In [11]:
```
import regex as re
# to check that a string contains only a certain set of characters.
def is_allowed_specific_char(string):
    charRe = re.compile(r'[^a-zA-Z0-9.]')
    string = charRe.search(string)
    return not bool(string)

print(is_allowed_specific_char("ABCDEFabcdef123450"))
print(is_allowed_specific_char("*&%@#!}{"))
```

```
True
False
```

In [12]:
```
# matches a string that has an a followed by zero or more b's.
def text_match(text):
        patterns = 'ab*?'
        if re.search(patterns,  text):
                return 'Found a match!'
        else:
                return('Not matched!')

print(text_match("ac"))
print(text_match("abc"))
print(text_match("abbc"))
```

```
Found a match!
Found a match!
Found a match!
```

In [40]:
```
# matches a string that has an a followed by one or more b's.
def text_match(text):
        patterns = 'ab+?'
        if re.search(patterns,  text):
                return 'Found a match!'
        else:
                return('Not matched!')

print(text_match("ab"))
print(text_match("acdefg"))
print(text_match("abbbbbcdjjruh"))
```

```
Found a match!
Not matched!
Found a match!
```

In [36]:
```
# matches a string that has an a followed by zero or one b.
def text_match(text):
        patterns = 'ab?'
        if re.search(patterns,  text):
                return 'Found a match!'
        else:
                return('Not matched!')

print(text_match("aacd"))
print(text_match("abc"))
print(text_match("abbc"))
print(text_match("aabbc"))
```

```
Found a match!
Found a match!
Found a match!
Found a match!
```

In [31]:
```
# matches a string that has an a followed by three 'b'.
def text_match(text):
        patterns = 'b{3}?'
        if re.search(patterns,  text):
                return 'Found a match!'
        else:
                return('Not matched!')

print(text_match("abbcd"))
print(text_match("aabbbbbc"))
```

```
Not matched!
Found a match!
```

In [37]:
```
# matches a string that has an a followed by two to three 'b'.
def text_match(text):
        patterns = 'ab{2,3}?'
        if re.search(patterns,  text):
                return 'Found a match!'
        else:
                return('Not matched!')

print(text_match("abbb"))
print(text_match("aabbbbbc"))
print(text_match("adverb is a part of speech"))
```

```
Found a match!
Found a match!
Not matched!
```

In [24]:
```
import regex as re
# matches a string that has an 'a' followed by anything, ending in 'b'.
def text_match(text):
        patterns = 'a.*?b$'
        if re.search(patterns,  text):
                return 'Found a match!'
        else:
                return('Not matched!')

print(text_match("any boy could develop"))
print(text_match("a crab"))
print(text_match("accddbbjjjb"))
```

```
Not matched!
Found a match!
Found a match!
```

In [22]:
```
# matches a word at the beginning of a string.
def text_match(text):
        patterns = '^\w+'
        if re.search(patterns,  text):
                return 'Found a match!'
        else:
                return('Not matched!')

print(text_match("India is a democratic country."))
print(text_match(" India is a democratic country."))
```

```
Found a match!
Not matched!
```

In [21]:
```
import regex as re
# matches a word at the end of a string.
def text_match(text):
        patterns = '\w+\S*$'
        if re.search(patterns,  text):
                return 'Found a match!'
        else:
                return('Not matched!')

print(text_match("India is a democratic country."))
print(text_match("India is a democratic country. "))
```

```
Found a match!
Not matched!
```

In [18]:
```
# to find all words that are 4 digits long in a string.
pattern = "\d{4}+"
text = "01 0132 231875 1458 301 2725"
matches = re.findall(pattern, text)
print(matches)
```

```
['0132', '2318', '1458', '2725']
```

In [20]:
```
# to find all words that are 4 digits long in a string.
pattern = "0132|1458|2725"
text = "01 0132 231875 1458 301 2725"
matches = re.findall(pattern, text)
print(matches)
```

```
['0132', '1458', '2725']
```