

# Duke CS101 Python Reference Sheet

## Mathematical Operators

Symbol	Meaning	Example
+	addition	$4 + 5 = 9$
-	subtraction	$9 - 5 = 4$
*	multiplication	$3 * 5 = 15$ and $4.0 * 5 = 20.0$
/	division	$6 / 4 = 1.5$
//	floor division	$6 // 4 = 1$
%	mod/remainder	$5 \% 3 = 2$
**	exponentiation	$2 ** 3 = 8$ , $3 ** 2 = 9$

## String Operators

Symbol	Meaning	Example
+	concatenation	"ab" + "cd" == "abcd"
*	repeat	"x" * 3 == "xxx"

## Comparison Operators

Symbol	Meaning	Example
==	is equal to	$3 == 3$ is True
!=	is not equal to	$3 != 3$ is False
>=	is greater than or equal to	$4 >= 3$ is True
<=	is less than or equal to	$4 <= 3$ is False
>	is strictly greater than	$4 > 3$ is True
<	is strictly less than	$3 < 3$ is False

## Boolean operators

Assume  $x = 5$

not	flips/negates the value of a bool	(not $x == 5$ ) is False
and	returns True only if both parts of the expression are true	( $x > 3$ and $x < 7$ ) is True
or	returns True if at least one part of the expression is True	( $x < 3$ or $x > 7$ ) is False ( $x < 3$ or $x < 7$ ) is True

## Type Conversion Functions

int(x)	turn x into an integer value. int can fail, e.g., int("abc") raises an error	int("123") == 123
float(x)	turn x into a float value, float can fail, e.g., float("abc") raises an error	float("2.46") == 2.46
str(x)	turn x into a string value	str(432) == "432"
type(x)	the type of x	type(1) == int type(1.2) == float

## List Functions

lst.append(...)	append an element to lst, changing lst	[1,2,3].append(8) == [1,2,3,8]
lst.index(elt)	returns the first index of elt if it is in lst, otherwise causes an error	[1,2,3].index(2) == 1 [1,2,3].index(4) # causes an error
lst.count(elt)	return number of occurrences of elt in lst	[1,2,1,2,3].count(1) == 2

## Miscellaneous Functions

len(x)	length of sequence x, e.g., String/List	len("duke") == 4
range(x)	a sequence of integers starting at 0 and going up to but not including x	range(5) == [0, 1, 2, 3, 4]
range(start, stop)	a sequence of integers starting at start and going up to but not including stop	range(3, 7) == [3, 4, 5, 6]
range(start, stop, inc)	a sequence of integers starting at start and going up to but not including stop with increment inc	range(3, 9, 2) == [3, 5, 7]

## String Functions

Assume  $s = \text{"colorful"}$

.find(str)	index of first occurrence	s.find("o") == 1 s.find("e") == -1
.rfind(str)	index of last occurrence	s.rfind("o") == 3 s.rfind("e") == -1
.count(str)	number of occurrences	s.count("o") == 2 s.count("r") == 1 s.count("e") == 0
.strip()	copy with leading/trailing whitespace removed	" big ".strip() == "big"
.split()	list of "words" in s separated by whitespace	"big bad dog".split() == ["big", "bad", "dog"]
.split(",")	list of "items" in s that are separated by a comma. In general can split on any string, not just a comma, e.g., s.split(":") will split on a colon and s.split("gat") will split on the string "gat".	"this,old,man".split(",") == ["this", "old", "man"]
' '.join(lst)	concatenate elements of lst, a list of strings, separated by ' ' or any string	' '.join(['a', 'b', 'c']) == "a:b:c"
.startswith(str)	boolean if starts with string	s.startswith("color") == True s.startswith("cool") == False
.endswith(str)	boolean if ends with string	s.endswith("ful") == True s.endswith("color") == False
.upper()	uppercase of s	s.upper() == "COLORFUL"
.lower()	lowercase of s	"HELLO".lower() == "hello"

## Math Functions (import math)

math.pi	3.1415926535897931	
math.sqrt(num)	returns square root of num as float	math.sqrt(9) == 3.0