

05-arithmetic

January 26, 2020

1 Arithmetic in Python

1.0.1 Number types:

- int: ... -3, -2, -1, 0, 1, 2, 3, ...
- float: 3.141592

1.0.2 Operators:

Addition

```
In [1]: 34.0 + 8.3
```

```
Out[1]: 42.3
```

Subtraction

```
In [2]: 57.6 - 15.2
```

```
Out[2]: 42.400000000000006
```

Multiplication

```
In [3]: 6.0 * 7.0
```

```
Out[3]: 42.0
```

Floating point division

```
In [4]: 8 / 2
```

```
Out[4]: 4.0
```

Integer division (always rounds down)

```
In [5]: 7 // 2
```

```
Out[5]: 3
```

Modulo (remainder of integer division)

```
In [6]: 7 % 2
```

```
Out[6]: 1
```

Note that:

- Can be used to check for divisibility
- If $q = x // y$, and $r = x \% y$ then: $x = y * q + r$

Exponentiation

```
In [7]: 3 ** 2
```

```
Out[7]: 9
```

Absolute value (disregard the sign)

```
In [8]: abs(-3)
```

```
Out[8]: 3
```

```
In [9]: abs(3)
```

```
Out[9]: 3
```

Ceiling function (rounding up)

```
In [10]: import math          # tells python that we want to use functions from the math library
                                     # import will be in the starter file for the most part of the course
         math.ceil(4.2)
```

```
Out[10]: 5
```

Flooring function (rounding down)

```
In [11]: import math
```

```
         math.floor(4.2)
```

```
Out[11]: 4
```

1.0.3 Exercise 1: temperature conversion

Fahrenheit temperatures can be calculated in terms of Celcius temperatures using the following formula:

$$F = C \times \frac{9}{5} + 32$$

Implement the functions `celciusToFahrenheit(c)` and `fahrenheitToCelcius(f)` that converts the temperatures.

```
In [12]: def celciusToFahrenheit(c):
         return 42.0
```

```
         def fahrenheitToCelcius(f):
         return 42.0
```

1.0.4 Exercise 2: mixing colors

Colors in a computer are sometimes represented using integer RGB values, corresponding to the amount of red, green, and blue it is composed of. These values are integers between 0 and 255. For example, (0,0,0) is black, and (255,255,255) is white.

Given two colors, we can combine them to form a palette (the colors in between the two). We can decide how many colors we want in this palette (the midpoints).

For example, we can combine the colors crimson (RGB = (220, 20, 60)) and mint (RGB = (189, 252, 201)) using 3 midpoints to obtain:

```
color0: (220, 20, 60) # crimson
color1: (212, 78, 95)
color2: (205, 136, 131)
color3: (197, 194, 166)
color4: (189, 252, 201) # mint
```

This palette has 5 colors. We could then ask ourselves: what is color 2?

Implement the function `getColor(r1, g1, b1, r2, g2, b2, midpoints, n)` that takes as input the two initial colors, the number of midpoints, and the number of the color in the palette, and returns the RGB value of the n-th color.

For example, following the case above: `getColor(220, 20, 60, 189, 252, 201, 3, 1)` returns (212, 78, 95).

Hint: for the sake of this exercise, it is ok if your result differs by one unit from the example, since integer division is not exact. We will learn how to fix this later.

Hint: try thinking about the problem with one component at a time (red, green, or blue), and with small midpoints (e.g. 1, 2, 3).

```
In [13]: def getColor(r1, g1, b1, r2, g2, b2, midpoints, n):
         return (1, 2, 3)
```

1.0.5 Exercise 3: the straggler

In motorsports it is very common that the leader pilot in a race, at a certain moment, overtakes the last pilot. The leader, at this moment, is one lap ahead of the last pilot, who then becomes a straggler (or tailender).

Implement the function `straggler(fast, slow)` that takes as input the time it takes for the faster and slowest pilots to complete a lap, and returns the number of the lap when the fastest pilot overtakes the slowest one. The first lap is number 0.

For example, `straggler(5, 7)` should return 3.

```
In [14]: def straggler(fastest, slowest):
         return 42
```