

python comments

types of comments

1.single line comments

2.multi line comments

1. single line comments

>with the help of the single line comments to display the title of the page
>a single line comment denoted the symbol as #

syntax:

```
# title of the page corresponding to markdown format
```

2. multi line comments

a multi line comments to display the multi lines of title to display the markdown format only.

syntax 1:

```
'''-----  
-----'''
```

syntax 2:

```
"-----  
-----"
```

example:

'Survey on Arun ice cream'

In [1]:

```
'''Sruthi'''
```

Out[1]:

```
'Sruthi'
```

python data types

integer-int()

>it holds the integer values

string-str()

>it holds string values

float-float()

>it holds the floating type of data values

In [4]:

```
a=15  
type(a)
```

Out[4]:

int

In [5]:

```
a=12.3  
type(a)
```

Out[5]:

float

In [6]:

```
a="bhargavi"  
type(a)
```

Out[6]:

str

In [7]:

```
# convert the integer to float
```

```
m=2207  
n=float(m)  
print(n)  
type(n)
```

2207.0

Out[7]:

float

In [9]:

```
# convert the integer to string
```

```
v=2202  
b=str(v)  
type(b)
```

Out[9]:

str

In [12]:

```
## keywords python
*keywords*
import keyword
print(keyword.kwlist)
```

```
['False', 'None', 'True', 'and', 'as', 'assert', 'async', 'await', 'break',
'class', 'continue', 'def', 'del', 'elif', 'else', 'except', 'finally', 'for',
'from', 'global', 'if', 'import', 'in', 'is', 'lambda', 'nonlocal', 'not',
'or', 'pass', 'raise', 'return', 'try', 'while', 'with', 'yield']
```

Keywords are some predefined and reserved words in python that have special meanings. Keywords are used to define the syntax of the coding. The keyword cannot be used as an identifier, function, and variable name. All the keywords in python are written in lower case except True and False. There are 33 keywords in Python 3.7 let's go through all of them one by one.

In []:

Keyword	Description
1.and	A logical operator
2.as	To create an alias
3.assert	For debugging
4.break	To break out of a loop
5.class	To define a class
6.continue	To continue to the next iteration of a loop
7.def	To define a function
8.del	To delete an object
9.elif	Used in conditional statements, same as else if
10.else	Used in conditional statements
11.exception	Used with exceptions, what to do when an exception occurs
12.False	Boolean value, result of comparison operations
13.finally	Used with exceptions, a block of code that will be executed no matter if the exception occurs
14.for	To create a for loop
15.from	To import specific parts of a module
16.global	To declare a global variable
17.if	To make a conditional statement
18.import	To import a module
19.in	To check if a value is present in a list , tuple , etc.
20.is	To test if two variables are equal
21.lambda	To create an anonymous function
22.None	Represents a null value
23.nonlocal	To declare a non-local variable
24.not	A logical operator
25.or	A logical operator
26.pass	A null statement, a statement that will do nothing
27.raise	To raise an exception
28.return	To exit a function and return a value
29.True	Boolean value, result of comparison operations
30.try	To make a try...except statement
31.while	To create a while loop
32.with	Used to simplify exception handling
33.yield	To end a function, returns a generator

control statements

In [1]:

```
print("bhargavi")
```

bhargavi

In [4]:

```
s="hello world"  
s1=s.split()  
print(s1)
```

['hello', 'world']

In []:

```
# write a programm to find the biggest of two numbers  
# write a programm to check the given number is even or not  
# write a programm to check the given age is eligible for vote or not
```

In [12]:

```
# biggest of two numbers  
a=10  
b=20  
if(a>b):  
    print(a)  
else:  
    print(b)
```

20

In [13]:

```
# given number is even or not  
a=18  
if(a%2==0):  
    print('given number is even')  
else:  
    print('given number is not even')
```

given number is even

In [17]:

```
# given age is eligible for vote or not  
a=22  
if(a>18):  
    print('given age is eligible')  
else:  
    print('given age is not eligible')
```

given age is eligible

In [26]:

```
a=int(input('enter first number'))
b=int(input('enter second number'))
if(a>b):
    print(a,'first number is big')
else:
    print(b,'second number is big')
```

```
enter first number44
enter second number23
44 first number is big
```

In [39]:

```
a=int(input('enter the values....'))
if(a%2==0):
    print(a,'given number is even')
else:
    print(a,'given number is not even')
```

```
enter the values....44
44 given number is even
```

In [29]:

```
a=int(input('enter the age...'))
if(a>18):
    print('given age is eligible for vote')
else:
    print('given age is not eligible for vote')
```

```
enter the age...22
given age is eligible for vote
```

In [31]:

```
print("hai"+"good morning")
```

```
haigood morning
```

In [37]:

```
print("hi"+"7")
```

```
hi7
```

In [35]:

```
str("12+12")
```

Out[35]:

```
'12+12'
```

In [1]:

```
n=int(input("enter a numbers"))
# even-divisible by 2
# 0,2,4,6,8
if(n%2==0):
    print("even")
else:
    print("odd")
```

enter a numbers60
even

In []:

```
# to check the given character is vowel or constant?
# vowels:a,e,i,o,u
# constant: rest all characters.
```

```
elif statements:
    -to check the 2 or more conditions.
    syntax:
        if(condition):
            statements
        elif(condition):
            statements
        elif(condition):
            statements
        else:
            statements
```

In [3]:

```
# to check the given character is vowel or constant?
# vowels:a,e,i,o,u
ch=str(input("enter character....")) #ch=i
if(ch=='a' or ch=='e' or ch=='i' or ch=='o' or ch=='u'):
    print(ch,"it is vowel")
else:
    print(ch,"it is a constant")
```

enter character....v
v it is a constant

In [1]:

```
a=int(input('enter first value'))
b=int(input('enter second value'))
c=int(input('enter third value'))
if(a>b and a>c):
    print(a,"is the biggest")
elif(b>a and b>c):
    print(b,"is the biggest")
else:
    print (c,"is the biggest")
```

```
enter first value45
enter second value23
enter third value49
49 is the biggest
```

```

```

In []:

```
<img src=
```