

# Dev Tools Domination

## Explanation

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Console Tricks!</title>
  <link rel="icon" href="https://fav.farm/✓" />
</head>
<body>

  <p onClick="makeGreen()">xBREAKxDOWNx</p>

  <script>
    const dogs = [{ name: 'Snickers', age: 2 }, { name: 'hugo', age: 8 }];

    function makeGreen() {
      const p = document.querySelector('p');
      p.style.color = '#BADA55';
      p.style.fontSize = '50px';
    }

    // Regular
    console.log('hello');

    // Interpolated
    console.log('Hello I am a %s string!', '💩');

    // Styled
    // console.log('%c I am some great text', 'font-size:50px; background:red; text-shadow: 10px 10px 0 blue');

    // warning!
    console.warn('OH NOOO');

    // Error :|
    console.error('Shit!');

    // Info
    console.info('Crocodiles eat 3-4 people per year');

    // Testing
    const p = document.querySelector('p');
```

```
console.assert(p.classList.contains('ouch'), 'That is wrong!');

// clearing
console.clear();

// Viewing DOM Elements
console.log(p);
console.dir(p);

console.clear();

// Grouping together
dogs.forEach(dog => {
  console.groupCollapsed(` ${dog.name} `);
  console.log(` This is ${dog.name}`);
  console.log(` ${dog.name} is ${dog.age} years old`);
  console.log(` ${dog.name} is ${dog.age * 7} dog years old`);
  console.groupEnd(` ${dog.name} `);
});

// counting

console.count('Wes');
console.count('Wes');
console.count('Steve');
console.count('Steve');
console.count('Wes');
console.count('Steve');
console.count('Wes');
console.count('Steve');
console.count('Steve');
console.count('Steve');
console.count('Steve');
console.count('Steve');

// timing
console.time('fetching data');
fetch('https://api.github.com/users/wesbos')
  .then(data => data.json())
  .then(data => {
    console.timeEnd('fetching data');
    console.log(data);
  });

console.table(dogs);

</script>
</body>
</html>
```

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. Event Handling:**

The `<p>` element has an `onClick` attribute, calling the `makeGreen()` function when clicked.

**2. Styling Console Output:**

Various `console.log` methods demonstrate different logging styles, including regular text, interpolated strings, and styled logging (commented out).

**3. Console Methods:**

`console.warn`, `console.error`, and `console.info` demonstrate warning, error, and info messages.

`console.assert` checks if the `<p>` element contains the class 'ouch' and logs a message if the assertion fails.

**4. Console Clearing:**

`console.clear()` is used to clear the console.

**5. Inspecting DOM Elements:**

`console.log` and `console.dir` are used to log and inspect the `<p>` element.

**6. Grouping Console Output:**

`console.groupCollapsed` and `console.groupEnd` group console output for each dog in the `dogs` array.

**7. Counting:**

`console.count` counts occurrences of 'Wes' and 'Steve' in the console.

**8. Timing:**

`console.time` and `console.timeEnd` measure the time taken to fetch data from a GitHub API.

**9. Tabular Data:**

`console.table` displays the `dogs` array as a table in the console.

## **What I have learned**

From this challenge, I have learned various techniques for effective debugging and logging in JavaScript through the browser's console.