difference between c,java,pyton

c int a, a=10; printf("a") java int a=10 system.out.println("+a"); pyton int a=10 print(a)

In [11]:

```
print("rvr&jc college")
```

```
rvr&jc college
```

In [12]:

```
#assign a variable to a value
a=("rvr&jc college")
print(a)
```

```
rvr&jc college
```

In [13]:

```
print('/n',a*10)
```

```
/n rvr&jc collegervr&jc collegervr&jc collegervr&jc collegervr&jc collegervr
&jc collegervr&jc collegervr&jc collegervr&jc collegervr&jc college
```

In [14]:

```
print(a*7)
```

```
rvr&jc collegervr&jc collegervr&jc collegervr&jc collegervr&jc collegervr&jc
collegervr&jc college
```

In [6]:

```
print("siva\n"*10)
```

```
siva
siva
siva
siva
siva
siva
siva
siva
siva
siva
```

In [16]:

```
# Adittion of two numbers
a=10
b=10
c=a+b
print("the addition of two numbers is=",a+b)
```

```
the addition of two numbers is= 20
```

In [17]:

```
# Adittion of two numbers
a=b=10
print("the addition of two numbers is=",a+b)
```

```
the addition of two numbers is= 20
```

In [18]:

```
# substriction of two numbers
a=20
b=10
c=a-b
print(c)
```

```
10
```

In [19]:

```
# multipliction of two numbers
a=12
b=19
c=a*b
print(c)
```

```
228
```

In [20]:

```
# division of two numbers
a=12
b=19
c=a/b
print(c)
```

```
0.631578947368421
```

In [7]:

```
# CHANGE A STRING TO LOWER TO UPPER
string ="siva"
string.upper()
```

Out[7]:

```
'SIVA'
```

In [8]:

```
# CHANGE A STRING TO UPPER TO LOWER
string ="SIVA"
string.lower()
```

Out[8]:

```
'siva'
```

In [9]:

```
# reverse of a string
string[::-1]
```

Out[9]:

```
'AVIS'
```

In [11]:

```
# string concatination
a="siva"
b="prasad"
c=a+b
print(c)
```

```
sivaprasad
```

In [12]:

```
#acsesing first element of a given string
a="siva"
a[0]
```

Out[12]:

```
's'
```

In [13]:

```
#acesing last element of a given string
a="siva"
a[-1]
```

Out[13]:

```
'a'
```

In [14]:

```
#length of the given string
a="siva"
print(len(a))
```

```
4
```

In [15]:

```
a[2:4]
```

Out[15]:

```
'va'
```

In [29]:

```
# dynamic values addition
a=10
b=20
c=a+b
print(c)
```

```
30
```

In [ ]:

```
a=int(input("Enter A Value"))
b=int(input("Enter B Value"))
c=a+b
print("Addition of Two Numbers A&B is:",c)
```

In [ ]:

```
a=int(input("Enter A Value"))
b=int(input("Enter B Value"))
c=a-b
print("Subtraction of Two Numbers A&B is:",c)
```

In [ ]:

```
a=int(input("Enter A Value"))
b=int(input("Enter B Value"))
c=a*b
print("Multiplication of Two Numbers A&B is:",c)
```

In [ ]:

```
a=int(input("Enter A Value"))
b=int(input("Enter B Value"))
c=a/b
print("Division of Two Numbers A&B is:",c)
```

In [ ]:

```
# HOW TO PRINT THE MULTIPLICATION TABLE
n=12
for i in range(1,11):
    print(n,'*',i,'=',n*i)
```

In [ ]:

```
# how to print the multiplication table
n=14
for i in range(1,11):
    print(n,'*',i,'=',n*i)
```

Pyton Defination

pyton is a most popular programming language

Server to create the web applications

It can used for network transcations

Pyton can be used to system scripting

Pyton can be used to connect the remoteservice

Pyton can be used to connect the datebase to real time operations

this is second comment

this is third comment

python operators

operators are used to perform operations on variables&values

Airthmetic operators

Assignment operators

Comparison operators

Logical operators

In [3]:

```
print(10+5)
```

```
15
```

In [4]:

```
print(15-10)
```

```
5
```

In [ ]:

```
print(12*12)
```

In [1]:

```
print(16%2)
```

```
0
```

In [1]:

```
print(10/5)
```

```
2.0
```

In [1]:

```
print(12**12)
```

```
8916100448256
```

In [ ]:

```
Assignment operators
```

In [2]:

```
x=5
print(x)
```

```
5
```

In [1]:

```
x=80
y=90
if(x==y):
    print("yes")
else:
        print("no")
```

```
no
```

