HELLO WORLD PROGRAMME

Print(helloworld)

difference between c,java,python ¶

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

In [1]:

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

```
rvr&jc college
```

In [2]:

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

```
rvr&jc college
```

In [7]:

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

```
gopal
gopal
gopal
gopal
gopal
gopal
gopal
gopal
gopal
gopal
```

In [8]:

```
# Addition ot two numbers
a=100
b=33
c=a+b
print("the addition of two numbers=",c)
```

```
133
```

In [5]:

```
# sub of two numbers
a=22
b=44
c=a-b
print(c)
```

```
-22
```

In [6]:

```
# Multiplication of two numbers
a=12
b=44
c=a*b
print(c)
```

```
528
```

In [7]:

```
# modulus of two numbers
a=44
b=22
d=a/b
print(d)
```

```
2.0
```

In [8]:

```
# comparison of two numbers
a=10
b=23
d=a>b
e=b>a
print(d)
print(e)
```

```
False
True
```

In [9]:

```
# comparison of three numbers
a=12
b=23
c=343
a>b
else
b>c
else
c>b
print(larger number)
```

```
  File "<ipython-input-9-416ee435ca7f>", line 6
    else
    ^
SyntaxError: invalid syntax
```

In [11]:

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

Out[11]:

```
'NAGA SAI'
```

In [12]:

```
string[::-1]
```

Out[12]:

```
'ias agan'
```

In [ ]:

```
# string concartiation
a="naga"
b="sai"
c=a+b
print(c)
```

In [13]:

```
# acsesing first two elements of a given string
a="nagasai"
a[1:3]
```

Out[13]:

```
'ag'
```

In [ ]:

```
# acsesing last two elements of a given string
a="nagasai"
a[-1:-2]
```

In [14]:

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

```
7
```

In [ ]:

```
# dynamic value addition
a=10
b=23
c=a+b
print(c)
```

In [ ]:

```
a=int(input("Enter A value"))
b=int(input("Enter B value"))
c=a+b
print("Additon of Two numbers A&B is:",c)
```

In [ ]:

```
a=int(input("Enter A Value"))
b=int(input("Enter B value"))
c=a+b
print("Additon 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=19
for i in range(1,11):
print(n,'*',i,'=',n*i)
```

In [ ]:

```
```

In [ ]:

```
# how to print the multiplication table
n=19
for i in range(1,11):
    print(n,'*',i,'=',n*i)
```

THIS IS A COMMENT

This is a second comment

this is a third comment

PYTHON DEFINATION

In [ ]:

```
#PYTHON DEFINATION
```

PYTHON DEFINATION

pyhton is a most popular programming language

server to create the web applications

it can be used for network transactions

python can be used for user setting

python can be used to connect the remoteserver

python can be used to connect the real time operations

In [ ]:

```
# addtion of two numbers
a=12
b=23
c=a+b
print(a)
```

In [ ]:

```
# print the multiplication table
n=21
for i in range(1,11):
    print(n,'*',i,'=',n*i)
```

In [4]:

```
# swaping of two numbers
a=10
b=22
temp=a
a=b
b=temp
print(a,b)
```

```
22 10
```

In [1]:

```
x=5
y=10
x= input('enter value of x: ')
y= input('enter value of y: ')
temp =x
x=y
y=temp
print('the value of x after swapping: {}'.format(x))
print('the value of y after swapping: {}'.format(y))
```

```
enter value of x: 5
enter value of y: 10
the value of x after swapping: 10
the value of y after swapping: 5
```

python operator

operators are used to perform operations on variables and values

Arthemetic operator

Assignement opeators

comparison operator

Logic operator

Bitwise Operator

In [5]:

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

```
15
```

In [6]:

```
print(10-5)
```

```
5
```

print(10*5)

ASSIGNMENT OPERATOS

In [10]:

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

```
10
```

In [14]:

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

```
yes
```

In [16]:

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

```
9
```

In [17]:

```
x=5
x-=10
print(x)
```

```
-5
```

In [18]:

```
x=15
x*=23
print(x)
```

```
345
```

In [19]:

```
x=15
x/=23
print(x)
```

```
0.6521739130434783
```

In [20]:

```
x=19
x%=222
print(x)
```

```
19
```

In [23]:

```
x=222222
x//=3333
print(x)
```

```
66
```

In [25]:

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

```
625
```

x=5 x&=15 print(x)

In [27]:

```
x=33
x&=44
print(x)
```

```
32
```

In [28]:

```
x=44
x|=23
print(x)
```

```
63
```

In [29]:

```
x=44
x^=444
print(x)
```

```
400
```

In [30]:

```
x=5
x>>44
print(x)
```

```
5
```

In [31]:

