

List Methods

Lists are data types that are **mutable**. Mutable means they can change. Here we will learn how we can add new items to a list. Methods are operations you can ACT on or DO to a data type. In other words, methods allow us to mutate a list!

I have this list of cities I've visited in Oregon and Washington:

```
cities = ["Portland", "Vancouver", "Eugene", "Seattle", "Boring",  
"Beaverton"]
```

But I just went on a trip and added a new city to my list – Hillsboro!!! To add it to my list I can use the **append method**.

```
cities.append("Hillsboro")
```

Where was "Hillsboro" added in my list? (What is the index)?

But what if I need to add multiple cities to my list? What if I visited Hillsboro and Tigard? We can **concatenate** lists!

Before we continue, let's remember what **concatenate** means.

Okay, we're ready to concatenate:

```
all_the_cities = cities + ["Hillsboro", "Tigard"]
```

What is the value of **all_the_cities**?

So **concatenating** lists and using the **append command**, both add list items to the end of the list. What if we want to choose where the item goes in our list? There's a command for that!

```
cities = ["Portland", "Vancouver", "Eugene", "Seattle", "Boring",  
"Beaverton"]
```

In repl try:

```
cities.insert(1, "Hillsboro")
```

What happened? Why? What is the role of the integer in this command?

Let's try some more examples:

```
cities.insert(0, "Tigard") =
```

```
cities.insert(2, "Milwaukie") =
```

```
cities.insert(3, "Cannon Beach") =
```

So now you have a really long list of cities! Try the following methods and write down what happens. After you're done, we'll go over it as a group. BUT, before you try the other methods, don't forget to define another list called, **more_cities**.

```
more_cities = ["Seaside", "Tacoma", "Medford"]
```

```
cities.extend(more_cities)
```

```
cities.index("Tigard")
```

```
cities.remove("Milwaukie")
```

```
cities.pop(2)
```

```
cities.sort()
```

```
cities.reverse()
```

OK! Now you come up with your own examples and try it out!

You can also add to an empty list!

```
letters = [ ]
```

Try these commands and explain what happened in your own words:

```
letters.append('a')
```

```
letters.extend( [ 'b', 'c' ] )
```
