Print Hello World!

In [4]:
```
print("Hello World!")
```

```
Hello World!
```

Variables

In [5]:
```
a=10
b=20
a+b
print(a+b)
```

```
30
```

String

In [9]:
```
# double quote
s="arya"
print(s)

# single quote
s1='arya'
print(s1)

# triple quote
s2='''
a
ry
a'''
print(s2)

# advantage with triple quote is we can print in diff lines without explicitly mentioning it
```

```
arya
arya

a
ry
a
```

String Indexing in 'python'

In [14]:
```
s="arya"
print(s[0])
print(s[1])
print(s[2])
print(s[3])

print()

print(s[-1])
print(s[-2])
print(s[-3])
print(s[-4])
```

```
a
r
y
a

a
y
r
a
```

In [16]:
```
s='''My
Name
Is
Arya'''

# print(s)

s
```

Out[16]:
```
'My\nName\nIs\nArya'
```

How Strings are stored

In [20]:
```
s="arya"
a="arya"

print(id(s))
print(id(a)) # addresses where it stores "arya" are same in case of a and s both
```

```
2335726790768
2335726790768
```

Immutability of strings in Python

strings are immutable in Python, we can retrieve a char from a string but cannot modify that position of the string

Functions on strings

split

In [25]:
```
str="my name is arya"

li=str.split() # automatically splits on the basis of space if no arguments passed
print(li)

str1="my,name,is,arya"
li1=str1.split(',')
print(li1)

li2=str1.split(',', 1) # the other argument allows us to split the pecified number of times
print(li2)
```

```
['my', 'name', 'is', 'arya']
['my', 'name', 'is', 'arya']
['my', 'name,is,arya']
```

replace

In [34]:
```
str="my name is arya"
str1 = str.replace("arya", "tito")

print(str) # will not make any change in original string (due to immutability principle)

print(str1)

str2 = str.replace("arjo", "tito") # no "arjo" present
print(str2)

strx="my name is arya arya arya"
strx1 = strx.replace("arya", "tito")
print(strx1)

strx2 = strx.replace("arya", "tito", 2) # specifying the no. of times we want to change
print(strx2)
```

```
my name is arya
my name is tito
my name is arya
my name is tito tito tito
my name is tito tito arya
```

find

In [42]:
```
str="my name is arya"
idx = str.find("na");
print(idx) # returns the start index if substring is present

idx1 = str.find("nae") # else returns -1
print(idx1)

str1="my name is arya arya arya"
idx2 = str.find("arya")
print(idx2)

idx3 = str1.find("arya", 12, 20) # from 12 to 20 if "arya" is present
print(idx3)
```

```
3
-1
11
16
```

lower and upper

In [47]:
```
str="my name is arya"
str1=str.lower();
str2=str.upper();

print(str1)
print(str2)
```

```
my name is arya
MY NAME IS ARYA
```

starts-with

In [52]:
```
str="my name is arya"
ans=str.startswith("my", 3, 10)
ans1=str.startswith("my")

print(ans)
print(ans1)
```

```
False
True
```