

Hold Shift and Check Checkboxes

Explanation

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8" />
  <title>Hold Shift to Check Multiple Checkboxes</title>
  <link rel="icon" href="https://fav.farm/0" />
</head>

<body>
  <style>
    html {
      font-family: sans-serif;
      background: #ffc600;
    }

    .inbox {
      max-width: 400px;
      margin: 50px auto;
      background: white;
      border-radius: 5px;
      box-shadow: 10px 10px 0 rgba(0, 0, 0, 0.1);
    }

    .item {
      display: flex;
      align-items: center;
      border-bottom: 1px solid #f1f1f1;
    }

    .item:last-child {
      border-bottom: 0;
    }

    input:checked+p {
      background: #f9f9f9;
      text-decoration: line-through;
    }

    input[type="checkbox"] {
      margin: 20px;
    }
  </style>
</body>
```

```
p {
  margin: 0;
  padding: 20px;
  transition: background 0.2s;
  flex: 1;
  font-family: "helvetica neue";
  font-size: 20px;
  font-weight: 200;
  border-left: 1px solid #d1e2ff;
}
</style>
<!--
  The following is a common layout you would see in an email client.
-->
```

When a user clicks a checkbox, holds Shift, and then clicks another checkbox a few rows down, all the checkboxes inbetween those two checkboxes should be checked.

```
-->
<div class="inbox">
  <div class="item">
    <input type="checkbox" />
    <p>This is an inbox layout.</p>
  </div>
  <div class="item">
    <input type="checkbox" />
    <p>Check one item</p>
  </div>
  <div class="item">
    <input type="checkbox" />
    <p>Hold down your Shift key</p>
  </div>
  <div class="item">
    <input type="checkbox" />
    <p>Check a lower item</p>
  </div>
  <div class="item">
    <input type="checkbox" />
    <p>Everything in between should also be set to checked</p>
  </div>
  <div class="item">
    <input type="checkbox" />
    <p>Try to do it without any libraries</p>
  </div>
  <div class="item">
    <input type="checkbox" />
    <p>Just regular JavaScript</p>
  </div>
```

```
<div class="item">
  <input type="checkbox" />
  <p>Good Luck!</p>
</div>
<div class="item">
  <input type="checkbox" />
  <p>Don't forget to tweet your result!</p>
</div>
</div>

<script>
  const checkboxes = document.querySelectorAll(
    '.inbox input[type="checkbox"]'
  );
  console.log(checkboxes);

  let lastChecked;

  function handleCheck(e) {
    //checks if shift key is down and if a checkbox is being checked

    let inBetween = false;
    if (e.shiftKey && this.checked) {
      //loop over every single checkbox in between first and last checked
      checkbox
      checkboxes.forEach((checkbox) => {
        console.log(checkbox);
        if (checkbox === this || checkbox === lastChecked) {
          inBetween = !inBetween;
          console.log("These are checked in between");
        }
        if (inBetween) {
          checkbox.checked = true;
        }
      });
    }

    lastChecked = this;
  }

  checkboxes.forEach(checkbox => checkbox.addEventListener('click',
  handleCheck));
</script>
</body>

</html>
```

The challenge here is to create an interactive checkbox list where users can select multiple checkboxes efficiently using the Shift key. The functionality is similar to what you might find in an email client, where selecting one checkbox, holding the Shift key, and then selecting another checkbox results in all checkboxes in between also being selected.

Below is a brief explanation of the script and the exercises performed:

The script essentially serves as a playground for demonstrating various console methods and tricks for debugging and logging information in the browser's console. It covers styling console output, warnings, errors, assertions, grouping, counting, timing, and displaying tabular data. The exercises are meant to showcase the versatility of console methods for debugging and logging in JavaScript.

1. Selection of Checkboxes:

The script selects all checkboxes within a div with the class "inbox" using `document.querySelectorAll('.inbox input[type="checkbox"]')`. This Node List of checkboxes is stored in the checkboxes variable.

2. Event Handling:

A click event listener is attached to each checkbox. When a checkbox is clicked, the handleCheck function is called.

3. Shift Key Functionality:

Inside the handleCheck function, there is logic to handle the case where the Shift key is held down while a checkbox is clicked. This logic checks if the Shift key is pressed (`e.shiftKey`) and if the checkbox clicked is being checked (`this.checked`).

4. Setting the Checkboxes:

To check all checkboxes between the first and last selected checkboxes, the script uses a flag variable `inBetween`. When iterating over all checkboxes, the script toggles this flag to true when it finds either the first or the last selected checkbox. All checkboxes encountered while `inBetween` is true are then checked.

5. Remembering the Last Checked Box:

The variable `lastChecked` is used to store the last checkbox that was clicked. This is essential for determining the range of checkboxes to be checked when the Shift key is used.

6. Debugging with Console Logs:

The script includes `console.log` statements, which are likely used for debugging purposes to track the flow of the function and see which checkboxes are being processed.

What I have learned

From this challenge, I have learned how to implement a feature that allows users to efficiently select multiple checkboxes using the Shift key.