In [7]:

```
x=5
x+=4
print(x)
```

```
9
```

In [8]:

```
x=3
x-=2
print(x)
```

```
1
```

In [9]:

```
x=6
x*=5
print(x)
```

```
30
```

In [10]:

```
x=8
x/=16
print(x)
```

```
0.5
```

In [11]:

```
x=12
x%=24
print(x)
```

```
12
```

In [12]:

```
x=15
x//=30
print(x)
```

```
0
```

In [13]:

```
x=17
x**=34
print(x)
```

```
6843264508857750340489467199257549104873329
```

In [14]:

```
x=24
x&=12
print(x)
```

```
8
```

In [15]:

```
x=11
x!=11
print(x)
```

```
11
```

In [16]:

```
x=17
x^=17
print(x)
```

```
0
```

In [17]:

```
x=19
x>>=19
print(x)
```

```
0
```

In [29]:

```
x=55
x<<=77
print(x)
```

```
831136500985057557610496O
```

Comparison operator

In [ ]:

```
== equal to
!=not equal to
>=greater than equal to
<=less than equal to
```

In [30]:

```
x=2
y=3
print(x==y)
```

```
False
```

In [33]:

```
x=2
y=3
print(x>=y)
```

```
False
```

In [34]:

```
x=2
y=3
print(x<=y)
```

```
True
```

In [35]:

```
x=2
y=3
print(x!=y)
```

```
True
```

Logical operator

In [1]:

```
x=50
print(x>3 & x<25)
type(x)
```

```
True
```

Out[1]:

```
int
```

In [2]:

```
x=50
print(x>3 or x<25)
type(x)
```

```
True
```

Out[2]:

```
int
```

In [3]:

```
x=50
print(not(x>3 & x<25))
type(x)
```

```
False
```

Out[3]:

```
int
```

Python comments

In [ ]:

```
Types of comments
1.Single line comments
2. Multi line comments


1.Single line comments
>with the help of single line comments to display the titile of the page.
>a single line comment denoted the symbol as #
syntax:
        #title of the page corresponding to markdown formate.
  2.Multi line comments
A multi line comment to display the multiple lines of title to display the markdown formate

1st Syntax:
        '''------------------------------
    ----------------------------'''
2nd Syntax:
            """--------------------------
            --------------------------"""
```

Example of single line comment

Defination Of Python

- Python is the most popular language
- Server to create the web application
- It is used for net transactions
- System scripting * connect to the remote servers
- Connect the database to real time

'''Siva prasad'''

PYTHON DATA-TYPES:

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 [4]:

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

```
10
```

Out[4]:

```
int
```

In [5]:

```
a=10.5
print(a)
type(a)
```

```
10.5
```

Out[5]:

```
float
```

In [6]:

```
k="Siva"
type(k)
```

Out[6]:

```
str
```

In [7]:

```
# convert the integer to string
k=234
n=str(k)
type(n)
```

Out[7]:

```
str
```

In [8]:

```
# convert the integer to float
k=2345
n=float(k)
type(n)
```

Out[8]:

```
float
```

Keywords python

keywords

import keyword print(keyword.kwlist)

In [ ]:

```
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 iden
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
```

Control Statements

In [1]:

```
print("welcome")
```

```
welcome
```

Write a program to find the biggest of two numbers

In [4]:

```
a=12
b=24
if (a>b):
    print("b is bigger than a")
else:
    print("a is bigger than b")
```

```
a is bigger than b
```

Write a program to check the given numbers is even or not

In [10]:

```
num=15
if(num % 5==0):

  print('{0}is an even number'.format(num))
else:
 print('{0}is not an even number'.format(num))
```

```
15is an even number
```

Write a program to check the given age is eligible for vote or not

In [11]:

```
age=int(input("Enter your age:"))

print("You are eligible for voting" if age>18
    else "You are not eligible for voting")
```

```
Enter your age:25
You are eligible for voting
```

In [12]:

```
n1=int(input("enter n1 values: "))
n2 =int(input("enter n2 values:"))
if(n1>n2):
    print('n1 is greater than n2')
else:
    print('n2 is greater than n1')
```

```
enter n1 values: 21
enter n2 values:31
n2 is greater than n1
```

In [14]:

```
n=int(input("Enter a number"))
# Even-divisible -5
# 5,10,15,20,25
if(n%2==0):
    print("even")
else:
    print("odd")
```

```
Enter a number0
even
```