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

```
55
```

Comparison operators

== equal to |=not equal to

> greater than <less than = greater than equal to <= less than equal to

In [32]:

```
x=5
y=3
print(x==y)
```

```
False
```

In [34]:

```
x=5
y=3
print(x>=y)
```

```
True
```

In [35]:

```
x=5
y=3
print(x<=y)
```

```
False
```

In [36]:

```
x=5
y=3
print(x!=y)
```

```
True
```

Logical Operator

In [38]:

```
x=55
print(x>3 and x<10)
type(x)
```

```
False
```

In [39]:

```
x=55
print(x>3 or x<10)
type(x)
```

```
True
```

Out[39]:

```
int
```

In [40]:

```
x=55
print(not(x>3 and  x<10))
type(x)
```

```
True
```

Out[40]:

```
int
```

In [1]:

```
x=5
y=6
temp=x
a=b
b=temp
print(a,b)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-1-3a9d140d7bee> in <module>
      2 y=6
      3 temp=x
----> 4 a=b
      5 b=temp
      6 print(a,b)

NameError: name 'b' is not defined
```

In [6]:

```
# SWAPPING OF TWO NUMBERS
x=5
y=3
temp=X
x=y
y=temp
print(x,y)
```

```
3 5
```

In [ ]:

```
python comments
```

types of comments

1. single line comments
2. multi line comments
3. single line comments

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

title of the page corresponding to markdown

formate.

```
 2. Multi line commetns
 a multi line commments to display the multiple lines of title to didplay the m
arkdownformate only.

 1 ST Syntax:
 '''-------------------------------------------------
 -------------------------------------------------'''
 2 nd syntax:
     """ ----------------------------------------
     ---------------------------------------
     """
```

''' naga sai9 '''

In [16]:

```
''' nagasai'''
```

Out[16]:

```
' nagasai'
```

In [20]:

```
"""naga sai kota\n
repalle"""
```

Out[20]:

```
'naga sai kota\n\nrepalle'
```

Example of single line of lline comments

def of python

python is a most popular programming language

- server to create the web application.
- to create the database connectivity to remote servers.
- it is most use full to system scripting.

''' naga sai'''

python data-types:

integer-int()

> it holdes the integervalues

string-str()

> it holdes the string values

FLOAT-float()

> IT Holdes the Floating type of data values

In [28]:

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

```
10
```

Out[28]:

```
int
```

In [29]:

```
b=4.5
print(b)
type(b)
```

```
4.5
```

Out[29]:

```
float
```

In [30]:

```
k="sai"
type(k)
```

Out[30]:

```
str
```

In [31]:

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

Out[31]:

```
str
```

In [34]:

```
# convert the integerto float
a=3455
m=float(a)
type(m)
```

Out[34]:

```
float
```

key words python

In [37]:

```
# 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']
```

In [38]:

```
# convert float to integer
a=34.6
b=int(a)
type(b)
```

Out[38]:

```
int
```

In [49]:

```
# convert int to string
a=33
b=str(a)
type(b)
```

Out[49]:

```
str
```

In [50]:

```
# convert float to string
a=39.5
b=str(a)
type(b)
```

Out[50]:

```
str
```

In [ ]:

```
Python keywords are special reserved words that have specific meanings and purposes and
    can't be used for anything but those specific purposes.
    These keywords are always available—you'll never have to import them into your code.
    Python keywords are different from Python's built-in functions and types.
```

In [73]:

```
# Python Program to calculate the square root
num=25

num_cuberoot = num **0.3
print(num_cuberoot)
```

```
2.626527804403767
```

Write a programme to find biggest of two numbers

In [107]:

```
a=22
b=67
if(a>b):
    print("a is big")
else:
    print("b is big")
```

```
b is big
```

In [159]:

```
# WRITE A PROGRAMME TO CHECK WHETHER GIVEN NUMBER EVEN OR NOT
num =2
if(num%2==0):
    print("num is even")
else:
        print("num is odd")
```

```
num is even
```

In [134]:

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

```
enter n1 value44
enter n2 value55
n2 is greater than n1
```

In [149]:

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

```
12+12
```

In [162]:

```
num =int(input("enter num value"))
if(num%2==0):
    print("num is even")
else:
        print("num is odd")
```

```
enter num value3
num is odd
```

In [166]:

```
age=23
if(age>18):
    print("eligible for vote")
else:
        print("not eligible for vote")
```

```
eligible for vote
```

In [167]:

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

```
enter age60
eligible for vote
```

