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 symble as # syntax: # title of the page corresponding to markdon formate.

2.multi line comments

a multi line comment to display the multiple lines of title to display the markdon formate only.

syntax t1: '''---------- --------------'''

syntax t2: """--------- -------------"""

In [ ]:

```
## this is a single line comment
```

In [10]:

```
### ex:-
'''A multi line comment to display the multiple lines of title to display
the markdon formate only.'''
```

Out[10]:

```
'A multi line comment to display the multiple lines of title to display \nth
e markdon formate only.'
```

In [12]:

```
## this is a multi line comment
```

In [13]:

```
### ex:-
"""A multi line comment to display the multiple lines of title to display
the markdon formate only."""
```

Out[13]:

```
'A multi line comment to display the multiple lines of title to display \nth
e markdon formate only.'
```

Def of python

In [ ]:

In [ ]:

python data type

integer int():

---->it holdes the integer values

string str():

---->it holdes the string values

float float():

---->it holdes the floating type of data values

In [6]:

```
a=17.17
type(a)
```

Out[6]:

```
float
```

In [7]:

```
int="kesava"
type(int)
```

Out[7]:

```
str
```

In [8]:

```
int=10
type(10)
```

Out[8]:

```
int
```

In [9]:

```
## convert integer into float
m=7654
n=float(m)
print(n)
type(n)
```

```
7654.0
```

Out[9]:

```
float
```

In [10]:

```
## convert integer into string
m=7654
n=str(m)
print(n)
type(n)
```

```
7654
```

Out[10]:

```
str
```

convert string to float

convert string to string

we can not change the both type of functions

key words in python

In [12]:

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

```
['False', 'None', 'True', 'and', 'as', 'assert', 'async', 'await', 'break',
'class', 'continue', 'def', 'del', 'elif', 'else', 'except', 'finally', 'fo
r', 'from', 'global', 'if', 'import', 'in', 'is', 'lambda', 'nonlocal', 'no
t', 'or', 'pass', 'raise', 'return', 'try', 'while', 'with', 'yield']
```

keywords

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. except--------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 there is an exception or not
```

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

In [ ]:

```
# control statements
```

In [13]:

```
print("kesava")
```

```
kesava
```

In [16]:

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

```
['hello', 'world']
```

In [ ]:

```
Annam Kesava
```

In [ ]:

```
## writw a program to find the biggest of two number
## writw a program to check the age eligible is  for vote or not
## writw a program to check the givem number is even or not
```

In [12]:

```
#1
a= int(input("enter the first number"))
b= int(input("enter the second number"))
if (a>b):
   print("a is grater")
if (b>a):
   print("b is grater")
```

```
enter the first number40
enter the second number70
b is grater
```

In [15]:

```
n1=int(input("enter n1 value"))
n2=float(input("enter n2 value"))
if(n1>n2):
    print(n1,"is grater value....")
else:
    print(n2,"is grater value....")
```

```
enter n1 value50
enter n2 value50.9
50.9 is grater value....
```

In [24]:

```
#2
a=int(input("enter your number: "))
if(a%2==0):
    print(a," is even")
else:
    print(a," is not even")
```

```
enter your number: 21
21  is not even
```

In [29]:

```
#3
age=int(input("enter your age:"))
b="years"
print(age,b)
if(age>=18):
     print("your eligible to vote")
else:
    print("your not eligible to vote")
```

```
enter your age:38
38 years
your eligible to vote
```

In [27]:

```
print("hai"+"kesava")
```

```
haikesava
```

In [17]:

```
print("hai",12345)
```

```
hai 12345
```

In [ ]:

```
## to check the givem character vowels or constant?
#vowels: a,e,i,o,u
#constant:rest all charcters
#find the biggest 3 numbers
```

In [13]:

```
v1=int(input("enter n1 value"))
v2=int(input("enter n2 value"))
v3=int(input("enter n2 value"))
v4=int(input("enter n2 value"))
if(v1>v2 and v1>v3 and v1>v4):
    print(v1,"is grater value....")
elif(v2>v3 and v2>v4):
    print(v2,"is grater value....")
elif(v3>v4):
    print(v3,"is grater value....")
else:
    print(v4,"is grater value....")
```

