Regular Expressions: Takeaways

Syntax

- To start using regex, use the `re` module.

WILDCARDS

- To indicate that *any* character can be put in its place:

```
strings = ["bat", "robotics", "megabyte"]

regex = "b.t"
```

BEGINNINGS AND ENDINGS OF STRINGS

- To match all strings that start with `"a"`, use `"^a"`.
- To match all strings that end with `"a"`, use `"a$"`.

COUNTING MATCHES WITHIN THE DATASET

- To check whether "needle" is a match for "haystack".
    - Input:

```
if re.search("needle", "haystack") is not None:
    print("We found it!")
else:
    print("Not a match")
```

    - Output:

```
Not a match
```

MATCHING MULTIPLE CHARACTERS

- To match multiple characters, specify the characters between "[ ]":

```
`"[bcr]at"`
```

- This expression would match "bat", "cat", and "rat".

ESCAPING SPECIAL CHARACTERS

- To escape a character use "\":

```
for row in posts:

    if re.search("\[Serious\]", row[0]) is not None:

        serious_count += 1
```

COMBINING REGEX CHARACTERS

- Checking if our code has either "[Serious]" or "[serious]":

```
serious_count = 0

for row in posts:

    if re.search("\[[Ss]erious\]", row[0]) is not None:

        serious_count += 1
```

- To match either one character or another, use "|":

ADDITIONAL REGEX

- To substitute strings, use sub():

```
re.sub("yo", "hello", "yo world")
```

- To match years, use:

```
"[1-2][0-9][0-9][0-9]"
```

- To repeat characters, use "{ }". To repeat the pattern "[0-9]" four times:

```
`"[0-9]{4}"`
```

Concepts

- A **regular expression** (regex) is a sequence of characters that describes a search pattern. We can use regular expressions to search for and extract data.
- In regular expressions, escaping a character means indicating that you don't want the character to do anything special
    - The `re` module provides a sub() function that takes the following parameters (in order):
    - `pattern` : The regex to match
    - `repl` : The string that should replace the substring matches
    - `string` : The string containing the pattern we want to search

Resources

- Python Documentation on re
- Python Documentation on re.search