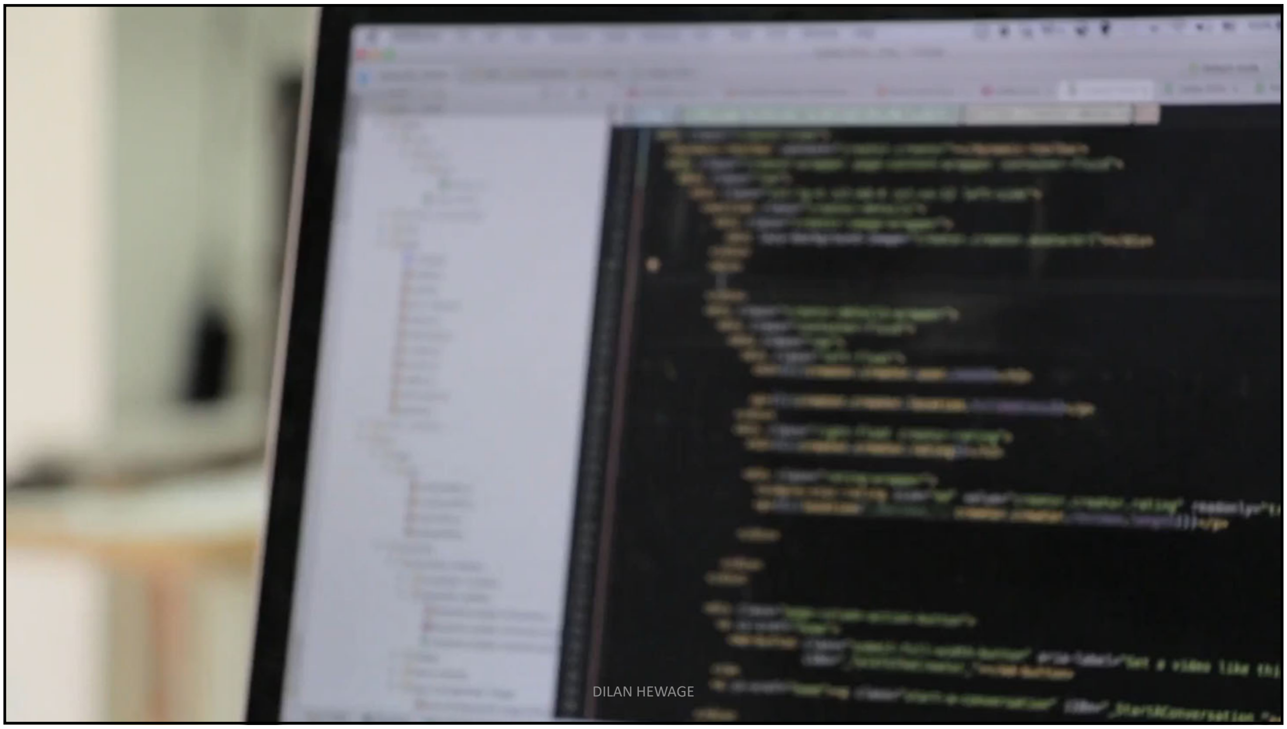


1

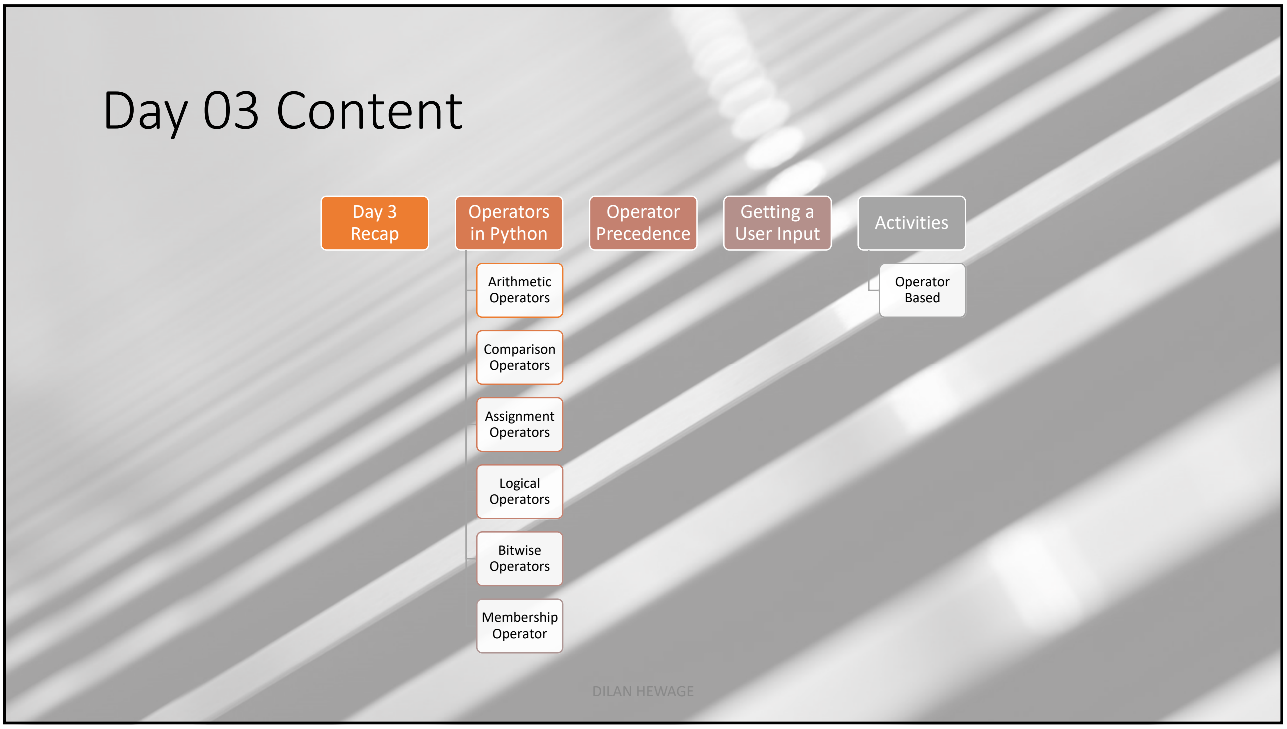
ඇල්ගොරිදම සංවර්ධනය සහ පයිතන් ක්‍රමලේඛනය

Algorithm Development & Python Programming

Workshop Day - 4

DILAN HEWAGE
BIT – University of Colombo / CIDM – APIDM
Founder of WiTeds Solutions
Co-Founder of Print Arcade

2



Day 03 Content

- Day 3 Recap
- Operators in Python
  - Arithmetic Operators
  - Comparison Operators
  - Assignment Operators
  - Logical Operators
  - Bitwise Operators
  - Membership Operator
- Operator Precedence
- Getting a User Input
- Activities
  - Operator Based

DILAN HEWAGE

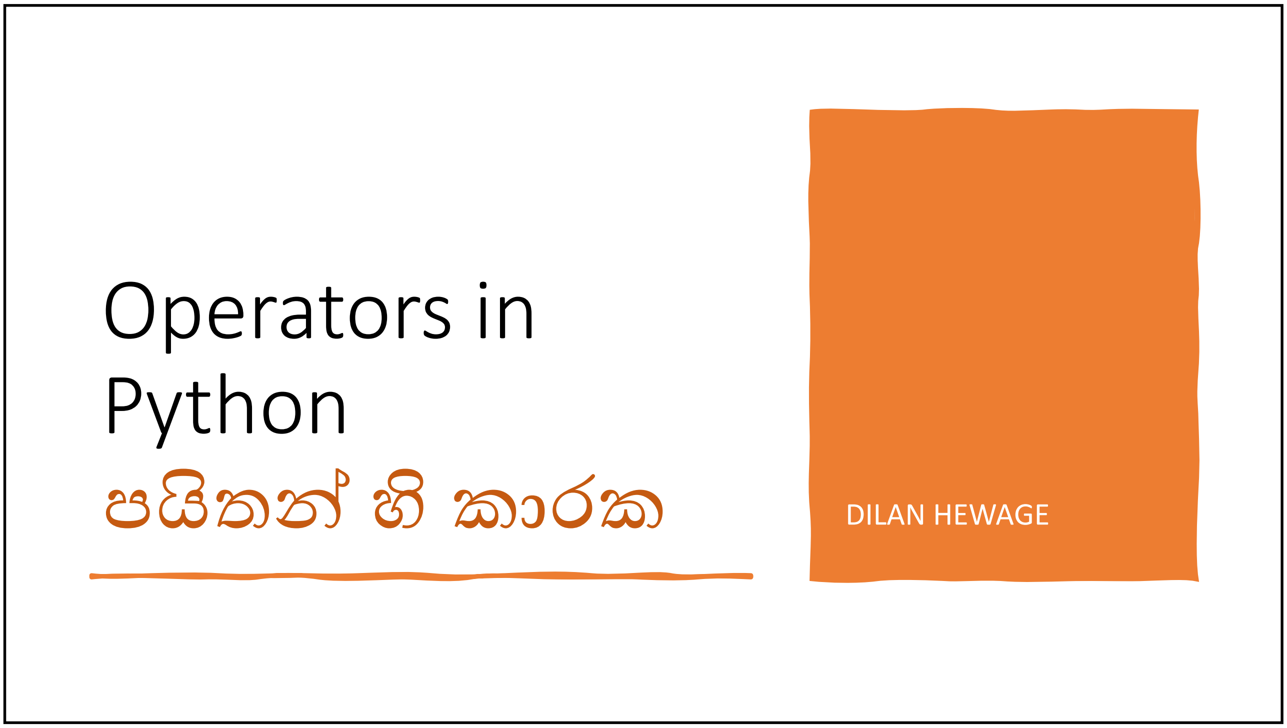
3

Day 3 Recap

3 දින සාරාංශය

DILAN HEWAGE

4

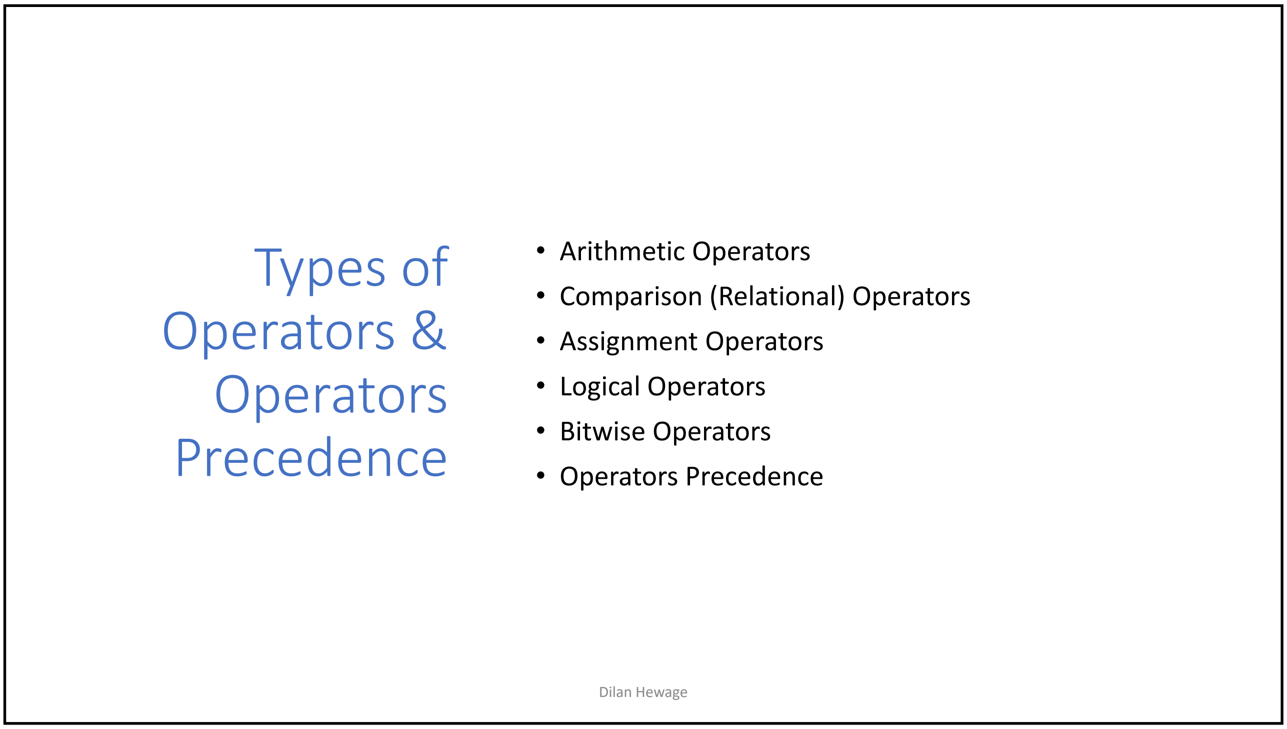


Operators in Python

පයිතන් හි කාරක

DILAN HEWAGE

5

