Topics

1. Operators
    - Arithmetic
    - Relational
    - Logical
    - Special
        - Identity
        - Membership
2. Conditional Statements
3. Strings

    - String Slicing
    - String Methods

Relational operators

- Greaterthan
- Lessthan
- Greaterthan or equal
- Lessthan or equal
- Equal
- not equal

In [1]:
```python
a = 50
b = 45
print(a>b)
print(a<b)
print(a>=b)
print(a<=b)
print(a==b)
print(a!=b)
```

```
True
False
True
False
False
True
```

Logical operators

- and
- or
- not

In [3]:
```
a = True
b = False
print(a or b) # Either one must be true
print(a and b) # Both must be true
print( not b) # Negation of the given value
```

```
True
False
True
```

Special Operators

- Identity
- Membership

Identity

- is # Check both value and same object
- is not

In [21]:
```
a ="Aditya#College"
b = "Aditya#College"
print(a is b)  # gives True without space only
print(a is not b)
print(id(a),id(b))
```

```
False
True
63810120 63851880
```

In [23]:
```
a = 1
b = 1
print(a is b)
print(id(a),id(b))
```

```
True
1613547680 1613547680
```

Membership

- in
- not in

In [34]:
```python
s ="Aditya#College"
b = "Aditya#College"
print("Aditya" in s)
print("A"  not in b)
```

```
True
False
```

Bitwise Operators

Bitwise and (&)

Bitwise or (|)

Bitwise not (~)

Bitwise xor (^)

bitwise shift left <<

bitwise shift right >>

In [ ]:
```python
l = 1
```

In [44]:
```python
2 and 3
```

Out[44]:
```
3
```

In [46]:
```python
a = 2
b = 3
print(a & b)
print(a | b)
print(a<<2)
print(a>>1)

```

```
2
3
8
1
```

Conditional Statements

- if
- else
- elif
- Nested if

In [5]:
```
## if Condition

a1 = "Aditya College of Engineering and Technology"
b1 = "Aditya  of Engineering and Technology"
if (a1==b1):
    print("Both are same")
else:
    print("both are different")


```
```
both are different
```

In [7]:
```
a = 35
b = 25
if (a>b):
    print("a is greaterthan b")
elif (a<b):
    print("a is lessthan b")
else:
    print("Both are equal")
```
```
a is greaterthan b
```

In [12]:
```
a = 3
if (a>0):
    if (a ==2):
        print("a value is 2")
    else:
        print("a is not equal to 2")
else:
    print("Given value is lessthan zero")
```
```
a is not equal to 2
```

Strings

Collection of Charecters

In [44]:
```
s = "Aditya College"
s[::3] # Accessing the charecters with a difference of 3

```
Out[44]:
```
'At lg'
```

In [45]:
```
s[-1::-1]  # printing the string in revese order
print(s[len(s)//2])  # Accessing the middle charecter
len(s)//2  # Getting the index of middle charecter
```
```
C
```
Out[45]:
```
7
```

```
In [46]: s[1:-1]  # printing the string except 1st and last charecters
Out[46]: 'ditya Colleg'
```

In [47]:
```
dir(str)
```

Out[47]:
```
['__add__',
 '__class__',
 '__contains__',
 '__delattr__',
 '__dir__',
 '__doc__',
 '__eq__',
 '__format__',
 '__ge__',
 '__getattribute__',
 '__getitem__',
 '__getnewargs__',
 '__gt__',
 '__hash__',
 '__init__',
 '__init_subclass__',
 '__iter__',
 '__le__',
 '__len__',
 '__lt__',
 '__mod__',
 '__mul__',
 '__ne__',
 '__new__',
 '__reduce__',
 '__reduce_ex__',
 '__repr__',
 '__rmod__',
 '__rmul__',
 '__setattr__',
 '__sizeof__',
 '__str__',
 '__subclasshook__',
 'capitalize',
 'casefold',
 'center',
 'count',
 'encode',
 'endswith',
 'expandtabs',
 'find',
 'format',
 'format_map',
 'index',
 'isalnum',
 'isalpha',
 'isascii',
 'isdecimal',
 'isdigit',
 'isidentifier',
 'islower',
 'isnumeric',
 'isprintable',
 'isspace',
 'istitle',
 'isupper',
 'join',
 'ljust',
 'lower',
 'lstrip',
 'maketrans',
 'partition',
 'replace',
 'rfind',
 'rindex',
 'rjust',
 'rpartition',
 'rsplit',
 'rstrip',
 'split',
 'splitlines',
 'startswith',
 'strip',
 'swapcase',
 'title',
 'translate',
 'upper',
 'zfill']
```

In [58]:
```python
s = "Aditya college"
s.capitalize()
s.upper()
s.swapcase()
s.zfill(20)
```

Out[58]:
```
'000000Aditya college'
```

In [65]:
```python
s = "Adi@tya@college"
a = s.split("@")  # splits based on the given parameter
```

In [66]:
```python
"#".join(a)  # join the parameter/symbol to a string
```

Out[66]:
```
'Adi#tya#college'
```

In [67]:
```python
s1 = "     Aditya college   "
r = s1.strip()   # Remove the spaces
```

Out[67]:
```
'Aditya college'
```

In [76]:
```python
 s1.lstrip()

