17. Computer Science for JavaScript Challenge: Regex Tutorial

Start Assignment

Due Mar 27 by 11:59pm **Points** 100

Submitting a text entry box or a website url

Your Task

Developers write code, but they also write *about* code. Take a moment to search the web for tutorials about any of the subjects you've learned so far in this course. You're likely to find thousands of tutorials written by developers of all skill levels, but especially by junior developers—like you!

Your Challenge this week is to create a tutorial that explains how a specific regular expression, or regex, functions by breaking down each part of the expression and describing what it does.

Before you start, clone the **starter code** (https://github.com/coding-boot-camp/bug-free-goggles) .

User Story

```
AS A web development student
I WANT a tutorial explaining a specific regex
SO THAT I can understand the search pattern the regex defines
```

Acceptance Criteria

```
GIVEN a regex tutorial
WHEN I open the tutorial
THEN I see a descriptive title and introductory paragraph exp
WHEN I click on the links in the table of contents
THEN I am taken to the corresponding sections of the tutorial
WHEN I read through each section of the tutorial
THEN I find a detailed explanation of what a specific compone
WHEN I reach the end of the tutorial
THEN I find a section about the author and a link to the auth
```

What Is a Regex?

A **regex**, which is short for **regular expression**, is a sequence of characters that defines a specific search pattern. When included in code or search algorithms, regular expressions can be used to find certain patterns of characters within a string, or to find and replace a character or sequence of characters within a string. They are also frequently used to validate input.

For example, the following regular expression can be used to verify that user input is a valid email address:

```
/^([a-z0-9_\.-]+)@([\da-z\.-]+)\.([a-z\.]{2,6})$/
```

Each component of this regex has a unique responsibility to make sure that a user enters an email address that begins with an unspecified number of characters preceding the `@` symbol, followed by a domain.

Before you get started, watch this **introduction to regular expressions video (https://youtu.be/7DG3kCDx53c)** and read this **Regex Tutorial (https://medium.com/factory-mind/regex-tutorial-a-simple-cheatsheet-by-examples-649dc1c3f285)** to learn how to identify the different components that make up a regex. If you need any additional help, there are many resources on the web. Feel free to do your own research to find one that can help you complete this Challenge.

Once you have a better understanding of what these different parts of a regular expression do, you'll need to explain what they do for a specific regex.

You can choose one of the following regular expressions or you can choose one that you found on your own (with the exception of the one that was covered in the virtual classes, Matching a Username):

- Matching a Hex Value – `/^#?([a-f0-9]{6}|[a-f0-9]{3})$/`
- Matching an Email – `/^([a-z0-9_\.-]+)@([\da-z\.-]+)\.([a-z\.]{2,6})$/`
- Matching a URL – `/^(https?:\/\/)?([\da-z\.-]+)\.([a-z\.]{2,6})([\/\w \.-]*)*\/?$/`
- Matching an HTML Tag – `/^<([a-z]+)([^<]+)*(?:>(.*)<\/\1>|\s+\/>)$/`

Getting Started

Instead of creating a repository, you'll publish a GitHub **gist**. GitHub describes a gist as a simple way to share code snippets with others. It's also an ideal way to demonstrate a technique, teach a principle, or show off a solution. It functions just like a repository, and you'll use Markdown to create it, just as you do with your READMEs. Gists can include code, images, links, and anything else you can include in a README.

After you've cloned the starter code, learn **how to create a gist (https://help.github.com/en/github/writing-on-github/creating-gists)** . You can also watch this video on **how to use gists (https://www.youtube.com/watch?v=wc2NlcWjQHw)** .

NOTE

> Make sure to create a **public** gist.

The starter code is a template with a title, introductory paragraph, summary, and table of contents. The table of contents should link to sections of the tutorial that describe the functionality of each component in the regex. Be sure to rename the template to a unique name that describes your tutorial.

NOTE

> The regular expression that you choose may not include all of the components outlined in the starter code. After you've finished your walkthrough, you can remove any sections that you didn't use.

Each section that describes a component should include more than just one sentence explaining what it does. It should also include a code snippet of that particular component and some examples that meet the requirements of that component.

IMPORTANT

> Make revisions to your gist in the GitHub gist UI. This will create a revision history that graders can use to verify that the tutorial content is yours.

Grading Requirements

NOTE

> If a Challenge assignment submission is marked as “0”, it is considered incomplete and will not count towards your graduation requirements. Examples of incomplete submissions include the following:
>
> - A repository that has no code
> - A repository that includes a unique name but nothing else
> - A repository that includes only a README file but nothing else
> - A repository that only includes starter code

This Challenge is graded based on the following criteria:

Deliverables: 30%

- A valid URL of your GitHub gist.

- Your GitHub gist that contains the tutorial Markdown.

Technical Acceptance Criteria: 50%

- Satisfies all of the above acceptance criteria plus the following:

  - Revisions to the tutorial must be made in the GitHub gist UI so that graders have access to your revision history.

  - The tutorial must cover either one of the regex examples listed above or another of your choice. You may NOT use the regex covered in your virtual classes (Matching a Username).

  - The tutorial must include sections that correspond to each of the components that make up the regex. You may not need to use all of the sections included in the starter code, but you should include all of the sections that correspond to the different components of the regex you chose.

  - Each section that describes a component must include more than just one sentence explaining what it does. It's okay to use online resources for assistance, but do not copy and paste; explain each component in your own words and be thorough.

  - Each section that describes a component must include a code snippet of that particular component. Use backticks to display your code snippets in Markdown.

  - Each section that describes a component must include at least one example that meets the requirements of that component.

Tutorial Clarity and Quality: 20%

- Tutorial provides a clear explanation of how the regex works. Be as concise as possible.
- Tutorial describes all of the components of the regex, each component in a separate section.

How to Submit the Challenge

You are required to submit the following for review:

- The URL of the GitHub gist. Give the gist a unique name.

NOTE

> You are allowed to miss up to two Challenge assignments and still earn your certificate. If you complete all Challenge assignments, your lowest two grades will be dropped. If you wish to skip this assignment, you may move on to the next module.

Comments are disabled for graded submissions in BootCamp Spot. If you have questions about your feedback, please notify your instructional staff or the Student Success Manager. If you would like to resubmit your work for an improved grade, you can use the Resubmit Assignment button to upload new links. You may resubmit up to three times for a total of four submissions.