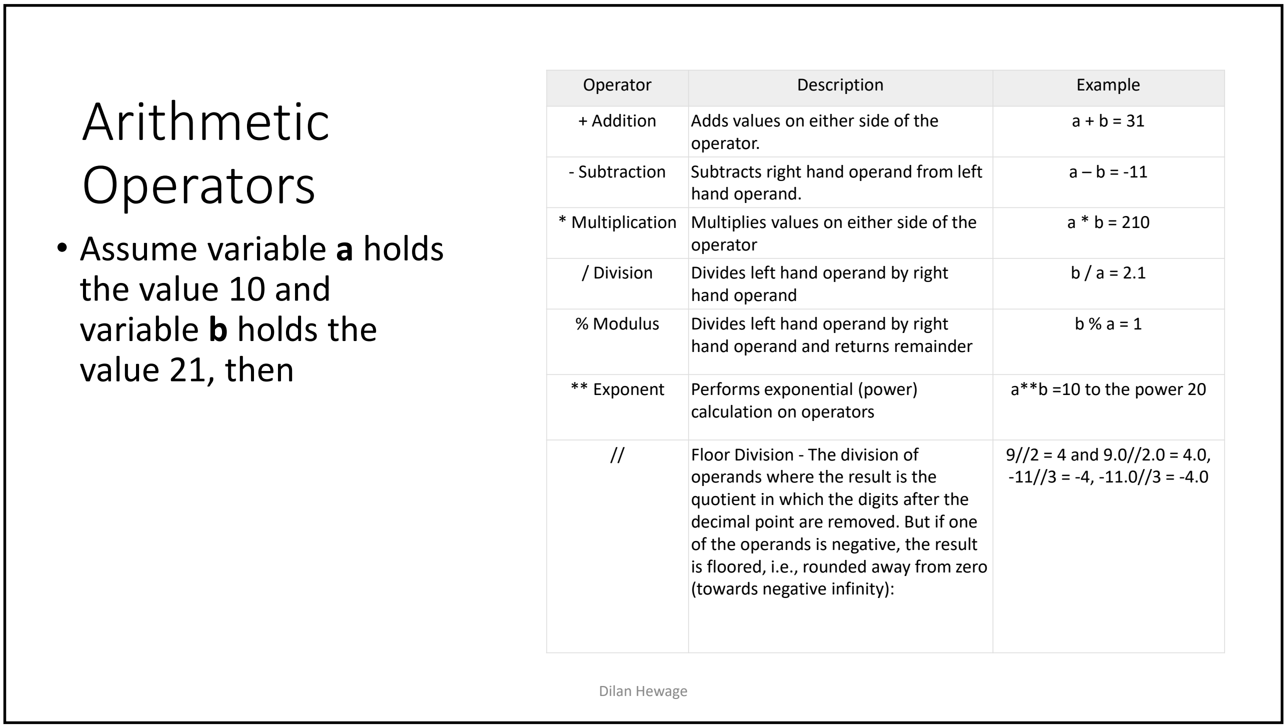


Types of Operators & Operators Precedence

- Arithmetic Operators
- Comparison (Relational) Operators
- Assignment Operators
- Logical Operators
- Bitwise Operators
- Operators Precedence

