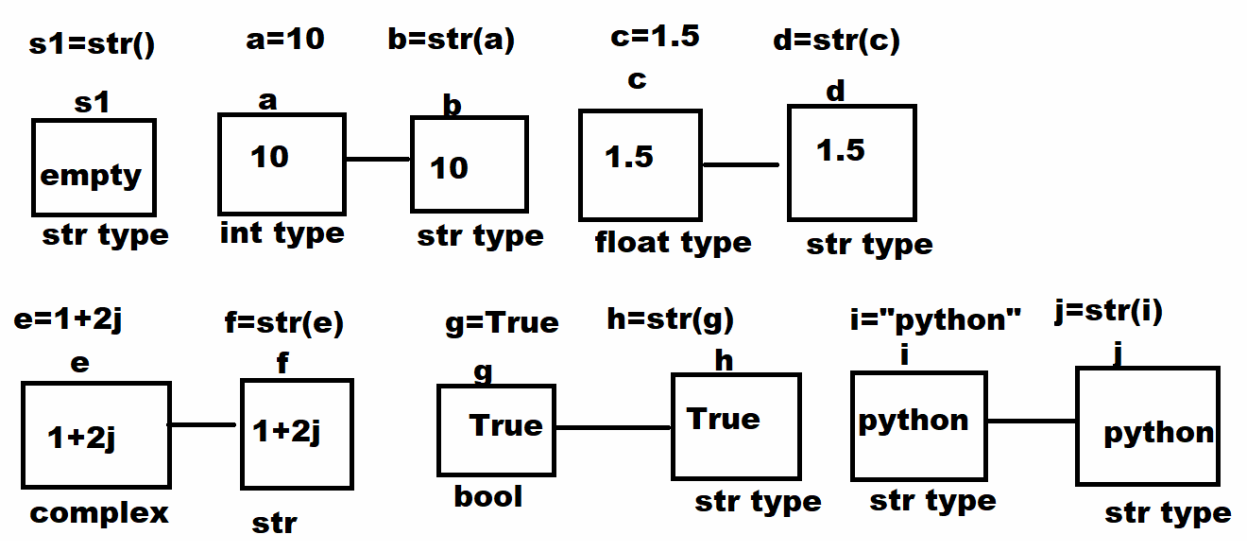
str() function

This function is used to perform the following conversions

1. Str to str
2. Int to str
3. Float to str
4. Complex to str
5. Bool to string

Syntax: str([value])



Example

```
a=str()
print(a,type(a))
b=str(10)
print(b,type(b))
c=str(1.5)
print(c,type(c))
d=str(1+2j)
print(d,type(d))
e=str(True)
print(e,type(e))
f=str("PYTHON")
print(f,type(f))
```

Output

```
<class 'str'>
10 <class 'str'>
1.5 <class 'str'>
(1+2j) <class 'str'>
True <class 'str'>
PYTHON <class 'str'>
```

Operators

What is operator?

Operator is special symbol, which is used to perform some operations.

Based on the operands on which it performs operation, operators are classified into 3 categories.

1. Unary Operators
2. Binary Operators
3. Ternary Operators

An operator required one operand to perform operation is called unary operator.
An operator required two operands to perform operation is called binary operator.
An operator required three operands to perform operation is called ternary operator.

Based on the operations, the operators are divided into different types.

1. Arithmetic Operators
2. Relational Operators
3. Logical Operators
4. Assignment Operators
5. Membership Operators
6. Identity Operators
7. Bitwise Operators
8. Conditional Operators
9. Walrus Operators

Arithmetic Operators

These operators are used to perform arithmetic operations.

+	Addition
-	Subtraction
*	Multiplication
/	Float division
//	Floor Division
%	Modulo
**	Power Of/Exponent

+ Operator in python is used to perform two operations.

1. Addition
2. Concatenation

If two operands are numbers, + operator performs addition
If two operands are sequences, + operator perform concatenation

Complex>float>int>bool

Int+int → int
Int+float → float
Int+float+complex → complex
Int+bool → int
Bool+float → float
Int+float+complex+bool → complex

Example:

```
a=10+20
b=1+2.5
c=1+2j+5
d=1+2j+3j
e=True+True
f=True+100
g=True-100
h=100+False
print(a,b,c,d,e,f,g,h)

s1="PYTHON"
```

```
s2="PROGRAMMING"
s3=s1+s2
print(s1,s2,s3,sep="\n")

s4="python"+str(3.12)
print(s4)
```

Output

```
30 3.5 (6+2j) (1+5j) 2 101 -99 100
PYTHON
PROGRAMMING
PYTHONPROGRAMMING
python3.12
```

-subtraction operator

This operator is used to the difference between two values, this operator is applied only on numeric type

Example:

```
a=10
b=5
c=a-b
f1=1.5
f2=1.3
f3=f1-f2
c1=1+2j
c2=1+1j
c3=c1-c2
print(a,b,c)
print(f1,f2,f3)
print(c1,c2,c3)
b1=True
b2=False
b3=b1-b2
print(b1,b2,b3)
b4=b2-b1
print(b4)
```

Output

```
10 5 5
```

1.5 1.3 0.19999999999999996
(1+2j) (1+1j) 1j
True False 1
-1

Example

```
# Write a program to swap two numbers without using 3rd variable


num1=int(input("Enter first number :"))
num2=int(input("Enter second number :"))


# Method-1 using third variable
num3=num1
num1=num2
num2=num3

print(num1,num2)

# Method-2 without using third variable

num1=num1+num2
num2=num1-num2
num1=num1-num2

print(num1,num2)

# Method-3 without using arithmetic operators
num1,num2=num2,num1
print(num1,num2)
```

Output

```
Enter first number :10
Enter second number :20
20 10
10 20
20 10
```

***In python this operator is used to perform two operations

1. Multiplying numbers
2. Repeating sequences

Example:

```
a=5
b=3
c=a*b
print(a,b,c)
f1=1.5
f2=1.4
f3=f1*f2
print(f1,f2,f3)
b1=True
b2=False
b3=b1*b2
print(b1,b2,b3)
```

Output

```
5 3 15
1.5 1.4 2.0999999999999996
True False 0
```

Example

```
s1="$"*50
print(s1)
s2="-"*30
print(s2)
list1=[0]*10
print(list1)
list2=[1]*20
print(list2)
s3="PYTHON"*5
print(s3)
s4=int("5")*int("6")
print(s4)
s5=5*"PYTHON"
print(s5)
s6="PY"+"JY"*3
print(s6)
```

Output

```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
------------------------------
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
PYTHONPYTHONPYTHONPYTHONPYTHON
30
PYTHONPYTHONPYTHONPYTHONPYTHON
PYJYJYJY
```

Example

```
# Write a program to find area of rectangle
# area=l*b

l=float(input("Enter L Value "))
b=float(input("Enter B Value "))
area=l*b
print(area)
```

Output

```
Enter L Value 1.5
Enter B Value 2.0
3.0
```