Word Masters

Status	In progress
Assign	

Javascript

Break into smaller piece and solve one by one.

Lets' handle user interaction (frontend) first and then we will handle backend and validation later.

User side (typing etc.)

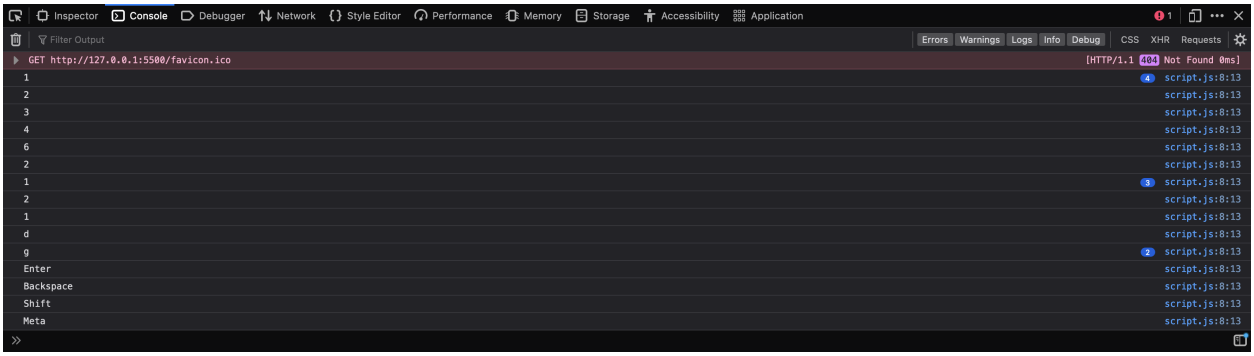
Let's grab everything we need from the DOM first.

```
// Let's query the DOM and get things we need first
const letters = document.querySelectorAll(".letter");
const loadingDiv = document.querySelector(".info-bar");
```

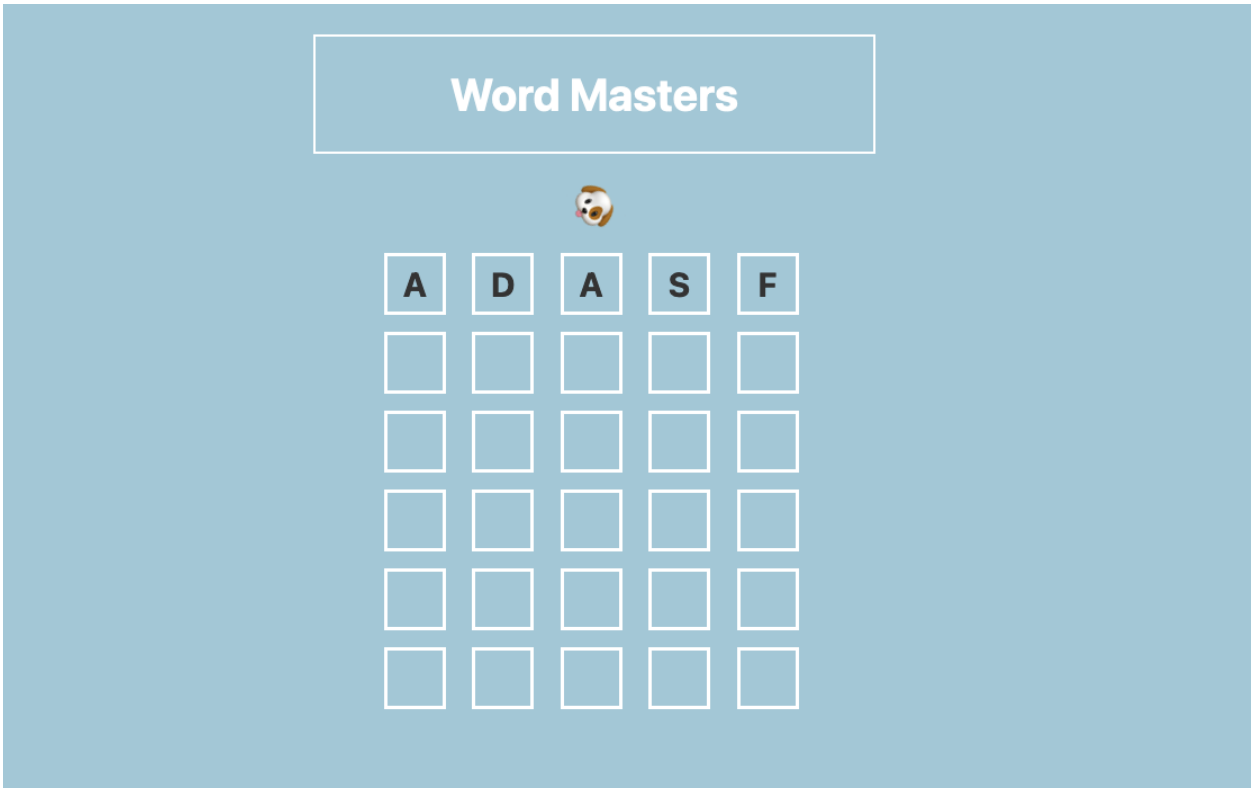
We will handle *key press* first.

```
async function init() {
  document.addEventListener("keydown", function handleKeyPress(event) {
    const action = event.key;
    console.log(action);
  });
}
init();
```