Dilan Hewage

6



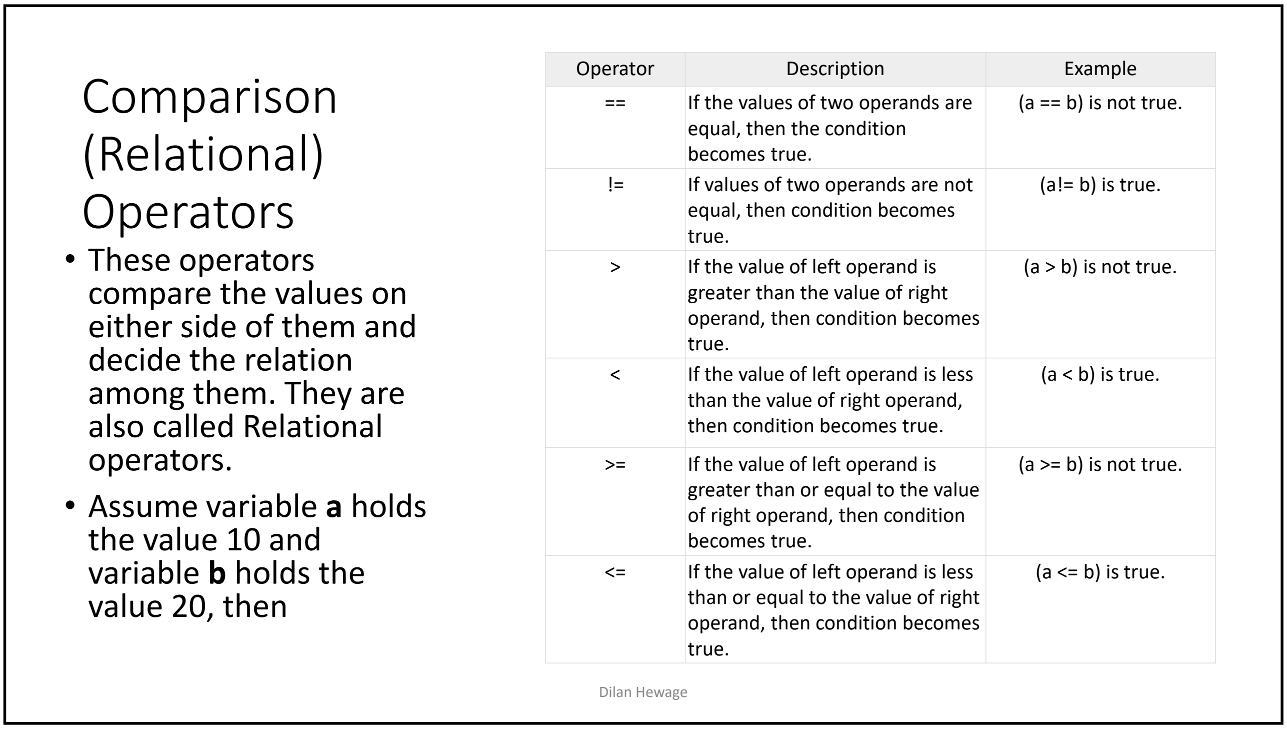
Arithmetic Operators

- Assume variable **a** holds the value 10 and variable **b** holds the value 21, then

Operator	Description	Example
+ Addition	Adds values on either side of the operator.	a + b = 31
- Subtraction	Subtracts right hand operand from left hand operand.	a – b = -11
* Multiplication	Multiplies values on either side of the operator	a * b = 210
/ Division	Divides left hand operand by right hand operand	b / a = 2.1
% Modulus	Divides left hand operand by right hand operand and returns remainder	b % a = 1
** Exponent	Performs exponential (power) calculation on operators	a**b =10 to the power 20
//	Floor Division - The division of operands where the result is the quotient in which the digits after the decimal point are removed. But if one of the operands is negative, the result is floored, i.e., rounded away from zero (towards negative infinity):	9//2 = 4 and 9.0//2.0 = 4.0, -11//3 = -4, -11.0//3 = -4.0

