PYTHON COMMENTS

TYPES OF COMMENTS:

1.SINGLE LINE COMMENTS

With the help of single line comment we can display the title of the page

A single line denoted by the symbol as

syntax:

title of the page corresponding to markdown formate

2.MULTI LINE COMMENTS

A multi line comments to display the multiple lines of title to display the multiple lines of title to display the markdown formate only.

1st syntax:

'''------------------------

----------------------'''

2nd syntax:

"""------------------------

"""

'''SURVEY ON APPLE PRODUCTS'''

"""SURVEY ON APPLE PRODUCTS A short survey to find consumers perception while purchasing Apple products """

python data-types:

integer-int()

it holds the integer value

string-str()

it holds the string value

float-float()

it holds the floating type of data values

In [3]:

```
a=10
type(a)
```

Out[3]:

```
int
```

In [4]:

```
b=10.23
type(b)
```

Out[4]:

```
float
```

In [6]:

```
a='asma'
type(a)
```

Out[6]:

```
str
```

In [7]:

```
# convert integer to string
m=5877
n=str(m)
type(n)
```

Out[7]:

```
str
```

In [9]:

```
# integer to float
a=7758
b=float(a)
type(b)
```

Out[9]:

```
float
```

python keywords

In [10]:

```
# 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']
```

Python keywords are reserved words. They are used by python interpreters to understand the program. Keywords define the structure of programs. We can't use keywords to name program entities such as variables, classes, and functions. all the keywords in python are written in lowercase except true and false. there are 33 keywords in python 3.7 lets go through all of them one by one

Keyword Description and A logical operator as To create an alias assert For debugging break To break out of a loop class To define a class continue To continue to the next iteration of a loop def To define a function del To delete an object elif Used in conditional statements, same as else if else Used in conditional statements except Used with exceptions, what to do when an exception occurs False Boolean value, result of comparison operations finally Used with exceptions, a block of code that will be executed no matter if there is an exception or not for To create a for loop from To import specific parts of a module global To declare a global variable if To make a conditional statement import To import a module in To check if a value is present in a list, tuple, etc. is To test if two variables are equal lambda To create an anonymous function None Represents a null value nonlocal To declare a non-local variable not A logical operator or A logical operator pass A null statement, a statement that will do nothing raise To raise an exception return To exit a function and return a value True Boolean value, result of comparison operations try To make a try...except statement while To create a while loop with Used to simplify exception handling yield To end a function, returns a generator

In [ ]:

```
# control statements
```

In [11]:

```
print("asma")
```

```
asma
```

In [12]:

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

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

In [ ]:

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

In [15]:

```
# biggest of two numbers
a=40
b=60
if(a>b):
    print(a)
else:
    print(b)
```

```
60
```

In [17]:

```
# even or not
n=22
if(n%2==0):
    print('the given number is even')
else:
    print('the given number is not even')
```

```
the given number is even
```

In [18]:

```
# age is eligible or not
x=25
if(x>=18):
    print("the given age is eligible for vote")
else:
    print("the given age is not eligible")
```

```
the given age is eligible for vote
```

In [23]:

```
a=int(input("enter first number:"))
b=int(input("enter second number:"))
if(a>b):
    print(a," is the biggest value")
else:
    print(b," is the biggest value")
```

```
enter first number:58
enter second number:77
77  is the biggest value
```

In [20]:

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

```
enter any number:77
number is not even
```

In [21]:

```
a=int(input("enter age:"))
if(a>=18):
    print("entered age is eligible for vote")
else:
    print("entered age is not eligible for vote")
```

```
enter age:22
entered age is eligible for vote
```

In [24]:

```
print("hai"+"sruthi")
```

```
haisruthi
```

In [25]:

```
print("hi",5877)
```

```
hi 5877
```

In [26]:

```
print(123456)
```

```
123456
```

In [27]:

```
print(6+6)
```

```
12
```

In [28]:

```
print("12+12")
```

```
12+12
```

In [29]:

```
print("hai12345")
```

```
hai12345
```

In [31]:

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

```
hai12345
```

In [ ]:

```
# elif statement
### to check two or more conditions
### syntax:
###   if(condition):
###       statements
###   elif(condition):
###       statements
###   elif(condition)
###       statements
###       else:
###           statements
```

In [34]:

```
n=int(input("enter a number:"))
# even-divisible by 2
# 2,4,6,8,10,12,14,16
if(n%2==0):
    print("even")
else:
    print("odd")
```

```
enter a number:77
odd
```

In [2]:

```
# write a program to check the given character is vowels or consonent?
# vowels: a,e,i,o,u
# consonents: rest all characters
cha=str(input("enter any character:"))
if(cha=='a' or cha=='e' or cha=='i' or cha=='o' or cha=='u'):
    print(cha,"given character is vowel")
else:
    print(cha,"given character is consonent")
```

```
enter any character:j
j given character is consonent
```

In [6]:

```
# find the biggest of 3 numbers
a=int(input("enter first value"))
b=int(input("enter second value"))
c=int(input("enter third value"))
if(a==b and b==c):
    print("three numbers are equal")
elif(a>b and a>c):
    print(a,"is the biggest value")
elif(b>a and b>c):
    print(b,"is the biggest value")
else:
    print(c,"is the biggest value")
```

```
enter first value77
enter second value58
enter third value34
77 is the biggest value
```

In [ ]: