String Replace

In [1]:
```
say="Helloo world"
```

In [2]:
```
say.replace('Helloo','hello')
```

Out[2]:
```
'hello world'
```

String Split

In [1]:
```
say='one two three four'
```

In [3]:
```
say.split()
```

Out[3]:
```
['one', 'two', 'three', 'four']
```

Variable Assignment

Single Variable assignment

Example

In [4]:
```
a=10
print(a)
```

```
10
```

Multi Variable assignment

Example

In [5]:
```
a,b,c=2,2.5,'hello'
print(type(a))
print(type(b))
print(type(c))
```

```
<class 'int'>
<class 'float'>
<class 'str'>
```

Operator

Arithmetic operators,

Assignment operators,

Comparison operators,

Logical operators,

Identity operators,

Membership operators,

Bitwise operators.

Arithmetic

Addition,

Subtraction,

Multiplication,

Division,

Floor Division,

Modulus,

Exponent

Example:

In [17]:
```
a=10+5    #addition
b=10-5    #subtraction
c=10*5    #multiplication
d=10/5    #normal division
e=12//5   #quotient
f=12%4    #remainder
g=3**4    #to the power of
```

In [18]:
```
print(a)
print(b)
print(c)
print(d)
print(e)
print(f)
print(g)
```

```
15
5
50
2.0
2
0
81
```

In [19]:
```
print(type(a))
print(type(b))
print(type(c))
print(type(d))
print(type(e))
print(type(f))
print(type(g))
```

```
<class 'int'>
<class 'int'>
<class 'int'>
<class 'float'>
<class 'int'>
<class 'int'>
<class 'int'>
```

Assignment

In [20]:
```
a=5
a+=3
print(a)
```

```
8
```

In [21]:
```
b=15
b-=10
print(b)
```

```
5
```

In [23]:
```
c=15
c-=20
print(c)
```

```
-5
```

Comparison/Relational

= is equal

! is not equal

.> is greater than

< is lesser than

<= is lesser than equal to

.>= is greater than equal to

Example

In [3]:
```
x=10
y=20
```

In [4]:
```
print(x==y)
print(x<y)
print(x>y)
```

```
False
True
False
```

Logical

and, or, not

Reffer Truth Table

In [5]:
```
print((x<20)and(y>10))
```

```
True
```

Identify

In [12]:
```
print((10) is (10.0))     #code will run by checking both type and value
```

```
False

<>:1: SyntaxWarning: "is" with a literal. Did you mean "=="?
<>:1: SyntaxWarning: "is" with a literal. Did you mean "=="?
C:\Users\user\AppData\Local\Temp\ipykernel_11688\1101379297.py:1: SyntaxWarni
ng: "is" with a literal. Did you mean "=="?
  print((10) is (10.0))     #code will run by checking both type and value
```

why python is an interpreted language

In [13]:
```
print('hello')
a=2
if(a<5):
    print("python")
print('welcome')
b=4
if(b**2):
    print('home')
```

```
hello
python
welcome
home
```

Type casting

Auto type casting,

Forced type casting.

In [14]:
```
a='2'
b=2.2
c=2
```

In [15]:
```
print(type(a))
print(type(b))
print(type(c))
```

```
<class 'str'>
<class 'float'>
<class 'int'>
```

In [17]:
```
4+4.4+False    #false value is zero
```

Out[17]:
```
8.4
```

In [18]:
```
4+4.4+True    #true value is one
```

Out[18]:
```
9.4
```

String strip

In [19]:
```
x=('***python***')
```

In [20]:
```
x.strip('*')
```

Out[20]:
```
'python'
```

In [21]:
```
x='    hello world    '
```

In [22]:
```
x.strip(' ')
```

Out[22]:
```
'hello world'
```

Indexing diff

In [23]:
```
say='welcome to india'
```

In [24]:
```
say[1:11:3]
```

Out[24]:
```
'eo  '
```

In [ ]:
```

```