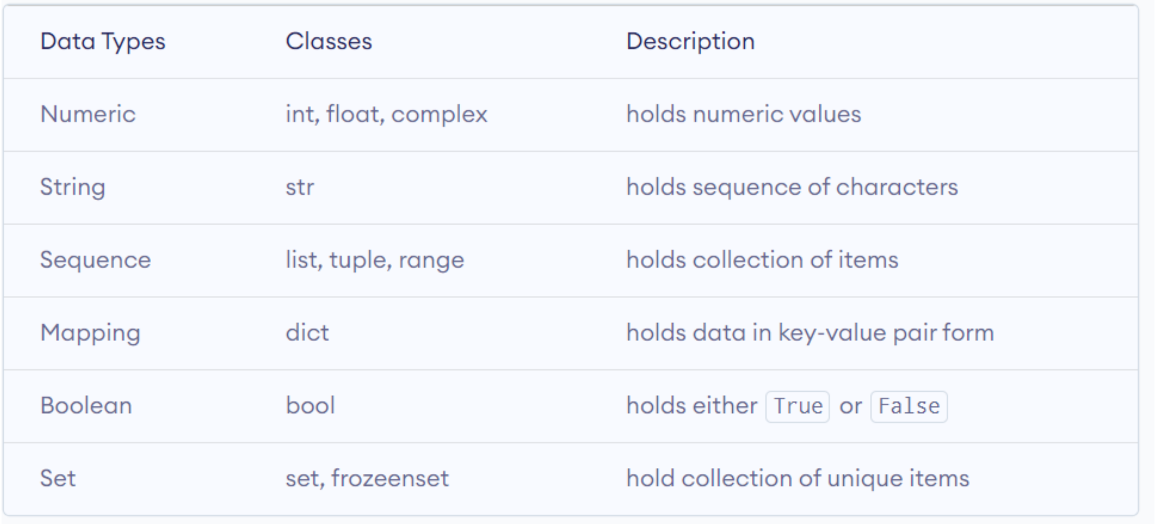
LECTURE

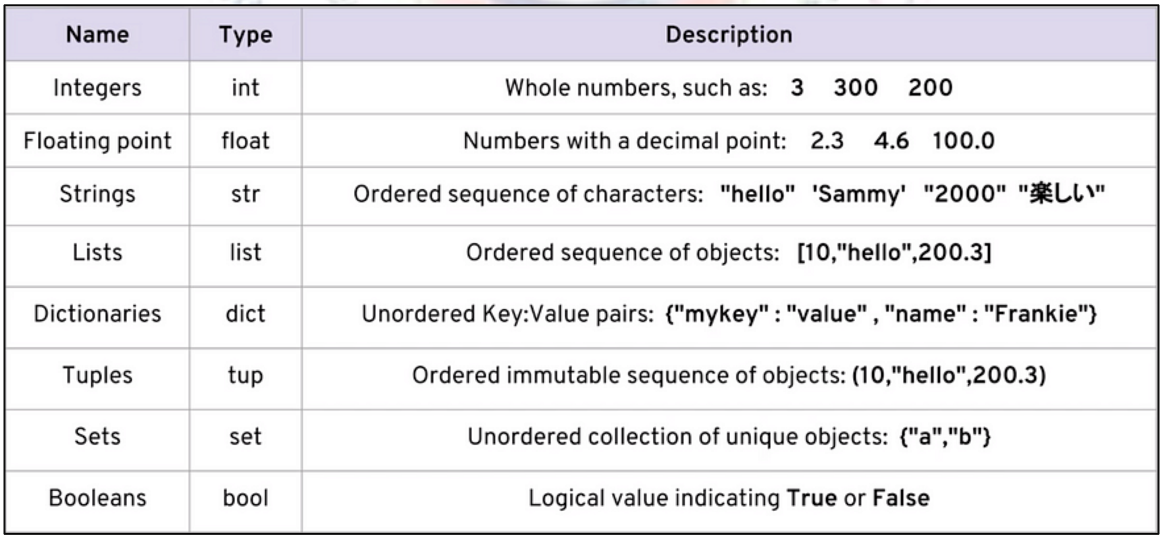
FIVE

Data types in Python

Every value in Python has a datatype, data type is an important concept. Variables can store data of different types, and different types can do different things. Python has the following data types built-in by default, in these categories:



Data Types	Classes	Description
Numeric	int, float, complex	holds numeric values
String	str	holds sequence of characters
Sequence	list, tuple, range	holds collection of items
Mapping	dict	holds data in key-value pair form
Boolean	bool	holds either `True` or `False`
Set	set, frozeenset	hold collection of unique items



