match_fun.py

```
import re

words = ('book', 'bookworm', 'Bible',
    'bookish','cookbook', 'bookstore', 'pocketbook')

pattern = re.compile(r'book')

for word in words:

    if re.match(pattern, word):
        print(f'The {word} matches')
```

##

fullmatch_fun.py

```
import re

words = ('book', 'bookworm', 'Bible',
    'bookish','cookbook', 'bookstore', 'pocketbook')

pattern = re.compile(r'book')

for word in words:

    if re.fullmatch(pattern, word):
        print(f'The {word} matches')
```

##

search_fun.py

```
import re

words = ('book', 'bookworm', 'Bible',
    'bookish','cookbook', 'bookstore', 'pocketbook')

pattern = re.compile(r'book')

for word in words:

    if re.search(pattern, word):
        print(f'The {word} matches')
```

```
#############################################################
dot_meta.py
import re

words = ('seven', 'even', 'prevent', 'revenge', 'maven',
    'eleven', 'amen', 'event')

pattern = re.compile(r'.even')

for word in words:
    if re.match(pattern, word):
        print(f'The {word} matches')

#############################################################
question_mark_meta.py
import re

words = ('seven', 'even','prevent', 'revenge', 'maven',
    'eleven', 'amen', 'event')

pattern = re.compile(r'.?even')

for word in words:

    if re.match(pattern, word):
        print(f'The {word} matches')
#############################################################
anchors.py
import re

sentences = ('I am looking for Jane.',
    'Jane was walking along the river.',
    'Kate and Jane are close friends.')

pattern = re.compile(r'^Jane')

for sentence in sentences:

    if re.search(pattern, sentence):
        print(sentence)
```

exact_match.py

```
import re

words = ('book', 'bookworm', 'Bible',
    'bookish','cookbook', 'bookstore', 'pocketbook')

pattern = re.compile(r'^book$')

for word in words:

    if re.search(pattern, word):
        print(f'The {word} matches')
```

##

character_class.py

```
import re

words = ('a gray bird', 'grey hair', 'great look')

pattern = re.compile(r'gr[ea]y')

for word in words:

    if re.search(pattern, word):
        print(f'{word} matches')
```

###

named_character_class.py

```
import re

text = 'We met in 2013. She must be now about 27 years old.'

pattern = re.compile(r'\d+')

found = re.findall(pattern, text)

if found:
    print(f'There are {len(found)} numbers')
```

```
############################################################
case_insensitive.py
import re

words = ('dog', 'Dog', 'DOG', 'Doggy')

pattern = re.compile(r'dog', re.IGNORECASE)

for word in words:
    if re.match(pattern, word):
        print(f'{word} matches')
################################################
alternations.py

import re

words = ("Jane", "Thomas", "Robert",
    "Lucy", "Beky", "John", "Peter", "Andy")

pattern = re.compile(r'Jane|Beky|Robert')

for word in words:

    if re.match(pattern, word):
        print(word)

#####################################################
finditer_fun.py
import re

text = 'I saw a fox in the wood. The fox had red fur.'

pattern = re.compile(r'fox')

found = re.finditer(pattern, text)

for item in found:

    s = item.start()
    e = item.end()
    print(f'Found {text[s:e]} at {s}:{e}')
```

```
##########################################
capturing_groups.py
import re

content = '''<p>The <code>Pattern</code> is a compiled
representation of a regular expression.</p>'''

pattern = re.compile(r'(</?[a-z]*>)')

found = re.findall(pattern, content)

for tag in found:
    print(tag

##############################################
emails.py
import re

emails = ("luke@gmail.com", "andy@yahoocom",
    "34234sdfa#2345", "f344@gmail.com")

pattern = re.compile(r'^[a-zA-Z0-9._-]+@[a-zA-Z0-9-]+\.[a-zA-Z.]{2,18}$')

for email in emails:

    if re.match(pattern, email):
        print(f'{email} matches')
    else:
        print(f'{email} does not match')
```