CHEATSHEET: PYTHON I

MATHEMATICAL OPERATORS

Symbol	Purpose
+	Addition
–	Subtraction
*	Multiplication
/	Division
**	Exponent (e.g., 2**3 = 8)
%	Modulus, i.e. remainder (e.g., 5%2 = 2)

LOGICAL OPERATORS

Symbol	Purpose
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to
==	Equal to
!=	Not equal to

Remember, logical statements can be combined with **`and`**, **`or`**

VARIABLE TYPES

Variable	Description	Defining	Examples
Integer	Whole number	int()	5, 10, -8
Float	Decimal number	float()	5.4, 10.2, -8.11, 9.0
String	Immutable container of characters	"" or ' ' str()	"This is a string." '12345'
List	Mutable container	[ ] list()	[1, 2, 4.5, 7, 10] [1, 2, “string”, -55.34] [1, 2, [3, 4, 5]]

Dictionary	Unordered container (associative array)	{ }
		dict()
			{“alpha”: “a”, “beta”: “b”}
			{“height”: 100, “length”: 20}
			{75: "odd", 4: "even", 12: "even"}

Tuple	Immutable container (unchangeable list)	( ) tuple()	(1,2,3) (1, 3.4, “goodbye”, [1,2,3])

CHEATSHEET: PYTHON I

INDEXING IN PYTHON

```
General paradigm [x:y:z]
   • x: inclusive first index  (default: 0)
   • y: exclusive final index  (default: last index)
   • z: step/increment         (default: 1)

Example:
     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
```

USEFUL FUNCTIONS

Function	Purpose	Examples

`len()`	Returns the length of a container	`a = [1, 2, 3]`
		`len(a)  # Returns 3`
		
		`b = "Words!"`
		`len(b)  # Returns 5`

`range()`	Returns a list according to indexing rules	`range(1,5)   # Returns [1,2,3,4]`
		`range(4)     # Returns [0,1,2,3]`
		`range(1,8,2) # Returns [1,3,5,7]`

`print()`	Prints	`print("I am printing this to screen.")`
		`print(5)`

`help()`	Obtain documentation for a function	`help(len)`
		`help(range)`

`dir()`	Determine available actions for an object	`a = [1,2,3]`
		`dir(a)`

`type()`	Determine the type of a variable	`a = [1,2,3]`
		`type(a)   # Returns <list>`
		`b = "hi"`
		`type(b)   # Returns <str>`
		`c = 52`
		`type(c)   # Returns <int>`

CHEATSHEET: PYTHON I

USEFUL STRING METHODS

```
Remember, these methods WILL NOT change the value of the variable!
Examples shown below are performed on one of these example strings:
    x = "AbCdEfG"
    y = "a b c d"
    z = "   hello"
```

Method	Description	Example

`.upper()`	Returns the uppercase version of the string	`x.upper()`
		`# 'ABCDEFG'`

`.lower()`	Returns the lowercase version of the string	`x.lower()`
		`# 'abcdefg'`

`.count()`	Count the occurrences of a given character (note: this is case-sensitive!)	`x.count("A")`
		`# 1`
		`x.count("a")`
		`# 0`

`.replace()`	Replaces occurrences of a given character with a different character	`x.replace("b", "5")`
		`# 'A5CdEfG'`

`.split()`	Convert a string to a list by "splitting" on a certain character	`y.split()`
		`# ['a','b','c','d']`

`.strip()`	Remove all leading and trailing whitespace. Note: .rstrip() removes **trailing** only, and .lstrip() removes **leading** only	`z.strip()`
		`# 'hello'`

USEFUL LIST METHODS

```
Examples shown below are performed on one of these example lists:
    x = [1, 2, 3, 4]
    y = [1, 2, 3, 4, 6, 6, 6]
```

Method	Description	Example

`.append()`	Add a new element to the end of the list. Remember, this redefines the list in place!	`x.append(5)`
		`# [1, 2, 3, 4, 5]`

`.index()`	Determine the list index of a certain value	`x.index(2)`
		`# 3`

`.count()`	Count the occurrences of a given value	`y.count(6)`
		`# 3`