

Greedy/Non-Greedy Matching

Python Greedy and non-greedy matching explained with examples.

There are times when you want to match a pattern only optionally! The `?` character flags the group that precedes it as an optional part of the pattern. For example, enter the following into the interactive shell:

```
1 import re
2
3 Regex = re.compile(r'(scientific)?programming')
4 m1 = Regex.search('Learn programming')
5 m2 = Regex.search('Learn scientific programming')
6
7 print m1.group()
8 print m2.group()
```



The output will be:

```
programming
scientific programming
```

This means that the `(scientific)?` part of the regular expression means that the pattern `scientific` (notice the white space!) is an **optional** group. The regex will match text that has zero instances or one instance of `scientific` in it. This is why the regex matches both 'programming' and 'scientific programming'.

Note that the `*`, `+`, and `?` qualifiers are all **greedy**; they match as much text as possible. Sometimes this behavior isn't desired; if the RE pattern `<.*>` is matched against '`<H1>Learn Scientific Programming</H1>`', it will match the entire string, and not just '`<H1>`'. Adding `?` after the qualifier makes it perform the match in non-greedy or minimal fashion; as few characters as possible will be matched. Using `.*?` in the previous expression will match only '`<H1>`'.

only `<H1>`.