**Introduction to Python**

**Day One Cheatsheet: Operations and Data Types**

**Mathematical Operators**

|  |  |  |
| --- | --- | --- |
| Symbol | What it does | Example |
| + | Addition | **5 + 2** |
| - | Subtraction | **5 - 2** |
| \* | Multiplication | **5 \* 2** |
| / | Division | **5 / 2** |
| % | Modulus (remainder) | **5 % 2** (results in 3) |
| \*\* | Exponent | **5 \*\* 2** (results in 25) |

**Variable types**

|  |  |  |
| --- | --- | --- |
| Type | Abbreviation | Example(s) |
| Integer | int | **5 ; -6; 0** |
| Float | float | **45.55 ; -9.21 ; 1.0** |
| List | list | [1,2,3] ; ["hi", "bye", 6] |
| String | str | "word" ; "several words" ; "55" |
| Dictionary | dict | {"a": 1, "b": 2, "c": 6} |

**Common list methods**

Each example below uses this list: my\_list = [2, 3, 4, 5]

|  |  |  |
| --- | --- | --- |
| Method | Description | Example(s) |
| .append() | Add item to end of list | my\_list.append(7)  [2, 3, 4, 5, 7] |
| .index() | Return first index of a value | my\_list.index(3)  1 |
| .pop() | Remove a given index from a list | my\_list.pop(2)  [2, 3, 5] |

**Common string methods**

|  |  |  |
| --- | --- | --- |
| Method | Description | Example(s) |
| .upper() | Convert a string to all uppercase | my\_str.upper()  I LOVE PYTHON |
| .lower() | Convert a string to all lowercase | my\_str.lower()  i love python |
| .count() | Count occurrences of a character (case-sensitive!) | my\_str.count("o")  2 |
| .split() | Convert a string to a list, splitting on the given character | my\_str.split(" ")  ["I", "love", "Python"] |

Each example below uses this string: my\_str = "I love Python"

**Common dictionary methods**

Each example below uses this dictionary:

my\_dict = {"alpha": "a", "beta": "b", "delta": "d"}

**Indexing**

General paradigm [x:y:z]

* x: inclusive first index (default: 0)
* y: exclusive final index (default: last index)
* z: step/increment (default: 1)

|  |  |  |
| --- | --- | --- |
| Method | Description | Example(s) |
| .keys() | Return the dictionary keys as a list | my\_str.keys()  ["alpha", "delta", "beta"] |
| .values() | Return the dictionary values as a list | my\_str.values()  ["a", "d", "b"] |

a = [90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100]

Indices: 0 1 2 3 4 5 6 7 8 9 10

a[0] = 90

a[5] = 95

a[:3] = [90, 91, 92]

a[6:] = [96, 97, 98, 99]

a[3:6] = [92, 93, 94, 95]

a[1:8:2] = [91, 93, 95, 97]

a[-1] = 100