To check the given charcter is vowel or constant

In [2]:

```
ch =str(input("enter charcters"))
if(ch=='a'or ch=='e'or ch=='1'or ch=='0'or ch=='u' ):
    print(ch,"vowel")
else:
    print(ch,"constant")
```

```
enter charcterss
s constant
```

In [7]:

```
n1=int(input("enter n1 value"))
n2=int(input("enter n2 value"))
n3=int(input("enter n3 value"))
if(n1>n2):
    print("n1 is greater than n2")
elif(n2>n3):
    print("n2 is greater than n3")
else:
    print("n3 is greater than n1 and n2")
```

```
enter n1 value21
enter n2 value23
enter n3 value27
n3 is greater than n1 and n2
```

Write a program to print 1to10 natural numbers

In [1]:

```
print('Numbers from 1to10:')
for n in range(1,11):
    print (n,end='')
```

```
Numbers from 1to10:
12345678910
```

To give the step value to print the odd numbers from starting value as 1 and endinding to 100 for

In [5]:

```
num=100

for num in range(1,num + 1,2):
     print(num,end=" ")
```

```
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31 33 35 37 39 41 43 45 47 49 51 53
55 57 59 61 63 65 67 69 71 73 75 77 79 81 83 85 87 89 91 93 95 97 99
```

In [ ]:

```
# to print the 0 to 50 elements
```

In [6]:

```
for i in range(0,50,3):
    print(i,end=" ")
```

```
0 3 6 9 12 15 18 21 24 27 30 33 36 39 42 45 48
```

In [ ]:

```
# To print 1 to n natural numbers in ascending order
```

In [7]:

```
n=int(input("enter a natural number size"))
for i in range(1,n+1):
    print(i,end=" ")
```

```
enter a natural number size50
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 2
9 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50
```

In [ ]:

```
# To print 1 to n natural numbers in descending order
```

In [10]:

```
 n=int(input("enter a natural number size"))
for i in range(n,0,-1):
        print(i,end=" ")
```

```
enter a natural number size23
23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1
```

Break statement example in python

In [14]:

```
for i in('apssdc'):
    if i=='d':
       break
    print(i,end=" )
```

```
a p s s
```

In [29]:

```
for i in ('1234567890'):
    if i=='6':
     break
    print(i,end= " ")
```

```
1 2 3 4 5
```

In [19]:

```
for i in('apssdc'):
    if i=='d':
       break
    print(i)
```

```
a
p
s
s
```

In [22]:

```
for i in range(1,10):
    if i=='5' :
        break
    else:
        print(i,=" ")
```

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

To print the even numbers in between 1to20 using continue keyword

In [42]:

```
for i in range(1,21):
    if(i%2!=0):
        continue
    else:
        print(i,end="")
```

```
2468101214161820
```

In [43]:

```
#Swaping of two numbers
a=28
b=23
temp=a
a=b
b=temp
print(a,b)
```

```
23 28
```

In [44]:

```
S=33
T=28
S,T=T,S

print(S)
print(T)
```

```
28
33
```

In [49]:

```
#How to generate a random number in python
import random
print(random.randint(0,8))
```

```
6
```

In [50]:

```
import random
print(random.randint(10,18))
```

```
13
```

In [54]:

```
# To print the english alphabets from upper case to lower case in python
import string
for letter in string.ascii_uppercase:
    print(letter,end=" ")
for letter in string.ascii_lowercase:
    print(letter,end=" ")
```

```
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z a b c d e f g h i j k l
m n o p q r s t u v w x y z
```

In [57]:

```
#progrm to display calender of the year and month
import calendar
year = 2022
month = 9
print(calendar.month(year,month))
```

```
   September 2022
Mo Tu We Th Fr Sa Su
          1  2  3  4
 5  6  7  8  9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28 29 30
```

In [62]:

```
import calendar
year = 1998
month =8
print(calendar.month(year,month))
```

```
    August 1998
Mo Tu We Th Fr Sa Su
                1  2
 3  4  5  6  7  8  9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29 30
31
```

In [17]:

```
def add(a,b):
    c=a+b
    return c
print (add(2,3))
```

```
5
```

In [25]:

```
#With arguments and with return values
n1=int(input("enter n1 values"))
n2=int(input("enter n2 values"))
def addition(a,b):
    c=a+b
    return c
addition(n1,n2)
```

```
enter n1 values10
enter n2 values20
```

