_list
```

[33]:
```
['NEW', 'b', 'c', 'd']
```

[34]:
```
nest = [1,2,3,[4,5,['target']]]
```

[35]:
```
nest[3]
```

[35]:
```
[4, 5, ['target']]
```

[36]:
```
nest[3][2]
```

[36]:
```
['target']
```

[37]:
```
nest[3][2][0]
```

[37]:
```
'target'
```

0.1.6 Dictionaries

[38]:
```
d = {'key1':'item1','key2':'item2'}
```

[39]:
```
d
```

[39]:
```
{'key1': 'item1', 'key2': 'item2'}
```

[40]:
```
d['key1']
```

[40]:
```
'item1'
```

0.1.7 Booleans

[41]:
```
True
```

[41]:
```
True
```

[42]:
```
False
```

[42]:
```
False
```

0.1.8 Tuples

[43]:
```
t = (1,2,3)
```

[44]:
```
t[0]
```

[44]:
```
1
```

[45]:
```
t[0] = 'NEW'
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[45], line 1
----> 1 t[0] = 'NEW'

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

0.1.9 Sets

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

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

[47]:
```
{1,2,3,1,2,1,2,3,3,3,3,2,2,2,1,1,2}
```

[47]:
```
{1, 2, 3}
```

0.2 Comparison Operators

[48]:
```
1 > 2
```

[48]:
```
False
```

[49]:
```
1 < 2
```

[49]:
```
True
```

[50]:
```
1 >= 1
```

[50]:
```
True
```

[51]:
```
1 <= 4
```

[51]:
```
True
```

[52]:
```
1 == 1
```

[52]:
```
True
```

[53]:
```
'hi' == 'bye'
```

[53]:
```
False
```

0.3 Logic Operators

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

[54]:
```
False
```

[55]:
```
(1 > 2) or (2 < 3)
```

[55]:
```
True
```

[56]:
```
(1 == 2) or (2 == 3) or (4 == 4)
```

[56]: True

0.4 if,elif, else Statements

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

Yep!

[58]:
```
if 1 < 2:
    print('yep!')
```

yep!

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

first

[60]:
```
if 1 > 2:
    print('first')
else:
    print('last')
```

last

[61]:
```
if 1 == 2:
    print('first')
elif 3 == 3:
    print('middle')
else:
    print('Last')
```

middle

0.5 for Loops

[62]:
```
seq = [1,2,3,4,5]
```

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

1
2
3

```
4
5
```

[64]:
```
for item in seq:
    print('Yep')
```

```
Yep
Yep
Yep
Yep
Yep
```

[65]:
```
for jelly in seq:
    print(jelly+jelly)
```

```
2
4
6
8
10
```

0.6 while Loops

[66]:
```
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
```

0.7 range()

[67]:
```
range(5)
```

[67]:
```
range(0, 5)
```

[68]:
```
for i in range(5):
    print(i)
```

```
0
1
2
3
4
```

[69]:
```
list(range(5))
```

[69]:
```
[0, 1, 2, 3, 4]
```

0.8 list comprehension

[70]:
```
x = [1,2,3,4]
```

[71]:
```
out = []
for item in x:
    out.append(item**2)
print(out)
```

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

[72]:
```
[item**2 for item in x]
```

[72]:
```
[1, 4, 9, 16]
```

0.9 functions

[73]:
```
def my_func(param1='default'):
    """
    Docstring goes here.
    """
    print(param1)
```

[74]:
```
my_func
```

[74]:
```
<function __main__.my_func(param1='default')>
```

[75]:
```
my_func()
```

```
default
```

[76]:
```
my_func('new param')
```

```
new param
```

[77]:
```
my_func(param1='new param')
```

```
new param
```

[78]:
```
def square(x):
    return x**2
```

[79]:
```
out = square(2)
```

[80]:
```
print(out)
```

```
4
```

0.10 lambda expressions

[81]:
```
def times2(var):
    return var*2
```

[82]:
```
times2(2)
```

[82]:
```
4
```

[83]:
```
lambda var: var*2
```

[83]:
```
<function __main__.<lambda>(var)>
```

0.11 map and filter

[84]:
```
seq = [1,2,3,4,5]
```

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

[85]:
```
<map at 0x7b27f87be050>
```

[86]:
```
list(map(times2,seq))
```

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

[87]:
```
list(map(lambda var: var*2,seq))
```

[87]:
```
[2, 4, 6, 8, 10]
```

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

[88]:
```
<filter at 0x7b28018e6bc0>
```

[89]:
```
list(filter(lambda item: item%2 == 0,seq))
```

[89]:
```
[2, 4]
```

0.12 methods

[90]:
```
st = 'hello my name is Sam'
```

[91]:
```
st.lower()
```

[91]: 'hello my name is sam'

[92]:
```
st.upper()
```

[92]: 'HELLO MY NAME IS SAM'

[93]:
```
st.split()
```

[93]: ['hello', 'my', 'name', 'is', 'Sam']

[94]:
```
tweet = 'Go Sports! #Sports'
```

[95]:
```
tweet.split('#')
```

[95]: ['Go Sports! ', 'Sports']

[96]:
```
tweet.split('#')[1]
```

[96]: 'Sports'

[97]:
```
d
```

[97]: {'key1': 'item1', 'key2': 'item2'}

[98]:
```
d.keys()
```

[98]: dict_keys(['key1', 'key2'])

[99]:
```
d.items()
```

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

[100]:
```
lst = [1,2,3]
```

[101]:
```
lst.pop()
```

[101]: 3

[102]:
```
lst
```

[102]: [1, 2]

[103]:
```
'x' in [1,2,3]
```

[103]: False

[104]:
```
'x' in ['x','y','z']
```

[104]: True

[ ]:

[ ]: