Pattern Matching Using search

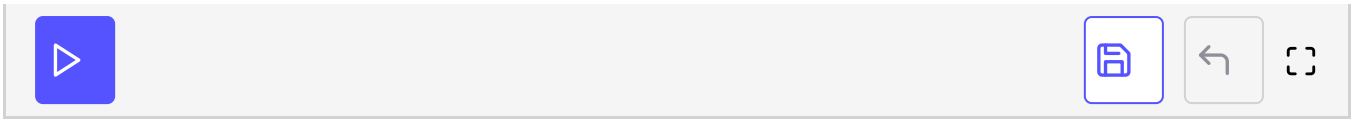
Let's take a moment to learn some pattern matching basics. When using Python to look for a pattern in a string, you can use the **search** function like we did in the example earlier in this chapter. Here's how:

```
import re

text = "The ants go marching one by one"

strings = ['the', 'one']

for string in strings:
    match = re.search(string, text)
    if match:
        print('Found "{}" in "{}"'.format(string, text))
        text_pos = match.span()
        print(text[match.start():match.end()])
    else:
        print('Did not find "{}"'.format(string))
```



For this example, we import the **re** module and create a simple string. Then we create a list of two strings that we'll search for in the main string. Next we loop over the strings we plan to search for and actually run a search for them. If there's a match, we print it out. Otherwise we tell the user that the string was not found.

There are a couple of other functions worth explaining in this example. You will notice that we call **span**. This gives us the beginning and ending positions of the string that matched. If you print out the **text_pos** that we assigned the span to, you'll get a tuple like this: (21, 24). Alternatively, you can just call some match methods, which is what we do next. We use **start** and **end** to grab the starting and ending position of the match, which should also be the two numbers that we get from span.