Assignment 10 - Python

Print() Function

In [1]:
```
a=10
b=20
a
b
```

Out[1]: 20

In [2]:
```
a=10
b=20
print(a)
print(b)
```

```
10
20
```

In [3]:
```
print(10)
print(10,20)
print('python')
print(10,20,'python')
```

```
10
10 20
python
10 20 python
```

In [4]:
```
num1=20
num2=30
add=num1+num2
print(add)
```

```
50
```

print result with string

num1=20 num2=30 add=num1+num2 print('The addition of',num1,'and',num2,'is=',add)

In [7]:
```
name='Python'
age=20
city='hyd'
#hellow my name is python and i am 10 year old from hydrabad

print('My name is',name,'and i am',age,'years old form',city)
```

```
My name is Python and i am 20 years old form hyd
```

print Format method

In [8]:
```
num1=20
num2=30
add=num1+num2
print('The addition of {} and {} is= {}'.format(num1,num2,add))
```

```
The addition of 20 and 30 is= 50
```

first deside how the print statement should be like:- The addition of 20 and 30 is = 50 then replace the variable position with curly brtacesm { } then appply .format(val1,val2,....val-n methodm

In [9]:
```
name='Python'
age=20
city='hyd'
#hellow my name is python and i am 10 year old from hydrabad

print('hello my name is {}, and i am {} years old from {} '.format(name,age,city))
```

```
hello my name is Python, and i am 20 years old from hyd
```

In [10]:
```
num1=100
num2=25
num3=333
avg=(num1+num2+num3)/3 # or we can use avg=round(num1+num2+num3)/3,2)
avg1=round((num1+num2+num3)/3,2)
# The avrage of num1,num2,num3 is = avg
print('The avrage of {}, {}, and {} is= {} or {}'.format(num1,num2,num3, avg,avg1))
```

```
The avrage of 100, 25, and 333 is= 152.66666666666666 or 152.67
```

In [11]:
```
round(avg,2) # round of till 2 digite after decimal
```

Out[11]:
```
152.67
```

More short format meythod(f string method) variable should be in curly braces and write everything inside quots " at starting simpaly add f

In [12]:
```
num1=20
num2=30
add=num1+num2
print(f'The addition of {num1} and {num2} is= {add}') # alwase prefer this
```

```
The addition of 20 and 30 is= 50
```

name='Python' age=20 city='hyd' #hellow my name is python and i am 10 year old from hydrabad print(f'hello my name is {name}, and i am {age} year old, from {city}.')

In [15]:
```
num1=100
num2=25
num3=333
avg=round((num1+num2+num3)/3,2) # or we can use avg=round(num1+num2+num3)/3,2) # Th

print(f'The avrage of {num1}, {num2} and {num3} is = {avg}')
```

```
The avrage of 100, 25 and 333 is = 152.67
```

In [17]:
```
# lete combine all
num1=10
num2=20
add = num1+ num2
print('The addition of',num1,'and',num2,'is=',add)
print('The addition of {} and {} is= {}'.format(num1,num2,add))
print(f'The addition of {num1} and {num2} is= {add}')
```

```
The addition of 10 and 20 is= 30
The addition of 10 and 20 is= 30
The addition of 10 and 20 is= 30
```

end statement

In [18]:
```
print('hello') # 1st statement
print('good moorning') # 2nd statement)
# i want print like:- hellow good morning
```

```
hello
good moorning
```

Here we will use end statement that joint line from end of one string to starting of other string

In [19]:
```
print('hello', end=' ') # 1st statement
print('world good day') # 2nd statement
```

```
hello world good day
```

seprator

- here one print statement only we use
- insisde one print statement we have multipal value s - we want to seperate these multipal values with anything

In [20]:
```
print('hello','hai','how are you',sep='--->')
```

```
hello--->hai--->how are you
```

In [21]:
```
print('hello','hai','how are you',sep='&')
```

```
hello&hai&how are you
```

In [22]:
```
print('hello','hai','how are you',sep='@')
```

```
hello@hai@how are you
```

In [23]:
```
print('hello','hai','how are you',sep=' ')
```

```
hello hai how are you
```

In [24]:
```
print(3,'.') # . is far from 3 so here we will use sep method
```

```
3 .
```

In [25]:
```
print(3,'.',sep='') # see now space setteld(also use to remove space B/W words)
```

```
3.
```

In [26]:
```
print(1,2,end=' ')
print(3,'.',sep='') # will print 1 2 3.
```

```
1 2 3.
```

String() Function

Escape characters

- An escape character is created by typing a backslash ` \ followed by the character youwant to insert.

In [27]:
```
print("Hello there!\nHow are you?\nI\'m doing fine.")
```

```
Hello there!
How are you?
I'm doing fine.
```

Raw strings

- A raw string entirely ignores all escape characters and prints any backslash that appearsin the string
- Raw strings are mostly used for regular expressio definition.

In [28]:
```
print(r"Hello there!\nHow are you?\nI\'m doing fine.")
```

```
Hello there!\nHow are you?\nI\'m doing fine.
```

Multiline Strings

In [29]:
```
print(
...."""Dear Alice,
...
... Eve's cat has been arrested for catnapping,
... cat burglary, and extortion.
...
... Sincerely,
... Bob"""
...)
```

```
  Cell In[29], line 2
    ...."""Dear Alice,
    ^
SyntaxError: invalid syntax. Perhaps you forgot a comma?
```

Indexing and Slicing strings

Indexing

In [31]:
```
spam = 'Hello world!'
print(spam[0])
```

```
H
```

In [32]:
```
spam[4]
```

Out[32]: 'o'

In [33]:
```
spam[-1]
```

Out[33]: '!'

