Python Operators Cheat Sheet

Assignment

=	Assignment a=2 value of a becomes 2	
+=	Addition and assignment	i+=1 is the same as i=i+1
-=	Subtraction and assignment	i-=1 is the same as i=i-1
*= /= etc	all other operators can be using in conjunction with the assignment operator	

Arithmetic operators

Operator	Description	
+	Addition	
-	Subtraction	
*	Multiplication	
/	Division	
%	Modulus (or remainder)	3%2=1 10%7=3
**	Exponent (or power)	2**4=16
//	Floor division (or integer division)	3//2=1 10//3=3
abs()	absolute value (distance from zero)	abs(-2) = 2 abs(2) = 2

String operators

+	Concatenation	“word “+” more” becomes `wordmore`
*	Multiplication	“word”*2 becomes `wordword`
<string> []	index	
<string> [:]	slice (or range)	
len(<string>)	length of string	len(“word”) is 4
for I in <string>	iteration through characters	

Comparison Operators

==	returns true if values are equal	1=1 true “test”==”test” true
!=	Returns true if values are not equal	2!=1 true “test”!=”test” false
<>	Returns true if values are not equal	2<>1 true “test”<>”test” false
>	Greater than	2>1 is true
<	Less than	2<1 is false
>=	Greater than or equal to	
<=	Less than or equal to	

Logic Operators

and	logical AND returns true if both operands are true	1=1 and “test”=”test” is true
or	logical OR returns true if either operand is true	1=1 or “test”=”bit” is true
not	logical NOT reverses operand	not(1=1) is false

Membership Operators

in	tests if left operand is in a collection	1 in [4,5,1,3,5] is true
not in	tests if left operand is not in a collection	1 in [4,5,1,3,5] is false

Binary Operators

&	Binary AND copies bit where it exists in both operands	0011&1001 becomes 0001
|	binary OR copies bits where it exists in either operand	0011|1001 becomes 1011
^	binary XOR copes the bit where it exists in only one operand	0011^1001 becomes 1010
~	binary one's complement (unary*) reverses bit value	0011 becomes 1100
<<	binary left shift Left operand is shifted left by number of bits in right operand	0011<<2 becomes 1100
>>	binary right shift Left operand is shifted right by number of bits in right operand	1001>>2 becomes 0010

Operator Precedence

**	Exponentiation
+x, -x, ~x	Positive, negative, bitwise NOT
*, /, //, %	Multiplication, division, remainder
+, -	Addition and subtraction
<<, >>	Shifts
&	Bitwise AND
^	Bitwise XOR
|	Bitwise OR
in, not in, is, is not, <, <=, >, >=, !=, ==	Comparisons, including membership tests and identity tests
not	Boolean NOT
and	Boolean AND
or	Boolean OR