Let's check if it is working.



Add text to the boxes



Change `addLetter` logic to handle new rows.

```
// Write char in the box
    letters[ANSWER_LENGTH * currentRow + currentGuess.length - 1].innerText =
      letter;
  }
```

Backend and API

Let’s grab the word of the day first.

```
async function init() {
  let currentGuess = "";
  let currentRow = 0;

  // Getting word of the day
  const res = await fetch("https://words.dev-apis.com/word-of-the-day");
  const resObj = await res.json();
  const word = resObj.word.toUpperCase();
  console.log(word);
```

Now let’s work on the spinner. Once the page is loaded, we want to stop or hide the spinner.

```
function setLoading(isLoading) {
    // if loading, then show loading spiral
    loadingDiv.classList.toggle("show", isLoading); // shortcut for classList.add or remove
  }
```

We stop the spiral when word is grabbed.

```
const resObj = await res.json();
  const word = resObj.word.toUpperCase();
  setLoading(false);
```

Also switched the logic from `hidden` to `show`.

```
.info-bar {
  visibility: hidden;
  display: flex;
  align-items: center;
  justify-content: center;
}

.show {
```

```
  visibility: visible;
}
```

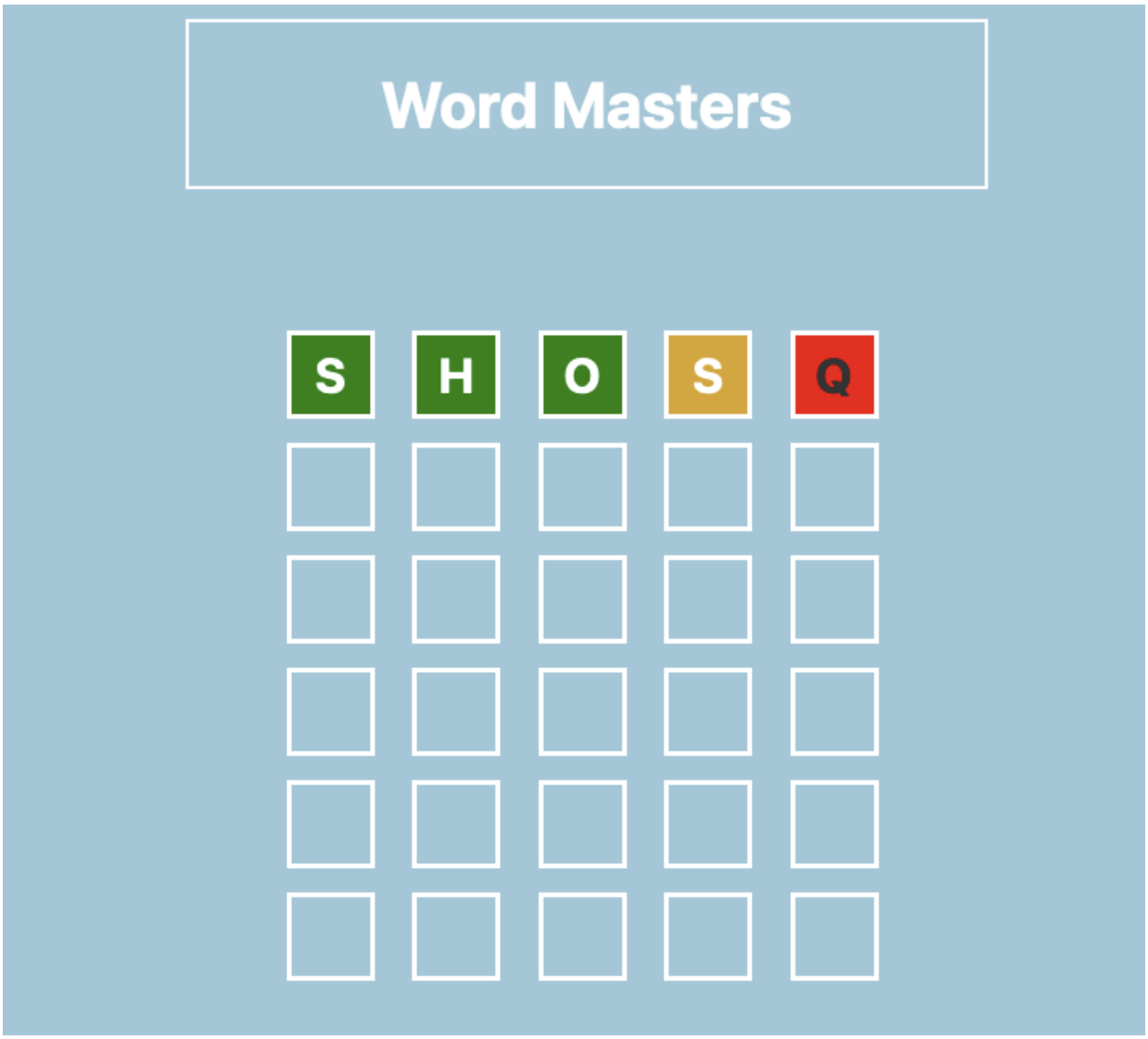
Remember that the order matters in CSS because of cascades.

Implementing the correct, wrong and close logic next. Will compare letters from guess with original word letters.

