# 1 Expressions

## 1.1 Arithmetic operators

| Operator | Operation | Expression | Description | Result |
|:---:|:---:|:---:|:---:|:---:|
| + | addition | 3 + 4 | 3 plus 4 | 7 |
| - | subtraction | 3 - 4 | 3 minus 4 | -1 |
| - | negation | -3 + 5 | negative 3 plus 5 | 2 |
| * | multiplication | 3 * 5 | 3 times 5 | 15 |
| / | division (of ints) | 10 / 4 | 10 divided by 2, truncating remainder | 2 |
| / | division (of floats) | 10.0 / 4.0 | 10 divided by 2 | 2.5 |
| ** | exponentiation | 2 ** 4 | 2 to the power of 4 | 16 |
| % | modulo (remainder) | 10 % 3 | 10 mod 3 | 1 |

The inputs to an operator are called **operands**. For example, in this expression, 3 * 5, the operator is * and the operands are 3 and 5.

## 1.2 Types & Expressions

A **type** is a set of values and operations that can be performed on those values.

- int: the type representing integers. Operators include the arithmetic operators above.

- float: the type representing floating points numbers, which are *approximations* of real numbers. Operators include the arithmetic operators above.

An **expression** is a combination of operators and values (and/or variables) that python can evaluate, resulting in a single value. This definition will evolve as the course progresses.

## 1.3 Operator precedence

Higher precedence operators are evaluated first. To break ties, python evaluates from left to right.

| Operator | Precedence |
|:---|:---:|
| ** | highest |
| - (negation) | |
| *, /, % | |
| + (addition), - (subtraction) | lowest |

To control what gets evaluated first, use parentheses. Example:

```
>>> 3 + 2 * 10
23
>>> (3 + 2) * 10
50
```

# 2  Variables & Assignments

Data values are stored in memory. A **variable** is a named memory location. To make or update a variable, use an **assignment statement**.

An assignment statement has this form:

> *variable = expression*

How python executes assignment statement:

1. Evaluate the expression on the right-hand side. This produces a value. The value is stored in memory, at a particular address.

2. Store the memory address in the variable on left-hand side.

Example:
```
>>> hourly_wage = 8.00
>>> hours_worked = 42
>>> total_pay = hourly_wage * hours_worked
>>> total_pay
336.0
```

You can use the python visualizer (`http://www.pythontutor.com/visualize.html`) to visualize of what happens when python executes. Be sure to adjust the settings so they look like this:



Variable names should be meaningful, like in the example above. Variable names must start with a letter or _ and contain only letters, digits, and _. Some names, like `print`, are special keywords in python and cannot be used.

# 3  Errors

There are three kinds of errors in programming:

1. **syntax error**: the statement does not match rules of python language. Ex: `3 + 5 * * 2`

2. **runtime error**: the expression causes a crash. Ex: `4 / (8 - 2 ** 3)`

3. **semantic error**: the program runs but produces the wrong answer. Ex:
```
>>> base = 3
>>> height = 3
>>> triangle_area = base * height / 2
>>> triangle_area    # calculation is wrong: it should be 4.5, where is the bug??
4
```