

<code>__eq__</code> compares addresses of two items by default. Overrides <code>==</code> .	Frozenset	two param function and a list, and goes iterating through with a "cur" and a "next" as the first and second parameter being passed into the function. For reduce, the default is either the first value or the third parameter
<code>__getitem__</code> gets the item at a specific key	<code>floordiv()</code> is like Java's integer division	
<code>__init__</code> is Python's constructor, first param is always passed in and is usually called "self"	<code>def</code> declares a function	
<code>__iter__</code> returns the iterable for a Python object	<code>issubclass</code> with parameter <code>classname</code> checks if a class is a subclass of another class	Positional arguments have to go before named arguments
<code>__next__</code> pops the next item off of an iterable	<code>truediv()</code> is like Java's double division	For parameters, you basically replace <code>*</code> with all the values as positional arguments, and replace <code>**</code> (dict values) with named arguments
<code>assert</code> is good for pre and postconditions, bad for user input or testing	<code>pass</code> is used when you have no code to put inside an <code>if/else/try/except</code>	Passing in the same parameter twice into a function causes an error
<code>unittest</code> is good for testing	Use a comma at the end of a single item tuple	Using <code>*</code> then <code>**</code> in the function declaration can let the user pass anything into a function
use <code>try</code> and <code>except</code> with <code>StopIteration</code> exception to build a custom iterator	<code>Yield</code> is used as a return inside generators	
Mutable: List, Set	<code>Map</code> applies a function to all items inside of an iterable. The first parameter is the function to be applied, the second is the list that is iterable. Reduce takes a	
Immutable: Tuple,		