To check the given character is vowel or constant

In [34]:

```
ch = str(input ("enter characters"))
if(ch=='a'or ch=='e'or ch=='i'or ch=='o'or ch=='u'):
    print(ch,"vowel")
else:
    print(ch,"consonant")
```

```
enter charactersa,e
a,e consonant
```

In [36]:

```
ch ='a'
if(ch=='a'or ch=='e'or ch=='i'or ch=='o'or ch=='u'):
    print(ch,"vowel")
else:
    print(ch,"consonant")
```

```
a vowel
```

find the biggest of three numbers

In [42]:

```
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 value23
enter n2 value33
enter n3 value44
n3 is greater than n1 and n2
```

Type *Markdown* and LaTeX: α^2

In [ ]:

```
<img src="mahesh.jpg" width=500 height=600>
```

In [48]:

```
<img src ="https://images.indulgexpress.com/uploads/user/imagelibrary/2020/1/25/original/Ma
```

```
  File "<ipython-input-48-ef1eb2722825>", line 1
    <img src ="https://images.indulgexpress.com/uploads/user/imagelibrary/20
20/1/25/original/MaheshBabuSourceInternet.jpg" width=500 height=600>
    ^
SyntaxError: invalid syntax
```

write a programme to print natural numbers

In [2]:

```
# to print natural 1 to 10 natural numbers
for i in range(1,11):
    print(i,end='')
```

```
12345678910
```

In [17]:

```
# to print 1 to 100 odd numbers
for i in range(0,101,3):
    print(i, end="")
```

```
036912151821242730333639424548515457606366697275788184879093969 9
```

In [13]:

```
# to print the  0 to 50 elements
for num in range(0,50,3):
    print(num,end='')
```

```
03691215182124273033363942454 8
```

In [1]:

```
# to print 1 to n natural numbers in ascending order
n=int(input("enter number"))
for i in range (1,n+1):
    print(i,end="")
```

```
enter number10
12345678910
```

In [19]:

```
# to print 1 to n natural numbers in descending orders
n=int(input("enter number value"))
for i in range (n,0,-1):
    print(i,end=" ")
```

```
enter number value22
22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1
```

In [8]:

```
# break statement example in python
for i in('apssdc'):
    if i=='s':
        break
    else:
            print(i,end="")
```

```
ap
```

In [9]:

```
for i in('123456'):
    if i=='4':
        break
    else:
            print(i,end="")
```

```
123
```

In [53]:

```
# to print even numbers between 1 to 20 using continue
for i in range(1,21):
    if(i%2!=0):
        continue
    else:
        print(i,end=" ")
```

```
2 4 6 8 10 12 14 16 18 20
```

In [52]:

```
# swapping of two numbers
a=20
b=23
c=34
temp=a
a=b
b=temp
print(a,b)
```

```
23 20
```

In [51]:

```
a=23
b=34
a,b=(b,a)
print(a,b)
```

```
34 23
```

In [54]:

```
# how to generate a  random number in python
import random
num = random.random()
print(num)
```

```
0.5819517267803912
```

In [58]:

```
import random
print(random.randint(0,8))
```

```
3
```

In [68]:

```
import string
print(string.ascii_uppercase)
```

```
ABCDEFGHIJKLMNOPQRSTUVWXYZ
```

In [69]:

```
import string
print(string.ascii_lowercase)
```

```
abcdefghijklmnopqrstuvwxyz
```

In [72]:

```
import string
for letter in string.ascii_lowercase:
    print(letter,end=" ")
for letter in string.ascii_uppercase:
    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 [79]:

```
# programme to display calender of the current year and month
import calendar
year=2022
month=9
date=23
print(calendar.month(year,month,date))
```

```
September 2022
        Monday                  Tuesday                Wednesday
Thursday                 Friday                 Saturday                 Sun
day

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 [80]:

```
import calendar
year=2022
month=9
print(calendar.month(2022,9,))
```

```
   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 [1]:

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

```
5
```

In [11]:

```
a=2
b=2
c=a+b
print(c)
```

```
4
```

In [7]:

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

```
enter a values6
enter b values7
```

Out[7]:

```
13
```

In [11]:

```
# with arguments and without returnvalues
n1=int(input("enter n1 values"))
n2=int(input("enter n2 values"))
def subtraction(a,b):
    c=a-b
    print(c)
subtraction(n1,n2)
```

```
enter n1 values44
enter n2 values23
21
```

In [16]:

```
a=5
b=5
c=a+b
print(c)
```

```
10
```

In [24]:

```
# with out arguments and with out values
# static
def adding():
    a=20
    b=10
    sum=a+b
    print("after calling:",sum)
adding()
```