Name	Type	Description
Integers	int	Whole numbers, such as: **3 300 200**
Floating point	float	Numbers with a decimal point: **2.3 4.6 100.0**
Strings	str	Ordered sequence of characters: **"hello" 'Sammy' "2000" "楽しい"**
Lists	list	Ordered sequence of objects: **[10,"hello",200.3]**
Dictionaries	dict	Unordered Key:Value pairs: **{"mykey" : "value" , "name" : "Frankie"}**
Tuples	tup	Ordered immutable sequence of objects: **(10,"hello",200.3)**
Sets	set	Unordered collection of unique objects: **{"a","b"}**
Booleans	bool	Logical value indicating **True** or **False**

Python Numeric Data type

In Python, numeric data type is used to hold numeric values:

- Integers can be of any length, it is only limited by the memory available.
- A floating point number is accurate up to 15 decimal places.
- Complex numbers are written in the form, x + yj, where x is the real part and y is the imaginary part. Here are some examples.

type() function in Python

We can use the **type() function** to know which class a variable or a value belongs to. The type() function is mostly used for debugging purposes.

Example: Python type() function for Numeric Data type

```
num1 = 5
print(num1, 'is of type', type(num1))
num2 = 2.0
print(num2, 'is of type', type(num2))
num3 = 1+2j
print(num3, 'is of type', type(num3))
```

Output

```
5 is of type <class 'int'>
2.0 is of type <class 'float'>
(1+2j) is of type <class 'complex'>
```

Python type Casting

There may be times when you want to specify a type on to a variable. This can be done with casting.

Casting in python is therefore done using constructor functions:

- int() - constructs an integer number from an integer literal, a float literal (by rounding down to the previous whole number), or a string literal (providing the string represents a whole number)
- float() - constructs a float number from an integer literal, a float literal or a string literal (providing the string represents a float or an integer)
- str() - constructs a string from a wide variety of data types, including strings, integer literals and float literals

Example: Python Casting for int

```
x = int(1)        # x will be 1
y = int(2.8)      # y will be 2
z = int("3")      # z will be 3
```

Example: Python Casting for float

```
x = float(1)         # x will be 1.0
y = float(2.8)       # y will be 2.8
z = float("3")       # z will be 3.0
w = float("4.2")     # w will be 4.2
```

Example: Python Casting for str

```
x = str("s1")      # x will be 's1'
y = str(2)         # y will be '2'
z = str(3.0)       # z will be '3.0'
```

Type Conversion

The process of converting the value of one data to another data type is called type conversion. Python has two types of type conversion:

1. **Implicit Type Conversion:** Python automatically converts one data type to another data type.

Example: Python Implicit Type Conversion

```
a = 123
b = 1.23
c = a + b
print(c)
print(type(c))
a = 123
b = "5"
c = a + b
print(c)
print(type(c))
```

Output

```
124.23
<class 'float'>
Traceback (most recent call last):
TypeError: unsupported operand type(s) for +: 'int' and 'str'
```

2. **Explicit Type Conversion(Type Casting)**: using to convert the data type of an object to required data type. We use the predefined functions like int(), float(), str(), etc to perform explicit type conversion.

Example: Python Implicit Type Conversion

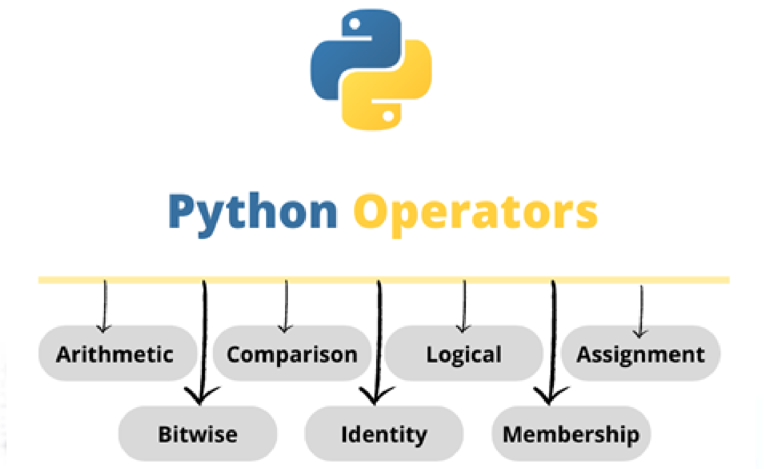
```
a = 123
b = int("5")
c = a + b
print(c)
print(type(c))
```

