String Slicing

In this lesson, we learn about string slicing in Python

One subject that you'll find yourself doing a lot of in the real world is string slicing. I have been surprised how often I have needed to know how to do this in my day-to-day job. Let's take a look at how slicing works with the following string:

```
my_string = "I like Python!"
```

Each character in a string can be accessed using slicing. For example, if I want to grab just the first character, I could do this:

```
my_string = "I like Python!"
print(my_string[0:1])
```

▷

This grabs the first character in the string up to, but **not** including, the 2nd character. Yes, Python is zero-based. It's a little easier to understand if we map out each character's position in a table:

0	1	2	3	4	5	6	7	8	9	10	11	12	13
I		l	i	k	e		P	y	t	h	o	n	!

Thus we have a string that is 14 characters long, starting at zero and going through thirteen. Let's do a few more examples to get these concepts into our heads better.

```
my_string = "I like Python!"

print(my_string[:1])   # 'I'
print(my_string[:1])   # 'I'
print(my_string[0:12]) # 'I like Pytho'
print(my_string[0:13]) # 'I like Python'

print(my_string[0:14]) # 'I like Python!'
print(my_string[0:-5]) # 'I like Py'
print(my_string[:])    # 'I like Python!'
print(my_string[2:])   # 'like Python!'
```

As you can see from these examples, we can do a slice by just specifying the beginning of the slice (i.e. my_string[2:]), the ending of the slice (i.e. my_string[:1]) or both (i.e. my_string[0:13]). We can even use negative values that start at the end of the string. So the example where we did my_string[0:-5] starts at zero, but ends 5 characters before the end of the string.

You may be wondering where you would use this. I find myself using it for parsing fixed width records in files or occasionally for parsing complicated file names that follow a very specific naming convention. I have also used it in parsing out values from binary-type files. Any job where you need to do text file processing will be made easier if you understand slicing and how to use it effectively.

You can also access individual characters in a string via indexing. Here is an example:

```
my_string = "I like Python!"
print(my_string[0])
print(my_string[2])
print(my_string[7])
print(my_string[100]) # IndexError: string index out of range
```

Notice how we got IndexError when we tried to access a character beyond the length of the string