Lecture 1

Basic data types: `int`, `long`, `float`, `bool`, `str`, `list`. There are others we aren't covering yet.

Operators for numerical data types (`int`,`long`,`float`):

Operator	Description	Example
$+$	add two values	$5+2$ gives 7
$-$	subtract two values	$5-2$ gives 3
$*$	multiply two values	$5*2$ gives 10
/	divide two values rounds down for ints	5.0/2.0 gives 2.5 5/2 gives 2
%	"mod", remainder after division	5%2 gives 1
$**$	exponentiation	$5**2$ gives 25
//	division with floor (rounds down to integer value)	5.0//2.0 gives 2.0 5//2 gives 2

Operators for bools:

Operator	Description	Example

`and`	logical and	`True and True` gives `True`
		`True and False` gives `False`
		`False and True` gives `False`
		`False and False` gives `False`

`or`	logical or	`True or True` gives `True`
		`True or False` gives `True`
		`False or True` gives `True`
		`False or False` gives `False`

`not`	logical not	`not True` gives `False` `not False` gives `True`

Operators and indexing for lists:

Operator	Description	Example
$+$	concatenate two lists	$[5]+[2]$ gives $[5,2]$
$*$	repeat a list several times (`list`*`int` or `int`*`list`)	$[1,2]*3$ gives $[1,2,1,2,1,2]$
$[i]$	gives ith element in the list	['a','b','c'][0] gives 'a'
$[i:j]$	take sublist from element i to $j-1$	['a','b','c','d','e'][1:3] gives ['b','c']

Operators and indexing for strings:

Operator	Description	Example
$+$	concatenate two strings	'hi' + ' there' gives 'hi there'
$*$	repeat a string several times (`str`*`int` or `int`*`str`)	'hi'*3 gives 'hihihi'
$[i]$	gives ith character in the string	'abcdefg'[3] gives 'd'
$[i:j]$	take substring from character i to $j-1$	'abcdefg'[3:5] gives 'de'

Other stuff:

Comparison Operators
(in below table, the description assumes a OPERATOR b)

Operator	Description	Example
$==$	True if a equals b	$5 == 2$ is False
$!=$	True if a doesn't equal b	$5! = 2$ is True
$<>$	Same as $!=$	$5 <> 2$ is True
$>$	True if a is greater than b	$5 > 2$ is True
$<$	True if a is less than b	$5 < 2$ is False
$>=$	True if a is greater than or equal to b	$5 >= 2$ is True
$<=$	True if a is less than or equal to b	$5 <= 2$ is False

Assignment Operators
Left hand side is a variable, call it "x", and right hand side is an expression, call it "b".
(In examples, assume x has the value 17)

Operator	Description	Example
$=$	Sets x to the result of b	$x = 5 + 2$ sets x to 7
$+=$	Sets x to $x + b$	$x+ = 5 + 2$ sets x to 14
$-=$	Sets x to $x - b$	$x- = 5 + 2$ sets x to 7
$*=$	Sets x to $x * b$	$x* = 5 + 2$ sets x to 49
$/=$	Sets x to x/b	$x/ = 5 + 3$ sets x to 6
$\%=$	Sets x to $x\%b$	$x\% = 5 + 2$ sets x to 6
$**=$	Sets x to $x**b$	$x** = 2$ sets x to 36
$//=$	Sets x to $x//b$	$x// = 5 + 2$ sets x to 5

Operator Precedence, from highest precedence to lowest
(remember you can override operator precedence using parentheses)

Operation
$**$
$*$ / % //
$+-$
$<= < > >=$
$<> == !=$
$= \%= /= //= -= += *= **=$
`not or and`

Example of defining and evaluating a function:

```
>>> def addTwo(n):
...     return n+2
...
>>> addTwo(5)
7
```

Note: variables can point to functions.

```
>>> x = addTwo
>>> x(2)
4
```