

The CSS Rockstar

DESCRIPTION

On your own, construct a page featuring the lyrics of a song that is particularly meaningful to you, featuring appropriate typography and layout applied using CSS. Include a short paragraph or two about the song and perhaps a bit about the artist or band.

Use appropriate page-level sectioning elements to structure the page, and semantically-meaningful elements to mark up the content. Select two (or at the most, three) typefaces that would be suitable for the lyrics. Employ a variety of CSS selector types to add visual styles to the page. Feature the font itself prominently on the document and use CSS-based text-sizing techniques and other typographic styles to make the page attractive.

INSTRUCTIONS

1. Select a song that is meaningful to you and obtain the official lyrics.
2. *Create a local code repository using Git (or add a "project-1" folder into your class files repository).*
3. Build a valid, well-structured HTML page for the song lyrics, along with a paragraph about the song and another regarding the artist or the band.
4. Use appropriate *sectioning* elements to structure the page (like article, section, header, main, footer, aside) and other semantically-meaningful elements to mark up the content (like headings, paragraphs, lists, blockquotes, quotes, etc.).
5. Validate your HTML to be sure there are no errors at <https://validator.w3.org/>.
6. Apply a CSS *reset* or *normalization* technique that also addresses *box-sizing*.
7. Select *two* typefaces that would be suitable for the lyrics - one for the page headings, and another for the page body. Be sure to consider readability—display fonts are designed for headings, while other fonts were made for large blocks of text (called body copy).
8. Use either CSS Flexbox or Grid (or a combination of both) to build a simple but effective layout for the page.
9. Employ the typographic techniques learned in this class to style the text using the two typefaces that you have selected (one or both may not be Web-safer fonts, so you may need to use a font-service like Google Fonts).
10. Add some suitable colours and other design elements to the page, but consider legibility—make sure that the page is easy to read and that there are no contrast issues).

11. Validate your CSS to be sure there are no errors at <https://jigsaw.w3.org/css-validator/>.
12. *Be sure to include regular Git commits describing each stage of your work from start to finish so that the development process is well-documented (this is important). Push all changes to a remote repository (like Github).*
13. Include some fine-print at the bottom of the page with links to any resources referenced in the creation of your assignment.
14. *No 3rd-party CSS themes or frameworks are permitted. All CSS and HTML must be your own work.*
15. Only use CSS and HTML techniques that we have learned in class - code beyond the scope of the course content may be subject to a code review. **Ensure that all work submitted adheres to Georgian College's Academic Integrity policies and procedures**, available at <https://cat.georgiancollege.ca/academic-regulations/integrity/>

TAKE IT FURTHER

1. Apply some suitable or background-images to the page.
2. Try to avoid using any classes or ids at all in your HTML, relying instead on advanced CSS selectors.
3. Apply some interesting CSS-powered text-effects (for display type, not body copy).
4. Anything else you can think of to make the page more fun.

SUBMITTING YOUR WORK

1. Upload your completed page and all the supporting files to a Web server and provide the URL (you can enable Github Pages and provide the URL to your Project 1 page if you'd like).
2. Provide the URL to your remote hosted code repository (like Github).

EVALUATION

Please refer to the attached rubric for individual category scores.