```

Out[76]:
```
'Aditya college   '
```

In [69]:
```python
s1.rstrip()
```

Out[69]:
```
'     Aditya college'
```

In [73]:
```
s1.count("Aditya") # to count how many times a perticuliar charecter is repea
```
Out[73]:
```
1
```

In [75]:
```
help(str.replace)
# To know the syntax or parameters will be given in that method
```
```
Help on method_descriptor:

replace(self, old, new, count=-1, /)
    Return a copy with all occurrences of substring old replaced by new.
    
      count
        Maximum number of occurrences to replace.
        -1 (the default value) means replace all occurrences.
    
    If the optional argument count is given, only the first count occurrences a
re
    replaced.
```

In [78]:
```
r.replace("a","s")  # replace("old","New")
```
Out[78]:
```
'Aditys college   '
```

In [ ]:
```

```

In [ ]:
```

```

Loops in python

```
* for loop
* while loop
```

for temp_var in collection_item: body_loop

In [79]:
```
name = 'python'
for each in name:
    print(each)
```
```
p
y
t
h
o
n
```

In [83]:
```
for number in range(20,10,-1):
    print(number)

```

...

In [85]:
```
name = 'python'
for char in name:
    if char=='z':
        break
print(char)
```

```
n
```

In [88]:
```
for number in range(10):
    print(number,end=' ')
```

```
0 1 2 3 4 5 6 7 8 9
```

write program to sum of even numbers from 0 to 100 (both inclusive)

In [90]:
```
total =0
for number in range(0,101):
    if number%2==0:
        total += number
print(total)
```

```
2550
```

note:

never use `sum` as variable in python in python has `sum()` function

In [94]:
```
for each in range(3):
    print(each)
else:
    print('loop executed successfully')
```

```
0
1
2
loop executed successfully
```

```
while condition:
    body_of_loop
```

print 10 to 20 using while loop

In [96]:
```
start = 10
while start<21:
    print(start)
    start += 1

```

...

take the data from user untill user give DONE ,print only numbers if user give any numbers

```
input
------
    10
    SAI
    20
    DONE


OUTPUT
    10
    20
```

In [98]:
```
s = '123'
s.isdigit()
```

Out[98]:
```
True
```

In [100]:
```
while True:
    data = input('')
    if data=='DONE':
        break
    if data.isdigit():
        print(data)

```

```
10
10
SAI
30
30
DONE
```

Functions in python

```
1) Pre-defined/system defined functions
2) user defined functions
3) Ananymous function
```

In [ ]:
```
range()
print()
len()
sum()

```

user defined function

```
def fun_name(arg1,arg2):
    body_of_function
```

In [102]:
```
def message(msg):
    print(msg)
```

In [103]:
```
message('welcome')
```

```
welcome
```

In [104]:
```
a = message('welcome everyone')
```

```
welcome everyone
```

In [105]:
```
print(a)
```

```
None
```

In [106]:
```
def addition(a,b):
    return a+b
```

In [107]:
```
a = addition(10,20)
print(a)
```

```
30
```

In [109]:
```
a = addition(10,10,30)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-109-a0e8a68c9b5e> in <module>
----> 1 a = addition(10,10,30)

TypeError: addition() takes 2 positional arguments but 3 were given
```

based on arguments

```
1) Required argument function
2) keyword argument function
3) Default argument function
4) Variable length argument function
```

Required arg function

In this type we must pass required arguments in the fuction structure

In [110]:
```
addition(10,20)
```
Out[110]:
```
30
```

Keyword arg function

In [111]:
```
addition(a=10,b=20)
```
Out[111]:
```
30
```

Default arg function

In [112]:
```
def addition(a,b=0):
    print('a',a)
    print('b',b)
    return a+b



```

In [113]:
```
addition(10)
```
```
a 10
b 0
```
Out[113]:
```
10
```

In [115]:
```
addition(10,20)
```
```
a 10
b 20
```
Out[115]:
```
30
```

variable length arg functions

In [116]:
```
def addition(a,b,*var):
    print('a',a)
    print('b',b)
    print(var)

```

In [120]:
```
def myPrint(*var):
    print(var)
```

In [121]:
```
myPrint(1,2,3,4)
```

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

In [119]:
```
addition(1,2,3,4,5)
```

```
a 1
b 2
(3, 4, 5)
```

Type *Markdown* and LaTeX: α^2

create function to check given number prime or not isPrime()

In [130]:
```
def isPrime(number):
    if number>1:
        flag = True
        for each in range(2,number):
            if number%each == 0:
                flag = False
                break
        return flag
    else:
        return False
```

find the sum of prime numbers from 10 to 100

In [131]:
```
total = 0
for number in range(10,101):
    if isPrime(number):
        total += number
print(total)

```

```
1043
```

Anonymous function in python

```
For anonymous function created using lambda keyword.
It is single line function
it can not use(not required) return,but directly
return evaluated expression

    lambda arguments:expression
```

In [133]:
```
addition = lambda a,b:a+b
```

```
In [134]: addition(10,20)
Out[134]: 30
```

```
In [140]: def addition(a,b):
              return a,b
```

```
In [141]: a = addition(10,20)
          print(a,type(a))

(10, 20) <class 'tuple'>
```

```
In [144]: addition(20,a=30)
...
```

```
In [146]: print(1,2,3,4,sep='\n')

1
2
3
4
```

```
In [147]: 2 and 3
Out[147]: 3
```

```
In [149]: 0 and 4
Out[149]: 0
```

```
In [ ]:
```