Introduction to built-ins

Built-ins are a somewhat overlooked part of Python. You use them every day, but there are a number of them that get overlooked or just aren't used to their full potential. This chapter won't be covering all the built-ins in Python, but will focus on the ones that you probably don't use every day.

any

The **any** built-in accepts an iterable and will return True is any element in said iterable is True. Let's take a look at an example:

```
print (any([0,0,0,1]))
#True
```

In this case, we pass **any** a list of zeros and a one. Because there's a one there, it returns True. You might be wondering when you would ever use this built-in. I did too at first. An example that cropped up in one of my jobs involved a very complex user iterface where I had to test various pieces of functionality. One of the items that I needed to test was if a certain list of widgets had been shown or enabled when they shouldn't be. The any built-in was very useful for that.

Here's an example that kind of demonstrates what I'm talking about, although it's not the actual code I used:

```
widget_one = ''
widget_two = ''
widget_three = 'button'
widgets_exist = any([widget_one, widget_two, widget_three])
print (widgets_exist)
#True
```

Basically I would query the user iterface and ask it if widgets one through three existed and put the responses into a list. If any of them returned True, then I'd raise an error.

You might want to check out Python's **all** built-in as it has similar functionality except that it will only return True if every single item in the iterable is True.