Dilan Hewage

7



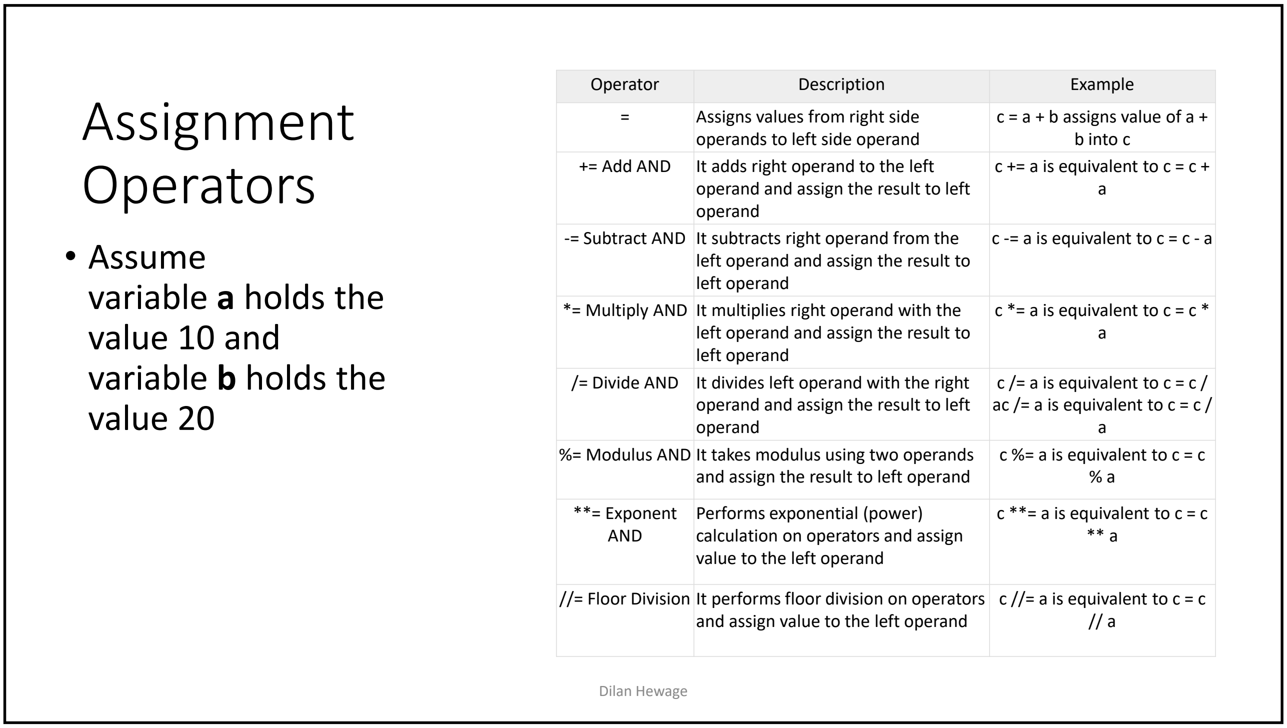
Comparison (Relational) Operators

- These operators compare the values on either side of them and decide the relation among them. They are also called Relational operators.
- Assume variable **a** holds the value 10 and variable **b** holds the value 20, then

Operator	Description	Example
==	If the values of two operands are equal, then the condition becomes true.	(a == b) is not true.
!=	If values of two operands are not equal, then condition becomes true.	(a!= b) is true.
>	If the value of left operand is greater than the value of right operand, then condition becomes true.	(a > b) is not true.
<	If the value of left operand is less than the value of right operand, then condition becomes true.	(a < b) is true.
>=	If the value of left operand is greater than or equal to the value of right operand, then condition becomes true.	(a >= b) is not true.
<=	If the value of left operand is less than or equal to the value of right operand, then condition becomes true.	(a <= b) is true.

