Exercise of Chapter 6

1) Quick Check: Split and Join

Look at the code below:

```
>>> x = 'this is a test'
>>> '-'.join(x.split())
```

The result would be:

```
'this-is-a-test'
```

2) Quick Check: Strings to Numbers

Numbers 1, 2, and 3 have at least one character that can’t be reversed to an integer, so the answer is number 3.

3) Quick Check: Strip

Number 4 will omit the comma, the newline, and the parentheses.

4) Quick Check: String Searching

Let’s use the endswith method:

```
 endswith('rejected')
```

5) Quick Check: Modifying Strings

As mentioned on my posts in linkedin and telegram, we could use the maketrans() and translate methods. Look at the coe below:

```
>>> symbols = str.maketrans(',!?', '   ')
>>> sentence = "Hello, World! How you doing?"
>>> sentence.translate(symbols)
```

6) Try This: String Operations

Look at the codes below:

Code #1

```
>>> for items in x:
        print(items.strip(""))
```

Code #2

```
>>> word = 'Mississippi'
>>> position = word.rfind('p')

>>> word = word[:position] + word[position + 1:]
>>> print(word)
```

And the result would be:

```
Mississipi
```

7) Quick Check: The format method

The output is: `'    1:$$$$3'`

8) Quick Check: Formatting strings with %

Look at the code below:

```
x1 = "%.2f" % 1.1111
x2 = "%(a).2f" % {'a':1.1111}
x3 = "%(a).08f" % {'a':1.1111}
print(x1)
print(x2)
print(x3)
```

Now, look at each result:

```
1.11
1.11
1.11110000
```

9) Quick Check: Bytes

I) Due to being binary, bytes are better choice.
II) Due to being unicode, strings are better choice.
III) Due to using Python version 3, strings are better choice.
IV) Due to the limited length of bytes (255), they are better choice.

10) Lab 6: Preprocessing Text

```
with open("moby_01.txt") as infile, open("moby_01_clean.txt", "w") as
  outfile:
    for line in infile:
        outfile.write(cleaned_words)
symbols = str.maketrans(', !?',  '   ')
```

```
with open("moby_01.txt") as infile, open("moby_01_clean.txt", "w") as
  outfile:
    for line in infile:
        cleaned_line = line.lower()
        cleaned_line = cleaned_line.translate(symbols)
```

```
words = cleaned_line.split()
cleaned_words = '\n'.join(words)

outfile.write(cleaned_words)
```