Slicing

In [34]:
```
spam = 'Hello world!'
spam[0:5]
```

Out[34]: 'Hello'

In [35]:
```
spam[:5]
```

Out[35]: 'Hello'

In [36]:
```
spam[6:-1]
```

Out[36]: 'world'

In [37]:
```
spam[:-1]
```

Out[37]: 'Hello world'

In [38]:
```
spam[::-1]
```

Out[38]: '!dlrow olleH'

In [41]:
```
fizz = spam[0:5]
fizz
```

Out[41]: 'Hello'

The in and not in operators

In [43]:
```
'Hello' in 'Hello World'
```

Out[43]: True

In [44]:
```
'HELLO' in 'Hello World'
```

Out[44]: False

upper(), lower() and title()

In [45]:
```
greet = 'Hello world!'
greet.upper()
```

Out[45]: 'HELLO WORLD!'

In [46]:
```
greet.lower()
```

Out[46]: 'hello world!'

In [47]:
```
greet.title()
```

Out[47]: 'Hello World!'

isupper() and islower() methods

-Returns `True` or `False` after evaluating if a string is in upper or lower case:

In [49]:
```
spam = 'Hello world!'
spam.islower()
```

Out[49]: False

In [50]:
```
spam.isupper()
```

Out[50]: False

In [51]:
```
'HELLO'.isupper()
```

Out[51]: True

In [52]:
```
'abc12345'.islower()
```

Out[52]: True

startswith() and endswith()

In [53]:
```
'Hello world!'.startswith('Hello')
```

Out[53]: True

In [54]:
```
'Hello world!'.endswith('world')
```

Out[54]: False

In [55]:
```
'abc123'.startswith('abcdef')
```

Out[55]: False

In [56]:
```
'abc123'.endswith('12')
```

Out[56]: False

join() and split()

join()

- The 'join()' method takes all the items in an iterable, like a list, dictionary, tuple or set and joins them into a string.

In [57]:
```
''.join(['My','name','is','Simon'])
```

Out[57]: 'MynameisSimon'

In [58]:
```
','.join(['My','name','is','Simon'])
```

Out[58]: 'My,name,is,Simon'

In [59]:
```
'ABC'.join(['My','name','is','Simon'])
```

Out[59]: 'MyABCnameABCisABCSimon'

split()

- The 'split()' method splits 'string' into a 'list'.
- By default, it will use whitespace to separate the items, but you can also set another character of choice:

In [60]:
```
'My name is Simon'.split()
```

Out[60]: ['My', 'name', 'is', 'Simon']

In [61]:
```
'MyABCnameABCisABCSimon'.split('ABC')
```

Out[61]: ['My', 'name', 'is', 'Simon']

In [63]:
```
'My name is Simon'.split('m')
```

Out[63]:
```
['My na', 'e is Si', 'on']
```

In [64]:
```
'My name is Simon'.split()
```

Out[64]:
```
['My', 'name', 'is', 'Simon']
```

In [66]:
```
'My name is Simon'.split(' ')
```

Out[66]:
```
['My', 'name', 'is', 'Simon']
```

Justifying text with rjust(), ljust() and center()

In [67]:
```
'Hello'.rjust(10)
```

Out[67]:
```
'     Hello'
```

In [68]:
```
'Hello'.rjust(20)
```

Out[68]:
```
'               Hello'
```

In [70]:
```
'Hello World'.rjust(20)
```

Out[70]:
```
'         Hello World'
```

In [71]:
```
'Hello'.ljust(10)
```

Out[71]:
```
'Hello     '
```

In [72]:
```
'Hello'.center(20)
```

Out[72]:
```
'       Hello        '
```

In [73]:
```
'Hello'.rjust(20, '*')
```

Out[73]:
```
'***************Hello'
```

In [74]:
```
'Hello'.ljust(20, '-')
```

Out[74]:
```
'Hello---------------'
```

In [75]:
```
'Hello'.center(20, "=")
```

Out[75]:
```
'=======Hello========'
```

Removing whitespace with strip(), rstrip(), and lstrip()

In [76]:
```
spam = ' Hello World '
spam.strip()
```

Out[76]: 'Hello World'

In [77]:
```
spam.lstrip()
```

Out[77]: 'Hello World '

In [79]:
```
spam.rstrip()
```

Out[79]: ' Hello World'

In [80]:
```
spam = 'SpamSpamBaconSpamEggsSpamSpam'
spam.strip('ampS')
```

Out[80]: 'BaconSpamEggs'

The Count Method

- Counts the number of occurance of a given charater or subtracting in the string it is applied to. can be optionally provided start and end index

In [81]:
```
sentence  = 'one sheep two sheep three sheep four'
sentence.count('sheep')
```

Out[81]: 3

In [82]:
```
sentence.count('e')
```

Out[82]: 9

In [83]:
```
sentence.count('e',6)
```

Out[83]: 8

Replace Method

- Replace all occurances of a given substring with another substring.
- Can be optionally provided a third argument to limit the number of replacements. Returns a new string

In [84]:
```
text = "Hello, world!"
>>> text.replace("world", "planet")
```

Out[84]: 'Hello, planet!'

In [85]:
```
fruits = "apple, banana, cherry, apple"
fruits.replace("apple", "orange", 1)
```

Out[85]: 'orange, banana, cherry, apple'

In [86]:
```
sentence = "I like apples, Apples are my favorite fruit"
sentence.replace("apples", "oranges")
```

Out[86]: 'I like oranges, Apples are my favorite fruit'