



برنامه‌سازی وب

تمرین اول

استاد پورسلطانی

رایا رضائی (۴۰۱۱۷۰۵۷۵)



توضیحات

این برنامه در سه کد `code.html` برای تعریف المان‌های صفحه، `styles.css` برای زیبا سازی و `script.js` برای محاسبات و اعمال تغییرات نوشته شده که در ادامه به توضیح هر یک می‌پردازیم. در اینجا هر یک المان‌های موجود در صفحه را با در نظر گرفتن اصول DOM تعریف می‌کنیم. نام و توضیحات تمرین را در بخش `head` و المان‌های مربوط به ورودی‌ها و محاسبات را در بخش `body` تعریف کرده‌ایم. همچنین پیام‌های خطای هر بخش قابل مشاهده است.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Calculation of Price, Volume, and Area</title>
    <link rel="stylesheet" href="styles.css" />
    <script src="script.js" defer></script>
  </head>
  <body>
    <!-- Header section -->
    <div class="header">
      <h2>First Exercise of Web Programming Course</h2>
      <p>By: Raya Rezaie</p>
    </div>
    <div class="container">
      <h1>Calculation Form</h1>
      <!-- Price Calculation Inputs -->
      <div class="input-section">
        <div class="input-group">
          <label for="fee">Unit price:</label>
          <input type="text" id="fee" placeholder="Enter the unit price" />
          <span id="feeError" class="error">Invalid input</span>
        </div>
        <div class="input-group">
          <label for="count">Count:</label>
          <input type="text" id="count" placeholder="Enter the count" />
          <span id="countError" class="error">Invalid input</span>
        </div>
        <div class="input-group">
          <label for="discount">Discount:</label>
          <input type="text" id="discount" placeholder="Enter the discount" />
          <span id="discountError" class="error">Invalid input</span>
        </div>
      </div>
      <!-- Cylinder Volume Inputs -->
      <div class="input-section">
        <div class="input-group">
```

```

        <label for="radius">Radius of Cylinder:</label>
        <input type="text" id="radius" placeholder="Enter the radius" />
        <span id="radiusError" class="error">Invalid input</span>
    </div>
    <div class="input-group">
        <label for="height">Height of Cylinder:</label>
        <input type="text" id="height" placeholder="Enter the height" />
        <span id="heightError" class="error">Invalid input</span>
    </div>
</div>
<!-- Rectangle Area Inputs -->
<div class="input-section">
    <div class="input-group">
        <label for="width">Width of Rectangle:</label>
        <input type="text" id="width" placeholder="Enter the width" />
        <span id="widthError" class="error">Invalid input</span>
    </div>
    <div class="input-group">
        <label for="length">Length of Rectangle:</label>
        <input type="text" id="length" placeholder="Enter the length" />
        <span id="lengthError" class="error">Invalid input</span>
    </div>
</div>
<!-- Results Section -->
<h3>Results:</h3>
<div class="result-group">
    <div id="formula1" class="result">Price: 0</div>
    <div id="cylinderVolume" class="result">Cylinder volume: 0</div>
    <div id="rectangleArea" class="result">Rectangle area: 0</div>
</div>
</div>
</body>

```

مشخصات ظاهری بخش‌های مختلف صفحه را در فایل styles.css تعریف می‌کنیم.

```

1  body {
2      font-family: 'Arial', sans-serif;
3      background-color: #f4f4f9;
4      color: #333;
5      margin: 0;
6      padding: 0;
7  }
8
9  .header {
10     background-color: #4CAF50;
11     color: white;
12     padding: 15px;
13     text-align: center;
14     margin-bottom: 30px;
15 }
16
17 .header h2 {
18     font-size: 1.8rem;
19     margin: 0;
20 }
21
22 .header p {
23     font-size: 1.2rem;
24     margin: 5px 0 0 0;
25 }
26
27
28 .container {
29     width: 60%;
30     margin: 0 auto;
31     background-color: #fff;
32     padding: 20px;
33     border-radius: 8px;
34     box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
35 }

```

