```
#!/usr/bin/env python
# coding: utf-8

# In[1]:


print("Python is fun")


# In[2]:


print("Print is here")


# In[3]:


print("KAPIL","PYTHON","2020")


# In[4]:


print("Hello","World",sep='-')
print("Hi","Jaipur",sep='$')


# In[5]:


print("Hello","World",sep='-',end='_')
print("Hi","Jaipur",sep='$',end='&')
print("Regex")


# In[6]:


#a = 10, b = 20 okay in C/C++
a,b = 10,20 #This is right way in python code....
print(a,b)
a,b = b,a #code for swapping the value
print(a,b)


# In[7]:


s1 = 'Python'
print(s1[0])
print(s1[-6])


# In[8]:
```

```
s1 = """This is
my code"""
print(s1)


# In[9]:


"""
This is simple code
for swapping numbers
"""
a,b = 10,20
a,b = b,a
print(a,b)


# In[10]:


a = 1
print(a,id(a))
a = a + 1
print(a,id(a))


# In[11]:


a = 10
b = 10
c = 10
print(a,id(a),b,id(b),c,id(c))
c = 11
b = 12
print(a,id(a),b,id(b),c,id(c))
c = c - 1
print(a,id(a),b,id(b),c,id(c))


# In[12]:


a = 10
b = 10
c = 10
print(a,id(a),b,id(b),c,id(c))
#value 10 has 3 references
del a
del b
#value 10 has 1 reference that is c
print(c,id(c))
del c
#value 10 has no references now memory is eligible for garbage
collection
```

```
# In[13]:


print(9/5)
print(9//5)


# In[14]:


print(2**3) #2 raised to the power 3


# In[ ]:


print(2**3**2)
```