Out[25]:

```
30
```

In [26]:

```
n1=int(input("enter n1 value:"))
n2=int(input("enter n2 values"))
def sub(a,b):
    c=a-b
    print(c)
sub(n1,n2)
```

```
enter n1 value:20
enter n2 values10
10
```

In [29]:

```
def adding():
    a=25
    b=45
    sum=a+b
    print("after calling:",sum)
adding()
```

```
after calling: 70
```

In [30]:

```
def multiplication():
    a=15
    b=35
    multi = a*b
    return multi
print("after calling the multiplication:",multiplication())
```

```
after calling the multiplication: 525
```

Lists

In [ ]:

```
> A list is a collection of charcter variables,and
>number variables and boolean values datatytpes
>a list is a to store multiple data with in a single variable
>a list is a ordered type of data
>a is denoted as []
>a list item as denoted with double quotes:

syntax:
    items=["item1","item2","item3"]
    print(items)
```

In [11]:

```
# example for the list
li=["apple","banana","orange","grapes","lemon"]
li
```

Out[11]:

```
['apple', 'banana', 'orange', 'grapes', 'lemon']
```

In [5]:

```
# type of the list
print (type(li))
```

```
<class 'list'>
```

In [12]:

```
# length of the list
print(len(li))
```

```
5
```

In [10]:

```
# accessing first element in a list

print(li[0])
```

```
apple
```

In [13]:

```
# accessing the last element in a list
print(li[-1])
```

```
lemon
```

In [15]:

```
li[0]="milk"
li
```

Out[15]:

```
['milk', 'banana', 'orange', 'grapes', 'lemon']
```

In [16]:

```
li
```

Out[16]:

```
['milk', 'banana', 'orange', 'grapes', 'lemon']
```

In [18]:

```
li.insert(1,"siva")
li
```

Out[18]:

```
['milk', 'siva', 'siva', 'banana', 'orange', 'grapes', 'lemon']
```

In [23]:

```
li1=["siva," "123","True"]
li1
```

Out[23]:

```
['siva,123', 'True']
```

In [29]:

```
li[2:5]
```

Out[29]:

```
['siva', 'banana', 'orange']
```

In [30]:

```
li[3:]
```

Out[30]:

```
['banana', 'orange', 'grapes', 'lemon']
```

In [31]:

```
li[:5]
```

Out[31]:

```
['milk', 'siva', 'siva', 'banana', 'orange']
```

In [33]:

```
li.remove("orange")
```

In [34]:

```
del li[1]
```

In [35]:

```
li.sort
li
```

Out[35]:

```
['milk', 'siva', 'banana', 'grapes', 'lemon']
```

In [39]:

```
# list using loop

for i in li:
    print(i,end="")
```

```
milksivabananagrapeslemon
```

In [40]:

```
#accessing the list using loop
for num in li:
    print(num,end=" ")
```

```
milk siva banana grapes lemon
```

Tuple

it is a collection of different types of data.

it is immutable(cant change)

we can using round brackets() to write a tuple.

to create the empty tuple

tuple_name=()

to create single values

tuple_name=(values) 1

to create multiple values

tuple_name=(value1,value2....)

In [41]:

```
#create tuple
t1=(10,20,30)
t1
print(type(t1))
```

```
<class 'tuple'>
```

In [43]:

```
# single value tuple
t2=(10)
print(type(t2))
t3=(20,)
print(type(t3))
```

```
<class 'int'>
<class 'tuple'>
```

In [44]:

```
t3
```

Out[44]:

```
(20,)
```

In [45]:

```
t2
```

Out[45]:

```
10
```

In [46]:

```
#how to access the values from the tuple

t1
print(t1[2])
```

```
30
```

In [47]:

```
print(t1[0:1])
```

```
(10,)
```

In [48]:

```
t2=(10,20,10,20,30,20,20,30,10)
# to count the number of ocurences
t2.count(10)
```

Out[48]:

```
3
```

In [49]:

```
t2.count(20)
```

Out[49]:

```
4
```

In [50]:

```
t2.count(30)
```

Out[50]:

```
2
```

In [52]:

```
#index
t2.index(20)
```

Out[52]:

```
1
```

In [53]:

```
t2.index(10)
```

Out[53]:

```
0
```

In [54]:

```
t2.index(30)
```

Out[54]:

```
4
```

In [55]:

```
tuple1 =("abc", 34, True, 40, "male")

print(tuple1)
```

```
('abc', 34, True, 40, 'male')
```