```

37 h1 {
38   text-align: center;
39   color: #4CAF50;
40 }
41 .input-section {
42   margin-bottom: 20px;
43 }
44 .input-group {
45   margin-bottom: 15px;
46   display: flex;
47   flex-direction: column;
48 }
49 label {
50   font-size: 1rem;
51   color: #333;
52   margin-bottom: 5px;
53 }
54 input {
55   padding: 10px;
56   font-size: 1rem;
57   border: 1px solid #ccc;
58   border-radius: 4px;
59   outline: none;
60   margin-bottom: 5px;
61 }
62 input:focus {
63   border-color: #4CAF50;
64 }
65 .error {
66   font-size: 0.875rem;
67   color: red;
68   display: none;
69 }
70 .result-group {
71   margin-top: 20px;
72   padding-top: 10px;
73   border-top: 2px solid #f0f0f0;

```

```

    .result {
      font-size: 1.2rem;
      color: #333;
      margin: 10px 0;
    }

    /* Responsive Design */
    @media (max-width: 768px) {
      .container {
        width: 90%;
      }
      h1 {
        font-size: 1.5rem;
      }
      .input-group {
        margin-bottom: 12px;
      }
      label {
        font-size: 0.9rem;
      }
      input {
        font-size: 1rem;
      }
      .result {
        font-size: 1rem;
      }
      .header h2 {
        font-size: 1.4rem;
      }
      .header p {
        font-size: 1rem;
      }
    }

```

بخش‌های مختلف پاسخگویی برنامه را نیز در کد JS تعریف می‌کنیم.

اولین کلاس، کلاس calculator برای تعریف فرمول‌ها و محاسبات آنها است.

```

class Calculator {
  constructor(){}

  calculateFormula(formula, variables) {
    switch (formula) {
      case 'price':
        return variables.count * variables.fee - variables.discount;
      case 'cylinderVolume':
        return Math.PI * Math.pow(variables.radius, 2) * variables.height;
      case 'rectangleArea':
        return variables.width * variables.length;
      default:
        return 0;
    }
  }
}

```

کلاس inputfield برای گرفتن ورودی‌ها و خطاهاست.

```

class InputField {
  constructor(id, validateCallback) {
    this.element = document.getElementById(id);
    this.errorElement = document.getElementById(id + 'Error');
    this.element.addEventListener('input', validateCallback);
  }

  getValue() {
    let value = this.element.value.trim();

    if (!/^(?!(\d+(\.\d+)?)|\.\d+)$/.test(value)) {
      return null;
    }
    return parseFloat(value);
  }

  setError(message) {
    this.errorElement.textContent = message;
    this.errorElement.style.display = message ? 'inline' : 'none';
  }

  validate(validationFunc) {
    validationFunc(this);
  }
}

```

در نهایت کلاس application برای تحلیل ورودی‌ها و نمایش خطا در صورت لزوم و یا فراخوانی توابع برای محاسبات و نمایش نتایج است.

```
47 class Application {
48   constructor() {
49     this.calculator = new Calculator();
50     this.allPriceInputsTouched = false;
51
52     // Initialize inputs with default values
53     this.feeInput = new InputField('fee', () => this.validateFee(), 0);
54     this.countInput = new InputField('count', () => this.validateCount(), 0);
55     this.discountInput = new InputField('discount', () => this.validateDiscount(), 0);
56     this.radiusInput = new InputField('radius', () => this.validateRadius());
57     this.heightInput = new InputField('height', () => this.validateHeight());
58     this.widthInput = new InputField('width', () => this.validateWidth());
59     this.lengthInput = new InputField('length', () => this.validateLength());
60
61     // Set initial values
62     this.updatePrice();
63     this.updateVolume();
64     this.updateArea();
65   }
66
67   checkAllPriceInputsTouched() {
68     this.allPriceInputsTouched = this.feeInput.touched &&
69     this.countInput.touched &&
70     this.discountInput.touched;
71   }
}
```

برای هر یک از بخش‌ها یک تابع validate تعریف شده که براساس شروط مربوط به هریک از ورودی‌ها آنها را بررسی می‌کند و سپس در صورتی که مشکلی داشته باشند خطای لازم را ارسال کرده و در صورت عدم مشکل، تابع update مربوط به آن فرمول فراخوانی می‌شود.

توابع بررسی ورودی‌ها