```
after calling: 30
```

In [25]:

```
# with out argumetnsand with return values
def multiplication():
    a=12
    b=24
    multi=a*b
    return multi
print("after calling the multiplication:",multiplication())
```

```
after calling the multiplication: 288
```

In [26]:

```
a=int(input("enter a value"))
b=int(input("enter b value"))
def multiplication():
    a=12
    b=24
    c=a*b
    return multi
print("after calling the multiplication:",multiplication())
```

```
enter a value33
enter b value55
after calling the multiplication: 288
```

In [6]:

```
a=("enter a value")
b=("enter b value")
def mul():
    mul=a*b
    return multi
print (mul)
```

```
<function mul at 0x0000025EF5E77940>
```

LISTS

In [ ]:

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

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

In [2]:

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

Out[2]:

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

In [3]:

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

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

In [4]:

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

```
3
```

In [6]:

```
# accessing first element in alist
print(li[0])
```

```
apple
```

In [7]:

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

```
orange
```

In [11]:

```
li[0]="apple"
li
```

Out[11]:

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

In [12]:

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

Out[12]:

```
['apple', 'opps', 'banana', 'orange']
```

In [19]:

```
li1=["gopal","123","true"]
li1
```

Out[19]:

```
['gopal', '123', 'true']
```

In [21]:

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

Out[21]:

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

In [22]:

```
li[3:]
```

Out[22]:

```
['orange']
```

In [23]:

```
li[:5]
```

Out[23]:

```
['apple', 'opps', 'banana', 'orange']
```

In [27]:

```
li.remove("gopal")
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-27-55d0de087d66> in <module>
----> 1 li.remove("gopal")

ValueError: list.remove(x): x not in list
```

In [ ]:

```
li.remove("gopal")
```

In [28]:

```
li
```

Out[28]:

```
['opps', 'banana', 'orange']
```

In [32]:

```
li1=["sbi","axex",]
li+li1
```

Out[32]:

```
['opps', 'banana', 'orange', 'sbi', 'axex']
```

In [ ]:

```
li.clear
```

In [29]:

```
li
```

Out[29]:

```
['opps', 'banana', 'orange']
```

In [33]:

```
li1
```

Out[33]:

```
['sbi', 'axex']
```

In [34]:

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

```
opps banana orange
```

Tuples

it is collection of deifferent of data

it is immutable(can't change)

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

to create the empty tuple

tuple_name=()

to create single values

tuple_name=()

to create multiple values

tuple_name=(value1,value2....)

In [48]:

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

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

In [50]:

```
# single value tuple
t2=(10)
print(type(t2))
t3=(23,)
print(type(t3))
```

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

In [51]:

```
t2
```

Out[51]:

```
10
```

In [52]:

```
t3
```

Out[52]:

```
(23,)
```

In [54]:

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

t1
print(t1[2])
```

```
30
```

In [56]:

```
# how to access the values from the tuple
print(t1[0:1])
```

```
(10,)
```

In [47]:

```
naga=("sai", "vijay", "arjun")
print(naga)
```

```
('sai', 'vijay', 'arjun')
```

In [ ]:

```

```

In [46]:

```
naga=("sai", "vijay", "arjun")
print(len(naga))
```

```
3
```

In [45]:

```
naga=("naga", 34, "arjun", 34)
print(naga)
```

```
('naga', 34, 'arjun', 34)
```

In [44]:

```
naga=("naga","arjun")
naga
print(type(naga))
```

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

In [41]:

```
# TUPLE CONSTRUCTOR TO CREATE TOUPLE
tuple=tuple(("naga", "sai"))
print(tuple)
```

```
('naga', 'sai')
```

In [58]:

```
# count the numbers of courances
t2=(12,23,12,12,66,45,66,324)
t2.count(12)
```

Out[58]:

```
3
```

In [ ]:

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

In [ ]:

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

In [ ]:

```
to create the dictionaries values:
    dictionaires_name=(key:value,key:value2....)
```

In [1]:

```
# 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 [4]:

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

```
{'a': 100, 'name': 'anusha', 'branch': 'cse', 'b': 45.8}
```

In [5]:

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

```
anusha
45.8
100
```

In [9]:

```
# update dictionary values
print(d2)
d2['branch']='EEE'
print(d2)
```

```
{'a': 100, 'name': 'anusha', 'branch': 'EEE', 'b': 45.8}
{'a': 100, 'name': 'anusha', 'branch': 'EEE', 'b': 45.8}
```

In [11]:

```
# dictionary key words in python
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 [ ]:

```
# git is a local system
# github is  a remote system
```