

```
...
```

```
regex.py
```

```
Jeff Ondich, 16 Jan 2009
```

```
Tianna Avery, November 2018, updated for Python 3
```

```
This program gives a brief illustration of a few of the  
features of the re (regular expression) module.
```

```
The re module has a lot of complicated and powerful features.  
You can find the official documentation here:
```

```
... https://docs.python.org/3/library/re.html  
...
```

```
import re
```

```
# We're going to look for stuff inside the following string.  
s = 'a frog, a dog, and a hog, were agog on the log in the bog'
```

```
print('This program will perform regular expression searches on the  
string:')  
print(' "{0}"'.format(s))  
print()
```

```
# Find a simple match  
regex = 'hog'  
print('Looking for "{0}"'.format(regex))  
result = re.search(regex, s)  
if result:  
    print('The first match goes from index {0} to  
{1}'.format(result.start(), result.end() - 1))  
else:  
    print('No match found.')
```

```
print()  
  
# Find a sequence of letters ending in "og".  
regex = '(\w*og)\W'  
print('Looking for "{0}"'.format(regex))  
result = re.search(regex, s)  
if result:  
    print('Found "{0}"'.format(result.groups()[0]))  
else:  
    print('No match found.')
```

```
print()  
  
# Find all the sequences of letters ending in "og".  
regex = '\w*og\b' # Any guesses why we do \b instead of \b?  
print('Looking for all occurrences of "{0}"'.format(regex))  
result = re.findall(regex, s)
```

```
if result:
    print(result)
else:
    print('No match found.')
print()

# Split the string on a pattern.
regex = '\w*og\b'
print('Splitting the string on all occurrences of
"{0}".'.format(regex))
result = re.split(regex, s)
if result:
    print(result)
else:
    print('No match found.')
print

# Replace all og-words with OGWORLD.
regex = '\w*og\b'
print('Splitting the string on all occurrences of
"{0}".'.format(regex))
print(re.sub(regex, 'OGWORD', s))
print()
```