```
validateFee() {
  const value = this.feeInput.getValue();
  if (value === null) {
    this.feeInput.setError('Invalid number');
  } else if (value <= 0) {
    this.feeInput.setError('Must be positive');
  } else {
    this.feeInput.setError('');
  }
  this.updatePrice();
}

validateCount() {
  const value = this.countInput.getValue();
  if (value === null) {
    this.countInput.setError('Invalid number');
  } else if (value <= 0) {
    this.countInput.setError('Must be positive');
  } else {
    this.countInput.setError('');
  }
  this.updatePrice();
}

validateDiscount() {
  const value = this.discountInput.getValue();
  const total = this.feeInput.getValue() * this.countInput.getValue();

  if (value === null) {
    this.discountInput.setError('Invalid number');
  } else if (value < 0) {
    this.discountInput.setError('Cannot be negative');
  } else if (value > total) {
    this.discountInput.setError('Exceeds total price');
  } else {
    this.discountInput.setError('');
  }
}
```

```

validateRadius() {
  const value = this.radiusInput.getValue();
  if (value === null) {
    this.radiusInput.setError('Invalid number');
  } else if (value <= 0) {
    this.radiusInput.setError('Must be positive');
  } else {
    this.radiusInput.setError('');
  }
  this.updateVolume();
}

validateHeight() {
  const value = this.heightInput.getValue();
  if (value === null) {
    this.heightInput.setError('Invalid number');
  } else if (value <= 0) {
    this.heightInput.setError('Must be positive');
  } else {
    this.heightInput.setError('');
  }
  this.updateVolume();
}

```

```

validateWidth() {
  const value = this.widthInput.getValue();
  if (value === null) {
    this.widthInput.setError('Invalid number');
  } else if (value <= 0) {
    this.widthInput.setError('Must be positive');
  } else {
    this.widthInput.setError('');
  }
  this.updateArea();
}

validateLength() {
  const value = this.lengthInput.getValue();
  if (value === null) {
    this.lengthInput.setError('Invalid number');
  } else if (value <= 0) {
    this.lengthInput.setError('Must be positive');
  } else {
    this.lengthInput.setError('');
  }
  this.updateArea();
}

```

در این توابع با استفاده از

```
updateVolume() {
  const allValid = this.radiusInput.isValid() && this.heightInput.isValid();
  const anyTouched = this.radiusInput.touched || this.heightInput.touched;

  if (allValid) {
    const radius = this.radiusInput.getValue();
    const height = this.heightInput.getValue();
    const volume = this.calculator.calculateFormula('cylinderVolume', { radius, height });
    document.getElementById('cylinderVolume').textContent = `Cylinder volume: ${volume.toFixed(2)}`;
  } else {
    document.getElementById('cylinderVolume').textContent = anyTouched
      ? 'Invalid Formula'
      : 'cylinder volume: 0.00';
  }
}

updateArea() {
  const allValid = this.widthInput.isValid() && this.lengthInput.isValid();
  const anyTouched = this.widthInput.touched || this.lengthInput.touched;

  if (allValid) {
    const width = this.widthInput.getValue();
    const length = this.lengthInput.getValue();
    const area = this.calculator.calculateFormula('rectangleArea', { width, length });
    document.getElementById('rectangleArea').textContent = `Rectangle area: ${area.toFixed(2)}`;
  } else {
    document.getElementById('rectangleArea').textContent = anyTouched
      ? 'Invalid Formula'
      : 'Rectangle area: 0.00';
  }
}
```

بررسی عملکرد صفحه
دو نمونه از محاسبه‌ی خروجی‌ها

First Exercise of Web Programming Course

By: Raya Rezaie

Calculation Form

Unit price:

Count:

Discount:

Radius of Cylinder:

Height of Cylinder:

Width of Rectangle:

Length of Rectangle:

Results:

Price: 17.00

Cylinder volume: 56.55

Rectangle area: 4.00

Calculation Form

Unit price:

2

Count:

31

Discount:

1

Radius of Cylinder:

1

Height of Cylinder:

3.2

Width of Rectangle:

4.33

Length of Rectangle:

4

Results:

Price: 61.00

Cylinder volume: 10.05

Rectangle area: 17.32

خطاها

Calculation Form

Unit price:

2

Count:

3.2

Invalid number

Discount:

1

Radius of Cylinder:

1

Invalid number

Height of Cylinder:

-2

Must be positive

Width of Rectangle:

0

Must be positive

Length of Rectangle:

4

Results:

Invalid Formula

Invalid Formula

Invalid Formula