```
for (let i = 0; i < ANSWER_LENGTH; i++) {
      // mark as correct
      if (guessParts[i] === wordParts[i]) {
        letters[currentRow * ANSWER_LENGTH + i].classList.add("correct");
      }
    }

    for (let i = 0; i < ANSWER_LENGTH; i++) {
      if (guessParts[i] === wordParts[i]) {
        // do nothing, we already did it
      } else if (
        wordParts.includes(guessParts[i]) /* TODO make this more accurate */
      ) {
        letters[currentRow * ANSWER_LENGTH + i].classList.add("close"); // correct but wrong place
      } else {
        letters[currentRow * ANSWER_LENGTH + i].classList.add("wrong");
      }
    }
```

Now the issue is that it shows `close` even if we already have the `correct` one.



Solution is to count the exact number of occurrences for each letter.

Next we add Win and Lose logic. I you run out of guesses (6) then you lose. If guessed word is same as the target word then you win.

Now we validate the word entered by the user.

Notes

```
// If we do not know that server will give us json
// JSON.parse() only works with JSON strings
response.text() + JSON.parse(reponse)

// If we know
response.json() // already an object
```

Links

Event delegation and bubbling

https://www.freecodecamp.org/news/event-delegation-javascript/

Destructuring assigment

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/Destructuring_assignment