Regular Expressions

What are Regular Expressions?

- Patterns for matching text
- Extremely powerful tool for doing things like:
  - searching logs
    - counting the number of errors
  - enforcing input
    - limiting a phone number input to only numbers

Matching Letters

Given the string "abc"

All of the following are valid regex patterns to match this string:

- a
- b
- c
- ab
- bc
- abc

Matching Numbers

Given the string "123"

All of the following are valid regex patterns to match this string:

- 1
- 2
- 3
- 12
- 23
- 123

"123", "abc"

The shortcut for matching any number is \d

This will match the first string, but not the second.

Matching Characters

Given the string "abc123"

Lots of ways to match this string. We're able to combine characters and numbers.

- ab
- c1
- 12

Wild Card Character

"." is a special character in Regex

Given the strings "abc123." and "abc123"

The pattern . will only match the first string.

\. will match any character.

Specific Characters

[] notation

Square Brackets can be used when you only want to match specific characters and no others.

"cat", "rat", "bat"

Using bracket notation, we can match the first two, but not the third with the following pattern [cr]

Exclude Characters

^ (caret)

The ^ character can be used to exclude specific characters when combined with []

"cat", "rat", "bat"

Using the caret character we can make the same match with [^b]at

Character Ranges

[a-z], [0-9], [n-p]

By using the dash symbol, we can match on ranges of characters without having to list the entire range.

"AaA", "BbB", "A1A"

[a-zA-Z0-9]

The above pattern will match any alphanumeric character.

The shortcut for this is \w

Character Repetitions

{}

By using a number surrounded by {}, we can specify the number of repetitions to match.

"hello", "goodbye"

We can match for two o's with the following pattern o{2}

Combining Repetitions

[ab]{2} matches any two occurrences of "a" and/or "b"

"ab", "aa", "bb"

Repetition Shortcuts

+

One or more of a character

[ab]+ will match "aab" and "abb"

*

Zero or more of a character

[ab]* will match "ab", and "aab"

Optional Character

?

? denotes a character may or may not be present.

a?c will match "abc" and "ac"

Starting and Ending

^ and $

^ is the symbol for the start of a string.

$ is the symbol for the end of a string.

"Hello World", "Goodbye Hello"

To match only the first string, we can use the pattern ^Hello to only match the string that starts with "Hello"

"Hello World", "Goodbye Hello"

To match only the second string, we can use the pattern Hello$ to only match the string that ends with "Hello"

"Hello World", "Goodbye Hello"

We can also match the entire string by combining both of these with

ˆHello World$

Whitespace

\s and \S

\s will match any whitespace character

- \n newline
- \t tab
- space

\S will match any non whitespace character

Regex and Javascript

```
var re1 = new RegExp('abc');

var re2 = /abc/
```

Flags

g -> Global, will continue to search after the first match

i -> Case insensitive

Flags

```
var re1 = new RegExp('abc', 'g');

var re2 = /abc/g
```