Dilan Hewage

8



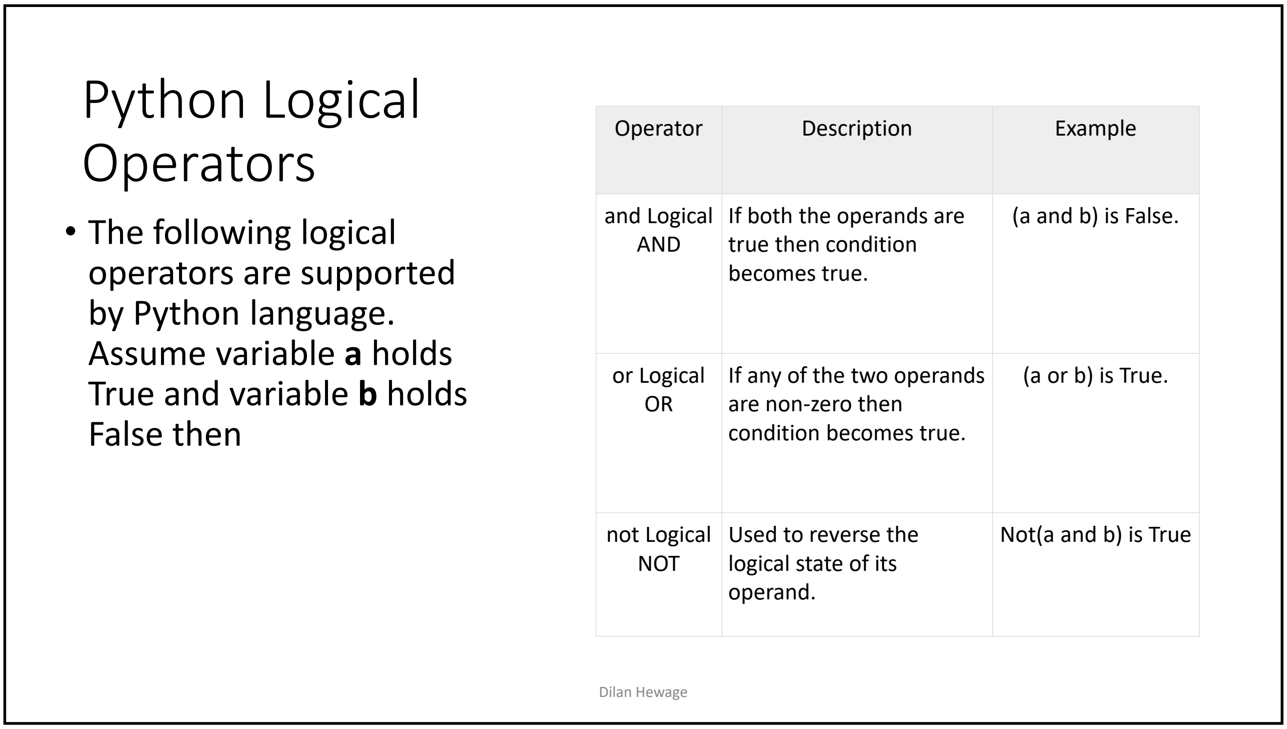
Assignment Operators

- Assume variable **a** holds the value 10 and variable **b** holds the value 20

Operator	Description	Example
=	Assigns values from right side operands to left side operand	c = a + b assigns value of a + b into c
+= Add AND	It adds right operand to the left operand and assign the result to left operand	c += a is equivalent to c = c + a
-= Subtract AND	It subtracts right operand from the left operand and assign the result to left operand	c -= a is equivalent to c = c - a
*= Multiply AND	It multiplies right operand with the left operand and assign the result to left operand	c *= a is equivalent to c = c * a
/= Divide AND	It divides left operand with the right operand and assign the result to left operand	c /= a is equivalent to c = c / ac /= a is equivalent to c = c / a
%= Modulus AND	It takes modulus using two operands and assign the result to left operand	c %= a is equivalent to c = c % a
**= Exponent AND	Performs exponential (power) calculation on operators and assign value to the left operand	c **= a is equivalent to c = c ** a
//= Floor Division	It performs floor division on operators and assign value to the left operand	c //= a is equivalent to c = c // a

Dilan Hewage

9



Python Logical Operators

- The following logical operators are supported by Python language. Assume variable **a** holds True and variable **b** holds False then

Operator	Description	Example
and Logical AND	If both the operands are true then condition becomes true.	(a and b) is False.
or Logical OR	If any of the two operands are non-zero then condition becomes true.	(a or b) is True.
not Logical NOT	Used to reverse the logical state of its operand.	Not(a and b) is True

Dilan Hewage

10

Bitwise Operators