```
enter n1 value0987
enter n2 value5648
enter n2 value98746
enter n2 value8765
98746 is grater value....
```

In [ ]:

```
## Dynamic
```

In [20]:

```
ch=str(input("Enter a letter:  "))
if(ch=='a'or ch=='e'or ch=='i' or ch=='o' or ch=='u'
   or ch=='A'or ch=='E'or ch=='I' or ch=='O' or ch=='U'):
    print(ch," is a vowel")
else:
    print(ch,"is not a vowel")
```

```
Enter a letter:  k
k is not a vowel
```

In [ ]:

```
## Static
```

In [19]:

```
ch="k"
if(ch=='a'or ch=='e'or ch=='i' or ch=='o' or ch=='u'
   or ch=='A'or ch=='E'or ch=='I' or ch=='O' or ch=='U'):
    print(ch,"is a vowel")
else:
    print(ch,"is not a vowel")
```

```
k is not a vowel
```

In [21]:

```
ch=str(input("Enter a letter:  "))
if(ch=='a'or ch=='e'or ch=='i' or ch=='o' or ch=='u'
   or ch=='A'or ch=='E'or ch=='I' or ch=='O' or ch=='U'):
    print(ch," is a vowel")
elif(ch=='b'or ch=='c'or ch=='d' or ch=='f' or ch=='g'
     or ch=='h'or ch=='j'or ch=='k' or ch=='l' or ch=='m'
     or ch=='n'or ch=='p'or ch=='q' or ch=='r' or ch=='s'
    or ch=='t'or ch=='v'or ch=='w' or ch=='x' or ch=='y' or ch=='z'
    or ch=='B'or ch=='C'or ch=='D' or ch=='F' or ch=='G'
     or ch=='H'or ch=='J'or ch=='K' or ch=='L' or ch=='M'
     or ch=='N'or ch=='P'or ch=='Q' or ch=='R' or ch=='S'
    or ch=='T'or ch=='V'or ch=='W' or ch=='X' or ch=='Y' or ch=='Z'):
    print(ch,"  is a consonant")
else:
    print("It is not a alphabet")
```

```
  File "<ipython-input-21-7be9d2ca4f75>", line 3
    or ch=='A'or ch=='E'or ch=='I' or ch=='O' or ch=='U'):
    ^
SyntaxError: invalid character in identifier
```

In [29]:

```
ch=str(input("Enter a letter:  "))
if(ch=='a'or ch=='e'or ch=='i' or ch=='o' or ch=='u'
     or ch=='A' or ch=='E' or ch=='I' or ch=='O' or ch=='U'):
    print(ch," is a vowel")
elif(ch=='b'or ch=='c'or ch=='d' or ch=='f' or ch=='g'
     or ch=='h'or ch=='j'or ch=='k' or ch=='l' or ch=='m'
     or ch=='n'or ch=='p'or ch=='q' or ch=='r' or ch=='s'
    or ch=='t'or ch=='v'or ch=='w' or ch=='x' or ch=='y' or ch=='z'
    or ch=='B'or ch=='C'or ch=='D' or ch=='F' or ch=='G'
     or ch=='H'or ch=='J'or ch=='K' or ch=='L' or ch=='M'
     or ch=='N'or ch=='P'or ch=='Q' or ch=='R' or ch=='S'
    or ch=='T'or ch=='V'or ch=='W' or ch=='X' or ch=='Y' or ch=='Z'):
    print(ch,"  is a consonant")
else:
    print("It is not a alphabet")
```

```
  File "<ipython-input-29-4bd975e0c91b>", line 3
    or ch=='A' or ch=='E' or ch=='I' or ch=='O' or ch=='U'):
    ^
SyntaxError: invalid character in identifier
```

```
<img src ="https://image.shutterstock.com/image-photo/mountains-under-mist-morning-amazing-260nw-1725825019.jpg" width = 400 length = 400>
```

In [44]:

```
<img src ="C:\Users\exam4\Downloads\k" width = 400 length = 400>
```

```
  File "<ipython-input-44-690cba179b17>", line 1
    <img src ="C:\Users\exam4\Downloads\k" width = 400 length = 400>
    ^
SyntaxError: invalid syntax
```

In [ ]: