Lesson | Wednesday

Introduction to Programming (/introduction-to-programming) / Arrays and Looping (/introduction-to-programming/arrays-and-looping) / Further Exploration: Introduction to Regular Expressions

Text | Cheat sheet

Terminology

Regular expression: Also known as a **regex**, a regular expression is a set of characters we can use to find patterns in a string. The set of characters is enclosed in `/ /` and may include flags after the second slash.

Methods That Use Regular Expressions

- **`String.prototype.replace()`** : Takes two arguments — the first is a regular expression, the second is what the pattern should be replaced by.
- **`String.prototype.match()`** : Takes a regular expression as an argument and then returns an array with all matches.
- **`RegExp.prototype.test()`** : Takes a string as an argument — the regular expression is the receiver — and returns a boolean if the string contains the pattern.

Regex Characters

- `\d` : Numbers
- `\D` : Not numbers
- `\w` : Matches any alphanumeric character (including underscores) — so numbers and letters
- `\W` : Matches any character that's not a number, letter or underscore
- `\s` : Matches a whitespace character
- `\S` : Matches any non-whitespace character
- `.` : Any single character (wildcard)
- `^` : *Not* this pattern

Regex Flags

Regex flags come after the second slash in a regular expression. For instance: `/cat/gi` .

- `g` is the global flag. Without this flag, regular expressions usually just find the first matching pattern in the string. With this flag, the regex will find *all* matching patterns in the string.
- `i` is the case insensitivity flag. When it's added, the regular expression will ignore case sensitivity.

Regex Groups and Ranges

- `[ ]` denotes that all characters inside the brackets should be considered a matching pattern. For instance, the pattern `/[aieou]/` will match any vowels in a string.
- `-` denotes a range of characters. For instance, the pattern `/[0-9]/` denotes all numerical digits. `[A-Z]` and `[a-z]` are other common ranges.

Regex Quantifiers

- `+` : Match the preceding character one or more times
- `*` : Match the preceding character zero or more times
- `?` : Match the preceding character zero or one times

- `{x}` : Match the pattern `x` number of times
- `{x,}` : Match the pattern at least `x` times
- `{x,y}` : Match the pattern at least `x` but no more than `y` times

Other Helpful Regex Symbols

- `|` : Represents or. For example, `/cat|dog/` states match either `"cat"` *or* `"dog"`
- `\b` : Denotes a pattern boundary. Can be used at beginning or end of a pattern. For example, `/\bcat\b/` represents an *exact* match with "cat" — and doesn't match with "cathedral".

Documentation

- Regular expression syntax cheatsheet. (https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Regular_Expressions/Cheatsheet)
- MDN guide to regular expressions. (https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Regular_Expressions)
- Regex Crossword (https://regexcrossword.com/) is a fun site for learning about regular expressions.
- Finally, it's very common to use regex generators that make it easier to get the regex we need to get the job done. A quick Google search will reveal many out there! Here's just a few to optionally check out:
  - https://regexr.com/ (https://regexr.com/)
  - https://regex-generator.olafneumann.org/ (https://regex-generator.olafneumann.org/)
  - https://regex101.com/ (https://regex101.com/)

Previous (/introduction-to-programming/arrays-and-looping/practice-pig-latin)
Next (/introduction-to-programming/arrays-and-looping/further-exploration-regular-expressions-with-text-analyzer)

disable dark mode

(http://www.epicodus.com)