- Bitwise operator works on bits and performs bit-by-bit operation. Assume if a = 60; and b = 13; Now in binary format they will be as follows –
- a = 0011 1100
- b = 0000 1101
- -----------------
- a&b = 0000 1100
- a|b = 0011 1101
- a^b = 0011 0001
- ~a = 1100 0011
- Python's built-in function bin() can be used to obtain binary representation of an integer number.
- The following Bitwise operators are supported by Python languag

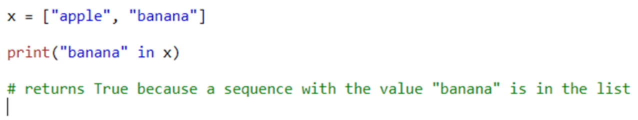
Operator	Description	Example
& Binary AND	Operator copies a bit, to the result, if it exists in both operands	(a & b) (means 0000 1100)
| Binary OR	It copies a bit, if it exists in either operand.	(a | b) = 61 (means 0011 1101)
^ Binary XOR	It copies the bit, if it is set in one operand but not both.	(a ^ b) = 49 (means 0011 0001)
~ Binary Ones Complement	It is unary and has the effect of 'flipping' bits.	(~a) = -61 (means 1100 0011 in 2's complement form due to a signed binary number.
<< Binary Left Shift	The left operand's value is moved left by the number of bits specified by the right operand.	a << 2 = 240 (means 1111 0000)
>> Binary Right Shift	The left operand's value is moved right by the number of bits specified by the right operand.	a >> 2 = 15 (means 0000 1111)

Dilan Hewage

11

Membership Operators

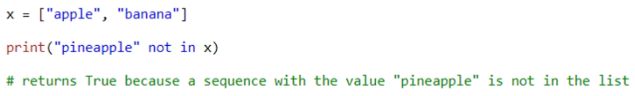
- Membership operators are used to test if a sequence is presented in an object:
- in Returns True if a sequence with the specified value is present in the object
- not in Returns True if a sequence with the specified value is not present in the object



```
x = ["apple", "banana"]

print("banana" in x)

# returns True because a sequence with the value "banana" is in the list
```



```
x = ["apple", "banana"]

print("pineapple" not in x)

# returns True because a sequence with the value "pineapple" is not in the list
```