Output

```
128
<class int>
```

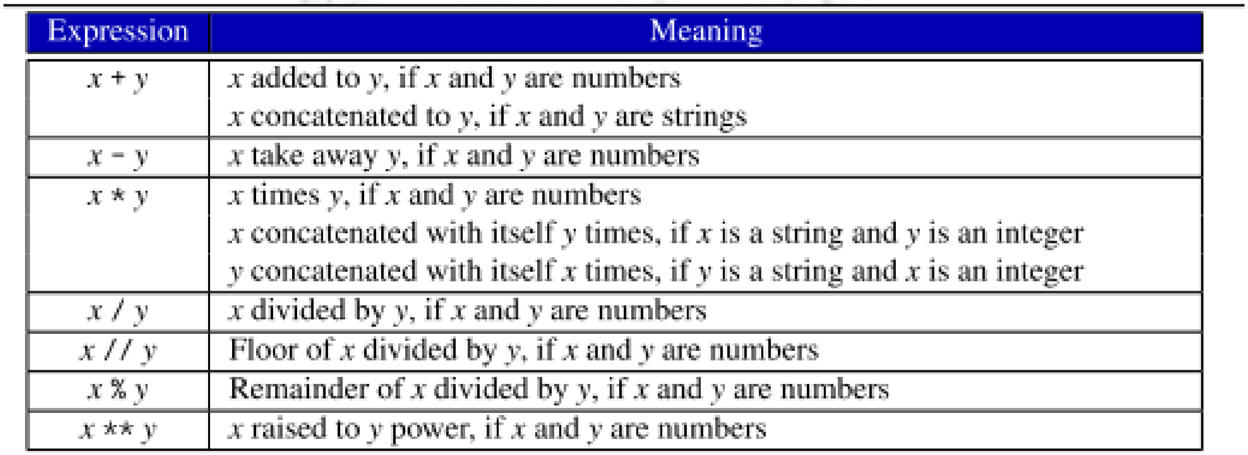
Operators in python

Operators in Python are special symbols used to perform operations on values or variables. Arithmetic, Comparison, Logical, and Assignment operators are the most commonly used. However, there are other types as well. The value that the operator operates on is called the operand.

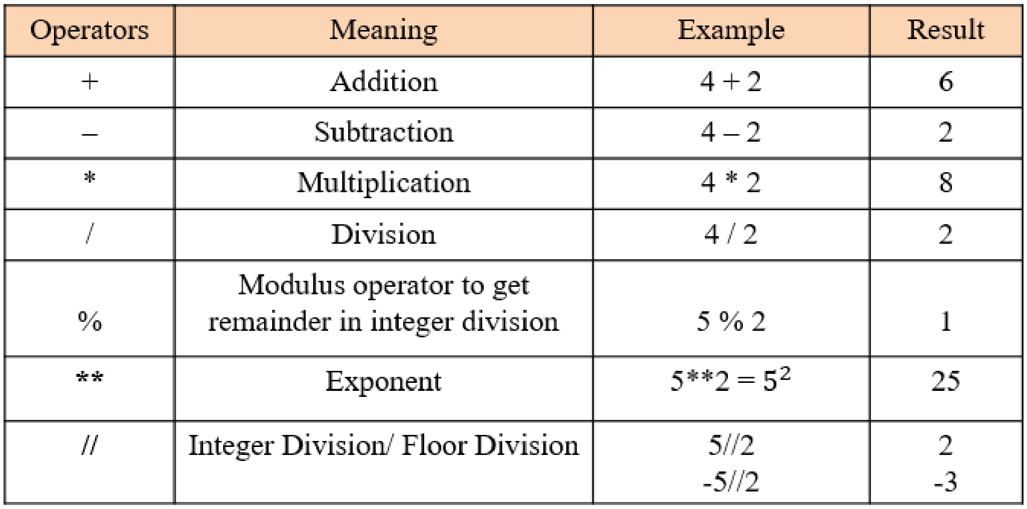


Arithmetic operators

Arithmetic operators are used to perform mathematical operations like addition, subtraction, multiplication, etc.



Expression	Meaning
x + y	*x* added to y, if *x* and y are numbers *x* concatenated to y, if *x* and y are strings
x - y	*x* take away y, if *x* and y are numbers
x * y	*x* times y, if *x* and y are numbers *x* concatenated with itself y times, if *x* is a string and y is an integer y concatenated with itself *x* times, if y is a string and *x* is an integer
x / y	*x* divided by y, if *x* and y are numbers
x // y	Floor of *x* divided by y, if *x* and y are numbers
x % y	Remainder of *x* divided by y, if *x* and y are numbers
x ** y	*x* raised to y power, if *x* and y are numbers



Operators	Meaning	Example	Result
+	Addition	4 + 2	6
–	Subtraction	4 – 2	2
*	Multiplication	4 * 2	8
/	Division	4 / 2	2
%	Modulus operator to get remainder in integer division	5 % 2	1
**	Exponent	5**2 = 5^2	25
//	Integer Division/ Floor Division	5//2 -5//2	2 -3

