day3

February 5, 2023

0.1 logical operator

#1.And #2.or #3.not

[3]:
```
not True
```

[3]:
```
False
```

[4]:
```
bool(0)
```

[4]:
```
False
```

[5]:
```
not(bool(0))
```

[5]:
```
True
```

[7]:
```
 int(bool(0))
```

[7]:
```
0
```

[8]:
```
 int(not(bool(0)))
```

[8]:
```
1
```

[9]:
```
not 2
```

[9]:
```
False
```

[10]:
```
not 0
```

[10]:
```
True
```

[11]:
```
not -1
```

[11]:
```
False
```

[12]:
```
not 1
```

[12]:
```
False
```

[13]:
```
zero=0
one=1

print(f"boolean value of no.{zero} is {bool(zero)}")
print(f"boolean value of no.{one} is {bool(one)}")
print("\n#-------------------#\n")
```

```
boolean value of no.0 is False
boolean value of no.1 is True

#-------------------#
```

[15]:
```
bool(1)
```

[15]: True

[16]:
```
bool(-5)
```

[16]: True

0.2 logical and

[20]:
```
VEGITABLES = True
SALT =False
DISH = VEGITABLES and SALT

print(f"Dish contain VEGITABLES:{VEGITABLES}")
print(f"Dish contain SALT:{SALT}")
print(f"Hence dish prepared was good:{DISH}\n")
```

```
Dish contain VEGITABLES:True
Dish contain SALT:False
Hence dish prepared was good:False
```

[26]:
```
not(False)*False == True
```

[26]: True

[27]:
```
not(False)*False
```

[27]: True

[29]:
```
print((not(False))*False)
```

```
0
```

[30]:
```
False * True
```

[30]: 0

[32]:
```
not(False * True)
```

[32]: True

[33]:
```
lst=[1,2,3,4,5]
lst1=[1,2,3,4]
lst==lst1
```

[33]: False

[3]:
```
lst=[1,2,3,4,5]
lst1=[1,2,3,4]
lst!=lst1
```

[3]: True

[36]:
```
id(lst)
```

[36]: 140474058834560

[4]:
```
lst=[1,2,3,4,5]
lst1=[1,2,3,4]
lst1=lst
print(id(lst))
print(id(lst1))
```

```
140388657815872
140388657815872
```

[5]:
```
lst
```

[5]: [1, 2, 3, 4, 5]

[6]:
```
lst[0]
```

[6]: 1

[7]:
```
lst[0]=10
```

[8]:
```
lst
```

[8]: [10, 2, 3, 4, 5]

[11]:
```
maximum_speed_of_bike=150
minimum_speed_of_bike=80
print(f"bike speed is {maximum_speed_of_bike<=minimum_speed_of_bike}")
```

```
bike speed is False
```

[12]:
```
6%2
```

[12]:
```
0
```

[13]:
```
var=10
bin(var)
```

[13]:
```
'0b1010'
```

[1]:
```
str1='kishor'
```

[2]:
```
str1
```

[2]:
```
'kishor'
```

[5]:
```
string="pw data science course"
```

[6]:
```
string[5]
```

[6]:
```
't'
```

[10]:
```
string[-1]
```

[10]:
```
'e'
```

[11]:
```
print(string[::-1])
```

```
esruoc ecneics atad wp
```

[12]:
```
str1.count
```

[12]:
```
<function str.count>
```

[14]:
```
string[:-1]
```

[14]:
```
'pw data science cours'
```

[15]:
```
string[:]
```

[15]:
```
'pw data science course'
```

[16]:
```
string[::-3]
```

[16]:
```
'eu ncadp'
```

[17]:
```
course_name="data science masters"
```

[18]:
```
course_name[::-1]
```

[18]:
```
'sretsam ecneics atad'
```

[20]:
```
course_name[5:12]
```

[20]:
```
'science'
```

[21]:
```
course_name[11:4:-1]
```

[21]:
```
'ecneics'
```

[23]:
```
course_name.count
```

[23]:
```
<function str.count>
```

[34]:
```
course_name*5
```

[34]:
```
'data science mastersdata science mastersdata science mastersdata science
mastersdata science masters'
```

[25]:
```
len(course_name)
```

[25]:
```
20
```

[28]:
```
course_name.find("s")
```

[28]:
```
5
```

[29]:
```
course_name.find("a")
```

[29]:
```
1
```

[31]:
```
course_name.find("s",2,5)
```

[31]:
```
-1
```

[35]:
```
## count Function()
course_name.count("s")
```

[35]:
```
3
```

[36]:
```
course_name.find(" ")
```

[36]:
```
4
```

[37]:
```
course_name.count("a",2,10)
```

[37]:
```
1
```

[38]:
```
# string split Function
```

[40]:
```
course_name.split(' ')
```

[40]:
```
['data', 'science', 'masters']
```

[41]:
```
course_name.split('s')
```

[41]:
```
['data ', 'cience ma', 'ter', '']
```

[ ]:
```
course_name.upper()
```

[ ]:
```

```