Dilan Hewage

12

Operators Precedence

Sr.No.	Operator & Description
1	****** Exponentiation (raise to the power)
2	**~ + -** Complement, unary plus and minus (method names for the last two are +@ and -@)
3	*** / % //** Multiply, divide, modulo and floor division
4	**+ -** Addition and subtraction
5	**>> <<** Right and left bitwise shift
6	**&** Bitwise 'AND'
7	**^ |** Bitwise exclusive `OR' and regular `OR'
8	**<= < > >=** Comparison operators
9	**<> == !=** Equality operators
10	**= %= /= //= -= += *= **=** Assignment operators
11	**is is not** Identity operators
12	**in not in** Membership operators
13	**not or and** Logical operators

Dilan Hewage

13

Getting a User Input

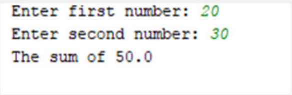
Code

- # Store input numbers
- num1 = input('Enter first number: ')
- num2 = input('Enter second number: ')

- # Add two numbers
- sum = float(num1) + float(num2)

- # Display the sum
- print('The sum of {0}'.format(sum))

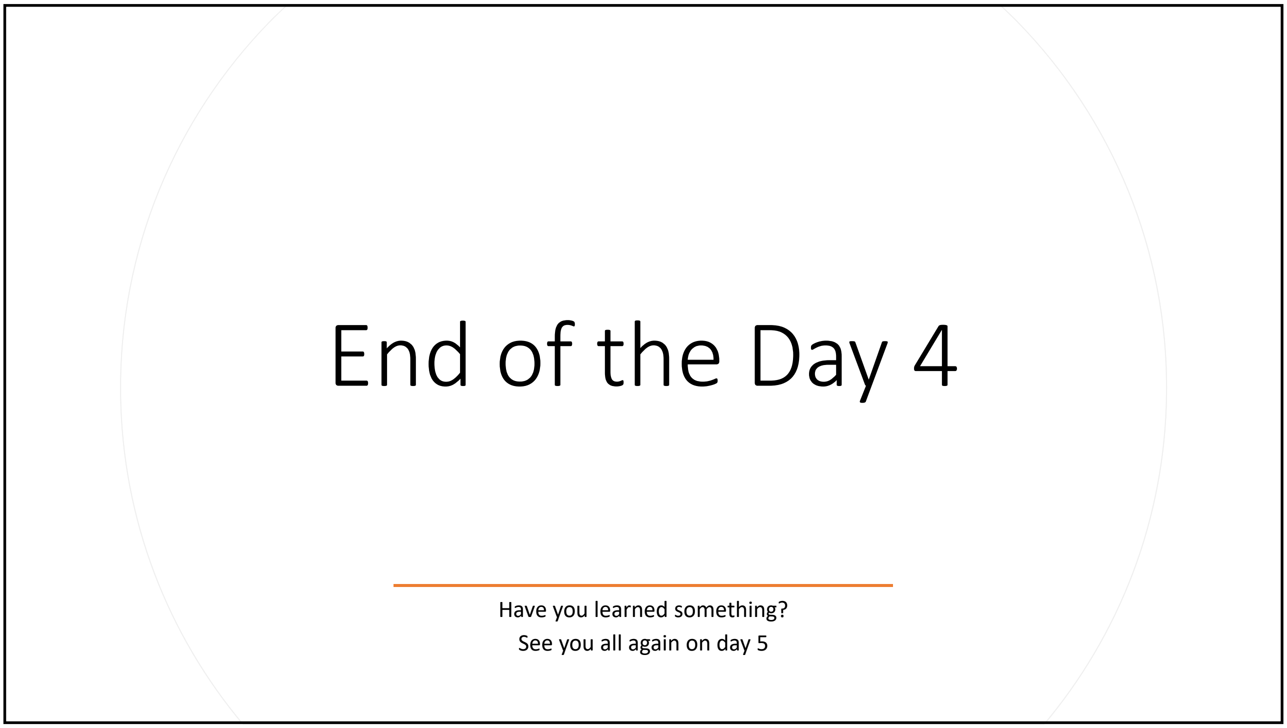
Output



```
Enter first number: 20
Enter second number: 30
The sum of 50.0
```

Dilan Hewage

14



End of the Day 4

Have you learned something?
See you all again on day 5

15