Example: Arithmetic operators

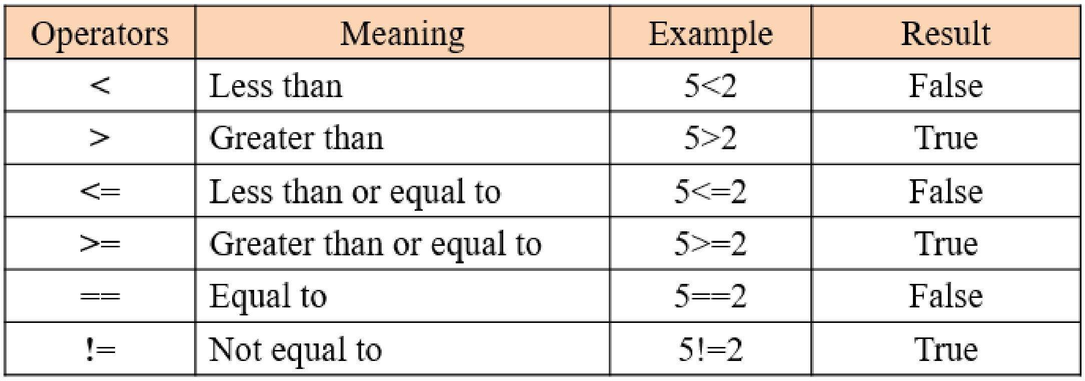
```
x = 10
y = 4
print('x + y =', x+y)
print('x - y =', x-y)
print('x * y =', x*y)
print('x / y =', x/y)
print('x // y =', x//y)
print('x ** y =', x**y)
```

Output

```
x + y = 14
x - y = 6
x * y = 40
x / y = 2.5
x // y = 2
x ** y = 10000
```

Comparison operators

Comparison operators are used to compare values. It either returns True or False according to the condition.



Operators	Meaning	Example	Result
<	Less than	5<2	False
>	Greater than	5>2	True
<=	Less than or equal to	5<=2	False
>=	Greater than or equal to	5>=2	True
==	Equal to	5==2	False
!=	Not equal to	5!=2	True

Example: Comparison operators

```
x = 10
y = 12
print('x > y  is',x>y)
print('x < y  is',x<y)
print('x == y is',x==y)
print('x != y is',x!=y)
print('x >= y is',x>=y)
print('x <= y is',x<=y)
```

Output

```
x > y  is False
x < y  is True
x == y is False
x != y is True
x >= y is False
x <= y is True
```

Assignment operators

Assignment operators are used in Python to assign values to variables.

a = 5 is a simple assignment operator that assigns the value 5 on the right to the variable a on the left.

There are various compound operators in Python like a += 5 that adds to the variable and later assigns the same. It is equivalent to a = a + 5.

Operator	Example	Equivatent to
=	x = 5	x = 5
+=	x += 5	x = x + 5
-=	x -= 5	x = x - 5
*=	x *= 5	x = x * 5
/=	x /= 5	x = x / 5
%=	x %= 5	x = x % 5
//=	x //= 5	x = x // 5
**=	x **= 5	x = x ** 5
&=	x &= 5	x = x & 5
|=	x |= 5	x = x | 5
^=	x ^= 5	x = x ^ 5
>>=	x >>= 5	x = x >> 5
<<=	x <<= 5	x = x << 5

Examples of Python Program

Example (1): Write Python Program for input two number and output their addition (summation).

```
x = int(input('Enter a number 1 : '))
y = int(input('Enter a number 2 : '))
z=x+y
print(z)
```

Example (2): Write Python Program for input three number and output their average.

```
a = int(input('Enter a number 1 : '))
b = int(input('Enter a number 2 : '))
c= int(input('Enter a number 3 : '))
av=(a+b+c) / 3
print(av)
```

Example (3): Write Python Program for input two number and computes the arithmetic operators.

```
a = int(input('Enter a number 1 : '))
b = int(input('Enter a number 2 : '))
sum = a + b
sub = a-b
mul = a * b
div = a / b
mod = a % b
print("sum= ",sum)
print("sub = ", sub)
print("mul = ", mul)
print("div = ", div)
print("mod = ", mod)
```

Example (4): Write Python Program to read length in feet and convert to centimeter.

```
feet = int(input("length in feet: "))
centimeter = feet *30
print("centimeter = ", centimeter)
```

Example (5): Write Python Program to read temperature in Fahrenheit and convert to Celsius.

```
Fahrenheit = int(input('temperature in Fahrenheit: '))
Celsius = (5/9) * (Fahrenheit-32)
print("Celsius = ", Celsius)
```

Example (6): Write Python Program to read length of slide and compute area and circumference (perimeter) of square.

```
slide = int(input('length of slide: '))
area = slide * slide
perimeter = 4 * slide
print("area of square = ", area)
print("perimeter of square = ", perimeter)
```