گزارش تمرین اول درس برنامه‌سازی وب

مدرس: استاد پورسلطانی

امیرحسین محمدپور

شماره دانشجویی: 402170024

ابتدا یک فایل HTML ساختم که ابتدا به توضیح بخش‌های مختلف آن میپردازم.

```html
<!DOCTYPE html>
<html lang="fa">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Formula Calculator</title>
    <link rel="stylesheet" href="styles.css">
    <script src="script.js" defer></script>
</head>
<body>
    <div class="main-content">
        <div class="container">
            <h1>Formula Calculator</h1>
            <input type="number" id="First" placeholder="First Number">
            <input type="number" id="Second" placeholder="Second Number">
            <input type="number" id="Third" placeholder="Third Number">
            <formula evaluator="First*Second+Third"></formula>
            <formula evaluator="(First-Third)*Second"></formula>
            <formula evaluator="First+Second*Third"></formula>
        </div>
    </div>

    <footer class="student-footer">
        <div class="footer-content">
            <span class="student-name">Amirhossein Mohammadpour</span>
            <span class="student-id">402170024</span>
        </div>
    </footer>
</body>
</html>
```

در تگ **head** خصوصیاتی که صفحه ما در پشت پرده باید داشته باشد را تعیین میکنیم. مثل اینکه مجموعه کاراکترهای از چه استانداردی استفاده میکند و یا اینکه عنوان صفحه در بالای مرورگر باید چه باشد. همچنین در همین قسمت فایل **css** و **javascript** را به فایل **html** وصل میکنیم.

در قسمت **body** محتوای اصلی صفحه را تعیین میکنیم. یعنی قسمت‌هایی که قرار است داده را وارد کنیم (تگ **input**) و نتایج فرمول‌ها را مشاهده کنیم (تگ **formula**). و در نهایت فوتر صفحه را تعیین کرده‌ایم که در آن مشخصات نویسنده را نوشته‌ام.

در تگ‌های **input** مطابق خواسته سوال، داده‌ها را وارد میکنیم و سپس در تگ‌های **formula** مطابق با فرمولی که بر اساس **id** ورودی‌ها مشخص کرده‌ایم، نتایج را مشاهده میکنیم.

حال سراغ عملکرد فایل **javascript** میرویم:

```
    constructor() {
        this.inputs = {};
        this.formulaElements = document.querySelectorAll('formula');
        this.initInputs();
        this.initFormulas();
        this.setupEventListeners();
    }

    initInputs() {
        document.querySelectorAll('input[type="number"]').forEach(input => {
            this.inputs[input.id] = 0;
            input.addEventListener('input', () => this.updateInput(input));
        });
    }
```

در تابع **constructor** متغیرهای موردنیاز و توابعی که برای محاسبه فرمول‌ها موردنیاز است را تعریف میکنیم و یا صدا میزنیم. سپس در تابع **initInputs** به ازای هر ورودی مقدار اولیه آن را به 0 مقداردهی میکنیم و بر روی آن یک **eventListener** صدا میکنیم که در زمان تغییر ورودی‌ها فرمول‌ها با استفاده از تابع **updateInput** دوباره محاسبه شوند.

```
    updateInput(input) {
        this.inputs[input.id] = parseFloat(input.value) || 0;
        this.updateAllFormulas();
    }
```

در تابع **updateInput** مقدار متناظر با **id** هر ورودی را با مقدار متناظر ورودی در صورت عددبودن ست میکنیم و در غیر این صورت به صفر مقداردهی میکنیم و سپس به سراغ محاسبه کردن همه فرمول‌ها میرویم.

```
    initFormulas() {
        this.formulaElements.forEach(formula => {
            formula.setAttribute('readonly', 'true');
            this.evaluateFormula(formula);
        });
    }
```

در تابع **iniFormulas** هر فرمول را **readonly** میکنیم و سپس با استفاده از تابع **evaluateFormula** به سراغ محاسبه آن میرویم.

```
    evaluateFormula(formula) {
        const expression = formula.getAttribute('evaluator');
        try {
            const result = this.calculateExpression(expression);
            formula.textContent = result;
        } catch (e) {
            formula.textContent = 'Invalid Formula';
        }
    }
```

در این تابع با استفاده از ویژگی **evaluator** که در آن فرمول متناظر با هر تگ **formula** را نوشته‌ایم را ذخیره میکنیم و سپس به سراغ محاسبه فرمول با استفاده از تابع **calculateExpression** میرویم.

```
39
40
41      calculateExpression(expression) {
42          const vars = Object.keys(this.inputs);
43          const values = vars.map(varName => this.inputs[varName]);
44          const func = new Function(...vars, `return ${expression};`);
45          return func(...values);
46      }
```

در نهایت در این بخش ابتدا **key** متناظر با هر ورودی را میگیریم و سپس مقادیر متناظر را نیز ذخیره میکنیم و سپس شی **func** را ایجاد میکنیم و فرمول خواسته شده را نیز در آن ذخیره میکنیم و در نهایت آن را با استفاده از جایگذاری مقادیر داده‌شده محاسبه میکنیم.

```
7
8       updateAllFormulas() {
9           this.formulaElements.forEach(formula => this.evaluateFormula(formula));
0       }
1   }
```

این تابع نیز برای این است که اگر یک ورودی تغییر کرد، هر فرمولی دوباره محاسبه شود.

در مورد فایل **css** نیز در ادامه توضیحاتی میدهیم:

همانطور که میدانیم **css** کلا برای استایل‌دهی به صفحات وب استفاده میشود و تمام استایل‌ها را در آن نوشتم. با استفاده از این فایل من نحوه قرارگیری نوشته‌ها در وسط صفحه (text-align: center) و یا قرار دادن عکس پس زمینه با استفاده از این قسمت:

```
6
7   body {
8     background-image: url('sample.png');
9     background-size: cover;
10    background-position: center;
11    background-repeat: no-repeat;
12    background-attachment: fixed;
13    margin: 0;
14    padding: 0;
15    min-height: 100vh;
16    display: flex;
17    flex-direction: column;
18    justify-content: space-between;
19  }
```

همچنین ستونی قرارگرفتن فیلدها در این قسمت:

```
27  .container {
28    display: flex;
29    flex-direction: column;
30    align-items: center;
31    width: 90%;
32    max-width: 500px;
33    margin: 20px auto;
34    background-color: rgba(85, 85, 85, 0.9);
35    padding: 20px;
36    border-radius: 10px;
37    box-sizing: border-box;
38  }
```

و همچنین ریسپانسیو بودن آن که در این قسمت آن را انجام داده‌ایم:

```
87  @media (max-width: 600px) {
88    .container {
89        width: 95%;
90        padding: 15px;
91    }
92
93    input, formula {
94        width: 90%;
95    }
96
97    .footer-content {
98        flex-direction: column;
99        gap: 10px;
100   }
101 }
```

همچنین با استفاده از این فایل من رنگ پس‌زمینه و یا رنگ نوشته و یا فاصله از مرز و یا فیلدهای اطراف را مشخص کرده‌ام:

```
formula {
  display: block;
  text-align: center;
  padding: 10px;
  width: 80%;
  margin: 10px 0;
  background-color: #e9f7ef;
  border: 2px solid #28a745;
  color: #155724;
  font-weight: bold;
  border-radius: 4px;
}
```