In [ ]:

```
# length if the list
print()
```

In [ ]:

```
#Dictionary
-it is a collection of different data types,
-it is a group of key and values(key:value)->item
-in dictionary keys are unique
-written in({}
-each and every item seperated with commas(,)
-ccessing dictionary values by using key names
-it is a mutable(changable)
```

In [ ]:

```
to create a empty dictionary:
    -dictionary_name={}
```

In [ ]:

```
to create the dictionary values:
    dictionaries-name=(key:value,key:value2...}
```

In [2]:

```
# to create a dictionary with values
d1={'a':10,'b':34,'c':45}
print(d1)
print (type(d1))
```

```
{'a': 10, 'b': 34, 'c': 45}
<class 'dict'>
```

In [14]:

```
# to create a dictionaries with different data types..
d2={'a':100,'name':'siva','branch':'MCA','b':45.8}
print(d2)
```

```
{'a': 100, 'name': 'siva', 'branch': 'MCA', 'b': 45.8}
```

In [8]:

```
# accessing the dictionary values using the key names
print(d2['name'])
print(d2['b'])
print(d2['a'])
```

```
siva
45.8
100
```

In [15]:

```
# update the dictionary values using the key names
print(d2)
d2['branch']='MBA'
print(d2)
```

```
{'a': 100, 'name': 'siva', 'branch': 'MCA', 'b': 45.8}
{'a': 100, 'name': 'siva', 'branch': 'MBA', 'b': 45.8}
```

In [10]:

```
print(dir(dict))
```

```
['__class__', '__contains__', '__delattr__', '__delitem__', '__dir__', '__do
c__', '__eq__', '__format__', '__ge__', '__getattribute__', '__getitem__',
'__gt__', '__hash__', '__init__', '__init_subclass__', '__iter__', '__le__',
'__len__', '__lt__', '__ne__', '__new__', '__reduce__', '__reduce_ex__', '__
repr__', '__reversed__', '__setattr__', '__setitem__', '__sizeof__', '__str_
_', '__subclasshook__', 'clear', 'copy', 'fromkeys', 'get', 'items', 'keys',
'pop', 'popitem', 'setdefault', 'update', 'values']
```

In [16]:

```
#keys
print(d2)
print(d2.keys())
```

```
{'a': 100, 'name': 'siva', 'branch': 'MBA', 'b': 45.8}
dict_keys(['a', 'name', 'branch', 'b'])
```

In [17]:

```
#values()
print(d2)
print(d2.values())
```

```
{'a': 100, 'name': 'siva', 'branch': 'MBA', 'b': 45.8}
dict_values([100, 'siva', 'MBA', 45.8])
```

In [18]:

```
#items
print(d2)
print(d2.items())
```

```
{'a': 100, 'name': 'siva', 'branch': 'MBA', 'b': 45.8}
dict_items([('a', 100), ('name', 'siva'), ('branch', 'MBA'), ('b', 45.8)])
```

In [19]:

```
#copy
print(d2)
d3=d2.copy()
print(d3)
print(type(d3))
```

```
{'a': 100, 'name': 'siva', 'branch': 'MBA', 'b': 45.8}
{'a': 100, 'name': 'siva', 'branch': 'MBA', 'b': 45.8}
<class 'dict'>
```

In [20]:

```
#get
print(d2)
print(d2.get('a'))
print(d2.get('name'))
```

```
{'a': 100, 'name': 'siva', 'branch': 'MBA', 'b': 45.8}
100
siva
```

In [21]:

```
#set default
print(d2)
print(d2.setdefault('rollno',310))
print(d2)
```

```
{'a': 100, 'name': 'siva', 'branch': 'MBA', 'b': 45.8}
310
{'a': 100, 'name': 'siva', 'branch': 'MBA', 'b': 45.8, 'rollno': 310}
```

In [22]:

```
#pop
print(d2)
print(d2.pop('b'))
```

```
{'a': 100, 'name': 'siva', 'branch': 'MBA', 'b': 45.8, 'rollno': 310}
45.8
```

In [23]:

```
#pop item
print(d2)
print(d2.popitem())
```

```
{'a': 100, 'name': 'siva', 'branch': 'MBA', 'rollno': 310}
('rollno', 310)
```

In [24]:

```
#clear
print(d2)
print(d2.clear())
```

```
{'a': 100, 'name': 'siva', 'branch': 'MBA'}
None
```

In [25]:

```
#clear
print(d2)
print(d2.clear())
```

```
{}
None
```

In [ ]:

```

```