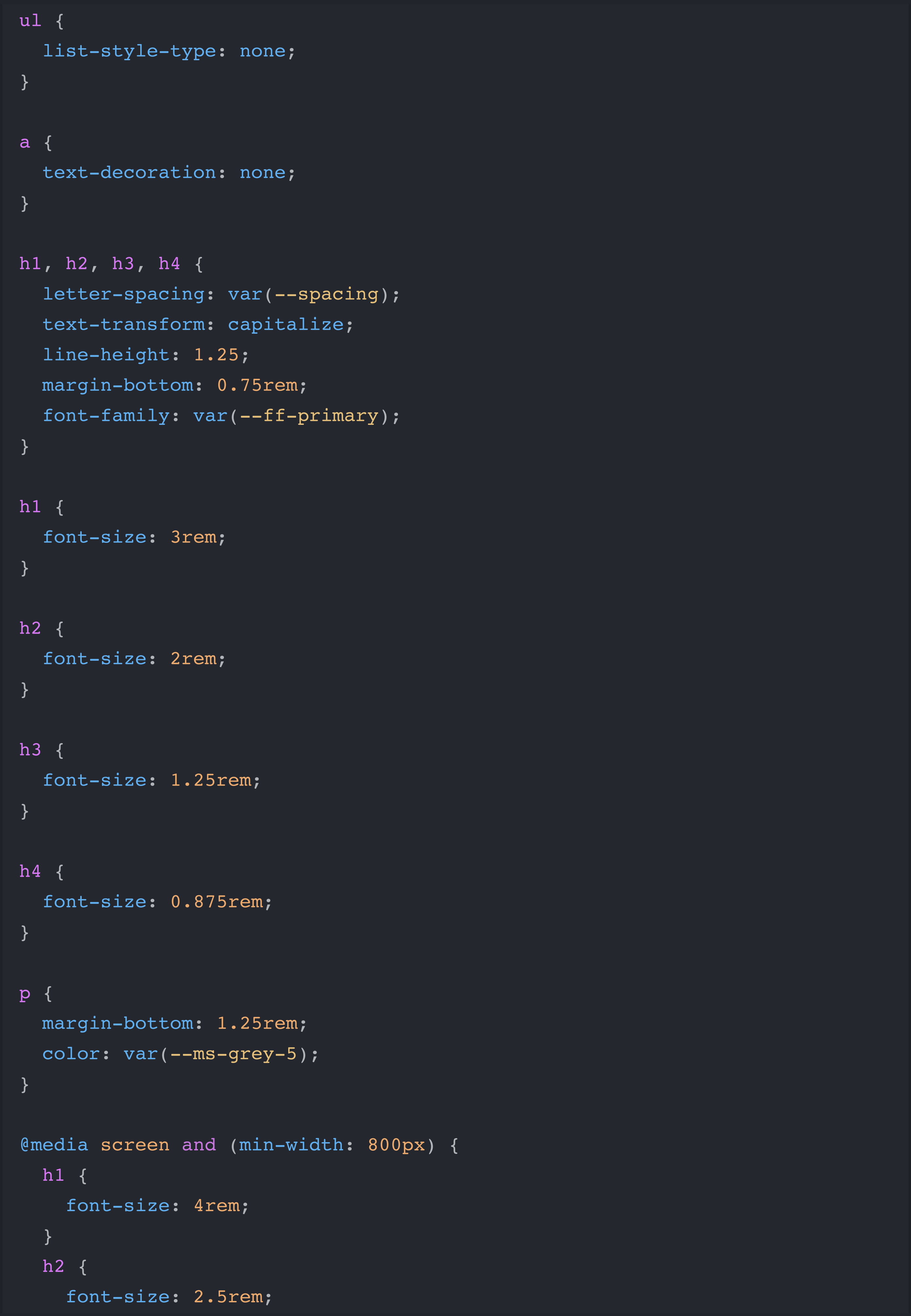
计数器练习

1、原生 JS

styles.css

```
/*
===============
字体
===============
*/

@import url("https://fonts.googleapis.com/css?family=Open+Sans|Roboto:400,700&display=swap");
/*
===============
全局变量
===============
*/

:root {
  /* primary color*/
  --ms-primary-1: hsl(205, 86%, 17%);
  --ms-primary-2: hsl(205, 77%, 27%);
  --ms-primary-3: hsl(205, 72%, 37%);
  --ms-primary-4: hsl(205, 63%, 48%);
  --ms-primary-5: hsl(205, 78%, 60%);
  --ms-primary-6: hsl(205, 89%, 70%);
  --ms-primary-7: hsl(205, 90%, 76%);
  --ms-primary-8: hsl(205, 86%, 81%);
  --ms-primary-9: hsl(205, 90%, 88%);
  --ms-primary-10: hsl(205, 100%, 96%);
  /* grey*/
  --ms-grey-1: hsl(209, 61%, 16%);
  --ms-grey-2: hsl(211, 39%, 23%);
  --ms-grey-3: hsl(209, 34%, 30%);
  --ms-grey-4: hsl(209, 28%, 39%);
  --ms-grey-5: hsl(210, 22%, 49%);
  --ms-grey-6: hsl(209, 23%, 60%);
  --ms-grey-7: hsl(211, 27%, 70%);
  --ms-grey-8: hsl(210, 31%, 80%);
  --ms-grey-9: hsl(212, 33%, 89%);
  --ms-grey-10: hsl(210, 36%, 96%);
  --ms-white: #fff;
  --ms-red-dark: hsl(360, 67%, 44%);
  --ms-red-light: hsl(360, 71%, 66%);
  --ms-green-dark: hsl(125, 67%, 44%);
  --ms-green-light: hsl(125, 71%, 66%);
  --ms-black: #222;
  --ff-primary: "Roboto", sans-serif;
  --ff-secondary: "Open Sans", sans-serif;
  --transition: all 0.3s linear;
  --spacing: 0.1rem;
  --radius: 0.25rem;
  --light-shadow: 0 5px 15px rgba(0, 0, 0, 0.1);
  --dark-shadow: 0 5px 15px rgba(0, 0, 0, 0.2);
  --max-width: 1170px;
  --fixed-width: 620px;
}

/*
===============
全局样式初始化
===============
*/

*, ::after, ::before {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
}

body {
  font-family: var(--ff-secondary);
  background: var(--ms-grey-10);
  color: var(--ms-grey-1);
  line-height: 1.5;
  font-size: 0.875rem;
}
```



```css
ul {
  list-style-type: none;
}

a {
  text-decoration: none;
}

h1, h2, h3, h4 {
  letter-spacing: var(--spacing);
  text-transform: capitalize;
  line-height: 1.25;
  margin-bottom: 0.75rem;
  font-family: var(--ff-primary);
}

h1 {
  font-size: 3rem;
}

h2 {
  font-size: 2rem;
}

h3 {
  font-size: 1.25rem;
}

h4 {
  font-size: 0.875rem;
}

p {
  margin-bottom: 1.25rem;
  color: var(--ms-grey-5);
}

@media screen and (min-width: 800px) {
  h1 {
    font-size: 4rem;
  }
  h2 {
    font-size: 2.5rem;
  }
  h3 {
    font-size: 1.75rem;
  }
  h4 {
    font-size: 1rem;
  }
  body {
    font-size: 1rem;
  }
  h1, h2, h3, h4 {
    line-height: 1;
  }
}

/* section */

.section {
  padding: 5rem 0;
}

.section-center {
  width: 90vw;
  margin: 0 auto;
  max-width: 1170px;
}

@media screen and (min-width: 992px) {
  .section-center {
    width: 95vw;
  }
}

main {
  min-height: 100vh;
  display: grid;
  place-items: center;
}

/*
===============
Counter
===============
*/

main {
  min-height: 100vh;
  display: grid;
  place-items: center;
}

.container {
  text-align: center;
}

#value {
  font-size: 6rem;
  font-weight: bold;
}

.btn {
  text-transform: uppercase;
  background: transparent;
  color: var(--ms-black);
  padding: 0.375rem 0.75rem;
  letter-spacing: var(--spacing);
  display: inline-block;
  transition: var(--transition);
  font-size: 0.875rem;
  border: 2px solid var(--ms-black);
  cursor: pointer;
