Operators

Now that we have used numeric and string types in Python, we look at operations that may be performed on them.

What Is an Operator?

An **operator** is a symbol that represents an operation that may be performed on one or more *operands*. For example, the + symbol represents the operation of addition. An **operand** is a value that a given operator is applied to, such as operands 2 and 3 in the expression 2 + 3. A **unary operator** operates on only one operand, such as the negation operator in -12. A **binary operator** operates on two operands, as with the addition operator.

Arithmetic Operators

Python provides the arithmetic operators listed below:

The + (addition), - (subtraction), `*` (multiplication) and / (division) arithmetic operators perform the usual operations. Note that the - symbol is used both as a unary operator (for negation) and a binary operator (for subtraction)

Arithmetic Operators		Example	Result
`-x`	negation	`-10`	`-10`
`x + y`	addition	`10 + 25`	`35`
`x - y`	subtraction	`10 - 25`	`-15`
`x * y`	multiplication	`10 * 5`	`50`
`x / y`	division	`25 / 10`	`2.5`
`x // y`	truncating div	`25 // 10`	`2`
		`25 // 10.0`	`2.0`
`x % y`	modulus	`25 % 10`	`5`
`x ** y`	exponentiation	`10 ** 2`	`100`

Python also includes an exponentiation `(**)` operator. Integer and floating-point values can be used in both the base and the exponent,

`2**4 → 16`

`2.5 ** 4.5 → 61.76323555016366`

Python provides two forms of division. **“true” division** is denoted by a single slash, /. Thus, `25 / 10` evaluates to `2.5`. **Truncating division** is denoted by a double slash, `//`, providing a truncated result based on the type of operands applied to. When both operands are integer values, the result is a truncated integer referred to as **integer division**. When at least one of the operands is a float type, the result is a truncated floating point. Thus, `25 // 10` evaluates to `2`, while `25.0 // 10` becomes `2.0`. This is summarized below:

	Operands	result type	example	result
/ Division operator	int, int	float	`7 / 5`	`1.4`
	int, float	float	`7 / 5.0`	`1.4`
	float, float	float	`7.0 / 5.0`	`1.4`
// Truncating division operator	int, int	truncated int ("integer division")	`7 // 5`	`1`
	int, float	truncated float	`7 // 5.0`	`1.0`
	float, float	truncated float	`7.0 // 5.0`	`1.0`

An example of the use of integer division would be to determine the number of dozen doughnuts for a given number of doughnuts. If variable `numDoughnuts` had a current value of `29`, the number of dozen doughnuts would be calculated by,

```
numDoughnuts // 12 → 29 // 12 → 2
```

Lastly, the **modulus operator** `(%)` gives the remainder of the division of its operands, resulting in a cycle of values. This is shown below:

Modulo 7		Modulo 10		Modulo 100	
`0 % 7`	**0**	`0 % 10`	**0**	`0 % 100`	**0**
`1 % 7`	**1**	`1 % 10`	**1**	`1 % 100`	**1**
`2 % 7`	**2**	`2 % 10`	**2**	`2 % 100`	**2**
`3 % 7`	**3**	`3 % 10`	**3**	`3 % 100`	**3**
`4 % 7`	**4**	`4 % 10`	**4**	.	.
`5 % 7`	**5**	`5 % 10`	**5**	.	.
`6 % 7`	**6**	`6 % 10`	**6**	`96 % 100`	**96**
`7 % 7`	0	`7 % 10`	**7**	`97 % 100`	**97**
`8 % 7`	1	`8 % 10`	**8**	`98 % 100`	**98**
`9 % 7`	2	`9 % 10`	**9**	`99 % 100`	**99**
`10 % 7`	3	`10 % 10`	0	`100 % 100`	0
`11 % 7`	4	`11 % 10`	1	`101 % 100`	1
`12 % 7`	5	`12 % 10`	2	`102 % 100`	2

The modulus and truncating (integer) division operators are complements of each other. For example, `29 // 12` gives the number of dozen doughnuts, while `29 % 12` gives the number of leftover doughnuts `(5)`.

Your Turn

From the Python Shell, enter the following and observe the results.

```
>>>10 + 35          >>> 4 ** 2        >>> 45 // 10.0
???                   ???             ???

>>>-10 + 35         >>> 45 / 10       >>> 25 % 10
???                 ???               ???

>>> 4 * 2     >>> 45 // 10      >>> 2025 // 10
???                 ???               ???
```

Part II - Your Place in the Universe

The following program calculates the approximate number of atoms that the average person contains, and the percentage of the universe that they comprise. This program utilizes the following programming features:

➤ floating-point scientific notation

➤ built-in format function

Program Execution ...

```
This program will determine your place in the universe.
Enter your weight in pounds: 150
You contain approximately 3.30e+28 atoms
Therefore, you comprise 3.30e-51 % of the universe
```

```
1  # Your Place in the Universe Program
2  
3  # This program will determine the approximate number of atoms that a
4  # person consists of and the percent of the universe that they comprise.
5  
6  # initialization
7  num_atoms_universe = 10e80
8  weight_avg_person = 70  # 70 kg (154 lbs)
9  num_atoms_avg_person = 7e27
10 
11 # program greeting
12 print('This program will determine your place in the universe.')
13 
14 # prompt for user's weight
15 weight_lbs = int(input('Enter your weight in pounds: '))
16 
17 # convert weight to kilograms
18 weight_kg = 2.2 * weight_lbs
19 
20 # determine number atoms in person
21 num_atoms = (weight_kg / 70) * num_atoms_avg_person
22 percent_of_universe = (num_atoms / num_atoms_universe) * 100
23 
24 # display results
25 print('You contain approximately', format(num_atoms, '.2e'), 'atoms')
26 print('Therefore, you comprise', format(percent_of_universe, '.2e'),
27       '% of the universe')
```

Notes:

Lines 1–4 describe the program. Needed variables `num_atoms_universe`, `weight_avg_person`, and `num_atoms_avg_person` are initialized in lines 7–9. The program greeting is on line 12. Line 15 inputs the person's weight. Line 18 converts the weight to kilograms for use in the calculations on lines 21–22 which compute the desired results. Finally, lines 25–27 display the results.

Concepts and Procedures

1. Give the results for each of the following.

a) `-2 * 3`

b) `15 % 4`

c) `3 ** 2`

2. Give the exact results of each of the following division operations.

a) `5 / 4`

b) `5 // 4`

c) `5.0 // 4`

3. Which of the expressions in question 2 is an example of integer division?

4. Do any two of the expressions in question 2 evaluate to the exact same result? (YES/NO)

5. How many operands are there in the following arithmetic expression?

```
2 * 24 + 60 - 10
```

6. How many binary operators are there in the following arithmetic expression?

```
-10 + 25 / (16 + 12)
```

Problem Solving

1. Which of the following operator symbols can be used as both a unary operator and a binary operator?

   +, -, *, /

2. What is the exact result of each of the following when evaluated?

a) `12 / 6.0`

b) `21 // 10`

c) `25 // 10.0`

3. If variable *n* contains an initial value of 1, what is the largest value that will be assigned to *n* after the following assignment statement is executed an arbitrary number of times?

```
n = (n + 1) % 100
```

4. Which of the following arithmetic expressions could potentially result in arithmetic overflow, where *n* and *k* are each assigned integer values?

a) n * k

b) n ** k

c) n / k

d) n + k

5. Modify the Your Place (from Part III) for international users, so that the user enters their weight in kilograms, and not in pounds.