Python regex `compile` function

Python regex compile() function explained with examples. Includes learning tasks.

WE'LL COVER THE FOLLOWING

Python regex compile

The `compile` function compiles a regular expression pattern into a regular expression `object`, which can be used for matching using its `match()`, `search()`, etc. methods.

Syntax

```
re.compile(pattern, flags=0)
```

The expression's behaviour can be modified by specifying a flags value (discussed earlier). Values can be any of the following variables, combined using bitwise OR (the `|` operator).

For example:

```
m = re.match(pattern, string)
```

is equivalent to:

```
p = re.compile(pattern)
m = p.match(string)
```

Note that the programs that use only **a few regular expressions at a time**

Note that the programs that use only **a few regular expressions at a time** don’t need to compile regular expressions (recent patterns are cached automatically due to `re._MAXCACHE` setting).

Example

Consider that you have an html file index.html like below:

```
<html>
<header>SP:</header>
<body>
<h1>Learn</h1>
<p>Scientific Programming</p>
</body>
</html>
```

You want to read this file and output:

```
SP:Learn Scientific Programming
```

The following code can do this:

main.py

index.html

```
import re
import os
def main():
    f = open('index.html')
    pattern = re.compile(r'(?P<start><.+?>)(?P<content>.*?)(</.+?>)')
    output_text = []
    for text in f:
        match = pattern.match(text)
        if match is not None:
            output_text.append(match.group('content'))

    fixed_content = ' '.join(output_text)

    print fixed_content

    f.close()

if __name__ == '__main__':
    main()
```

Learning Tasks

- How the regex pattern `(?P<start><.+?>)(?P<content>.*?)(</.+?>)` matches the string `<h1>Learn</h1>`.
- Learn the use of the Python `.append` and `.join` functions. Hints: `join()` returns a string in which the string elements of sequence have been joined by `str` separator and `append()` Add an item to the end of the list; equivalent to `a[len(a):] = [x]`.