  box-shadow: 0 1px 3px rgba(0, 0, 0, 0.2);
  border-radius: var(--radius);
  margin: 0.5rem;
}

.btn:hover {
  color: var(--ms-white);
  background: var(--ms-black);
}
```

index.html

```
<!DOCTYPE html>
```

```html
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>计数器</title>
    <link rel="stylesheet" href="styles.css" />
  </head>

  <body>
    <main id="app">
      <div class="container">
        <h1>计数器</h1>
        <span id="value">0</span>
        <div class="button-container">
          <button class="btn decrease">减</button>
          <button class="btn reset">重置</button>
          <button class="btn increase">加</button>
        </div>
      </div>
    </main>

    <script src="app.js"></script>
  </body>
</html>
```

app.js

```javascript
let count = 0;

const value = document.querySelector("#value");
const btns = document.querySelectorAll(".btn");

btns.forEach(function (btn) {
  btn.addEventListener("click", function (e) {
    const styles = e.currentTarget.classList;
    if (styles.contains("decrease")) {
      count -= 5;
    } else if (styles.contains("increase")) {
      count += 5;
    } else {
      count = 0;
    }

    if (count < 0) {
      value.style.color = "red";
    } else if (count > 10) {
      value.style.color = "green";
    } else {
      value.style.color = "#222";
    }

    value.textContent = count;
  });
});
```

2、 Vue 3 Options API

index.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>计数器</title>
    <link rel="stylesheet" href="styles.css" />
  </head>

  <body>
    <main id="app">
      <div class="container">
        <h1>计数器</h1>
        <span id="value" :style="{ color: resColor}">{{count}}</span>
        <div class="button-container">
          <button class="btn decrease" @click="decrement">减</button>
          <button class="btn reset" @click="reset">重置</button>
          <button class="btn increase" @click="increment">加</button>
        </div>
      </div>
    </main>

    <script src="https://unpkg.com/vue@3/dist/vue.global.js"></script>
    <script src="vue3-options-api.js"></script>
  </body>
</html>
```

vue3-options-api.js

```
const app = Vue.createApp({
  data() {
    return {
      count: 0,
      resColor: "grey",
    };
  },
  methods: {
    increment() {
      this.count += 5;
      if (this.count > 10) {
        this.resColor = "green";
      }
    },
    decrement() {
      this.count -= 5;
      if (this.count < 0) {
        this.resColor = "red";
      }
    },
    reset() {
      this.count = 0;
    },
  },
}).mount("#app");
```

3、 Vue 3 Compotion API

index.html 同2

换成引入 vue3-composition-api.js

```
<script src="vue3-compotion-api.js"></script>
```

vue3-composition-api.js

```
const { ref, createApp } = Vue;
createApp({
  setup() {
    const count = ref(0);
    const resColor = ref("grey");

    const increment = () => {
      count.value += 5;
      if (count.value > 10) {
        resColor.value = "green";
      }
    };

    const decrement = () => {
      count.value -= 5;
      if (count.value < 0) {
        resColor.value = "red";
      }
    };

    const reset = () => {
      count.value = 0;
    };

    return {
      count,
      resColor,
      increment,
      decrement,
      reset,
    };
  },
}).mount("#app");
```