The goal of this project is to give you an opportunity to show off your strengths. Our prompt below is intentionally open-ended: we want to see what you're capable of building.

Project

Build "Snake", the classic arcade game in which a player controls a snake, guiding it towards food and away from walls or itself. The game should listen to the arrow keys, and the snake should grow as it consumes food. When the snake collides with itself or a wall, the game should end.

Project Rubric

- **Try to spend about three hours on your project**. After three hours, feel free to describe next steps in a readme. Focus on writing a small amount of excellent code, rather than a large amount of poor code.
- Focus on the front end. A strictly client-side app is preferable, a basic server is permissible.
- Play to your strengths. Given the time goal, prioritize the parts of the project that show off your talents. After you've hit three hours, feel free to put your next steps in your readme.
- **Submit your project as a .zip, including a readme with build/usage instructions**. If you decide to host your app somewhere, provide a link to the demo in your readme
- Please write vanilla javascript. Do not use any frameworks or component library. If it would make you feel more comfortable, feel free to use Lodash/underscore and/or a module loader of your choice.
- ES6 is preferred.

Your submission will be evaluated based on the following metrics:

Code Cleanliness: Write clean code that adheres to standard conventions. Be consistent and intentional with naming and documentation. Write legible, extensible, DRY code. Comment where appropriate. Demonstrate skillful use of advanced language features.

Program Structure & Design: Sensibly divide your code into modules and files. Keep module interfaces simple and easy to understand. Abstract functionality where appropriate, and use appropriate data structures for storing data. Maintain a logical directory structure. Constants should be separated out, functions should have singular purposes.

Program Robustness: Neither your code nor your app should be fragile. Edge cases should be considered and handled appropriately.

UI/UX Design: From a user standpoint, the final product should be polished, functional, and intuitive.

Performance: Your app should adhere to best web development practices, and architectural decisions should reflect an understanding of how the browser renders content.

The following will not affect the evaluation of your submission:

Cross-browser compatibility: While important for production code, modern development practices can generally transpile and polyfill missing features for non-modern browsers. Therefore, for this project, any app that runs well in the latest version of Chrome or Firefox is acceptable.

Relevance to Squarespace: Your project will be evaluated on its own merits. There is no need for it to relate to any of Squarespace's products, and incorporating any Squarespace products, designs, or patterns will not improve your project's evaluation.

Code Style: Your code *will* be evaluated on its consistency, readability, and cleanliness. Your specific stylistic decisions, however, will not impact your evaluation unless they are inconsistently implemented. For example, you may choose to declare multiple variables on a line or declare them separately, as long as you do so consistently throughout your app.