Practice Answers

1.

```
l = [1,2,3,4,5]
l
[1, 2, 3, 4, 5]
```

This list functions.

b.

```
+ = [1, 2,3,4,5]
  File "<iPython-input-13-51e3815d86e1>", line 1
    + = [1, 2,3,4,5]
      ^
SyntaxError: invalid syntax
```

and c.

```
1 = [1, 2,3,4,5]
  File "<iPython-input-13-51e3815d86e1>", line 1
    1 = [1, 2,3,4,5]
      ^
SyntaxError: invalid syntax
```

In the latter two cases, Pythonis giving us an informative error. ASyntaxError tells us that something is wrong with the way in which we have stated our commands. In this case, our syntax is wrong because we have tried to declare our list to be called something stupid. Calling a list + is confusing. If you later try to do some addition, Pythonwill insert the list. Numbers, mathematical operands and bools (True, False) cannot be assigned as list or variable names. Python*will* allow you to name your list "list". Not a great idea - that name is not very descriptive. If youre me, youll forget what is in it. No bueno.

d.

```
l.remove(2)
l.remove(4)
l
 [1, 3, 5]

```

The remove function allows us to remove specific values.

```
l = []
```

The above function will remove all of them. Scary.

```
l.append(2)
l.append(4)
l
[1, 2,3,4,5]
```

We didnt actually cover a way to append in place (i.e. to a specific spot in your list). The way you know how to do this is:

```
l.append(2)
l
[1,3,5,2]
```

Which is cool, but out of order. Some of you tried some clever stuff like this:

```
l[2] = 2
l
[1,3,2,2]
```

Now weve replace our variable, which is no good either. So Im going to show you a trick.

```
l.insert(3, 4)
l
[1, 3, 2, 4, 2]
```

This has inserted the number four at the third postition in the list. Remember that we count from zero in Python!

2a.

```
l = [1,2,3,4,5]
added_list = []


for x in l:
    added_list.append(x + 1)

added_list

[2, 3, 4, 5, 6]
```

2b.

```
rev_list = added_list[::-1]

```

Other clever stuff y'all asked about

Permanence: If I close the Pythoninterpreter, what happens to my stuff?

It vanishes! We'll talk about scripting in the future, which will help you create permanent records of your variables and commands.

Spaces

Did you try to name a variable my variable? Did it fail? It should have! Python, like most other languages, doesn't do great with spaces in variable names. Instead of my variable, try MyVariable or my_variable. Also, Pythonis case sensitive. So MyVariable is different than myvariable. Try goofing with names until you've convinced yourself I'm telling the truth.

Good work team, see you next week!