Arithmetic

August 19, 2016

1 Arithmetic

Python can do basic arithmetic.(Make sure they are in code mode for the next few boxes).

```
In [1]: 6 + 4
```

```
Out[1]: 10
```

Notice we have two lines now: the input line and the output line. Also note the above command was the first one I entered, and has a [1] in both the input and output lines. If I entered it again, it would change to two.

```
In [2]: 6 + 4;
```

We can suppress (get rid of) output with a semicolon. This is useful sometimes when we don't care about the result of a line of code.

Python can do everything your calculator can do. Basic arithmetic examples are here.

```
In [3]: 5 + 9
```

```
Out[3]: 14
```

```
In [4]: 14 - 9
```

```
Out[4]: 5
```

```
In [5]: 24 / 3
```

```
Out[5]: 8
```

```
In [6]: 1999 * 109294
```

```
Out[6]: 218478706
```

```
In [7]: 10**2
```

```
Out[7]: 100
```

The exponential operator isn't the carrot key ^ but rather **.

1.0.1 PEMDAS

Who remembers PEMDAS? Or maybe you learned "Please excuse my dear aunt sally." This is the order in which operations will be carried out when they are present in the same expression. Some examples follow.

```
In [8]: 10 + 3 * 3 - 6 / 2
```

```
Out[8]: 16
```

```
In [9]: 10 + (3 ** 2) ** 3 / 9 - 6
```

```
Out[9]: 85
```

```
In [10]: 1 ** 2 * 3 - 4 + 5
```

```
Out[10]: 4
```

Now to show you something you won't expect. Try division.

```
In [11]: 5 / 2
```

```
Out[11]: 2
```

Why did this happen? Ultimately it's a consequence of how computers store different numbers. But in simple terms, because we gave the computer two integers to divide, it tried to give us an integer result. This ends up being the quotient that an elementary school student would give us when asked to divide 5 and 2. In elementary school terms, 5/2 would be 2 with remainder 1. When Python is asked to divide two integers, it throws away the remainder and just gives the quotient. If we wanted the remainder, we have a special operator just for that, called the modulus. The modulus functions gives *only* the remainder that is left after division, and is performed using the percent sign %

```
In [1]: 5 % 2
```

```
Out[1]: 1
```

If we wanted the "true" anwer of 2.5, we would need to use what is called a floating point number such as 5.0 or 2.0. Example here.

```
In [2]: 5.0/2.0
```

```
Out[2]: 2.5
```

More will come later in the "Data Types" section that we cover next, but essentially, if you expect a decimal answer, you should give Python decimal inputs.

```
In [2]: (1755 + 2281 + 4157, 1605 + 1957, 2346) + (, , 1245) + (, 1926, 876)
```

```
Out[2]: (1755, 1605, 2346, 2281, 1957, 1245, 4157, 1926, 876)
```

```
In [ ]:
```