

## تمرین دستگرمی اول

سید مبین رضوی 401110267

روند کلی کار به این صورت است که ابتدا به عنوان یک مثال پایه ای ، وبسایت موردنظر را برای فقط یک فرمول که از قبل داده شده است ( فرمول  $Fee \times count - discount$  ) پیاده سازی کرده ام تا مواردی مانند layout و style کلی صفحه و همچنین عملکرد کلی وبسایت در اینجا مشخص شود. این مورد در پوشه ی Basic Example قرار دارد که شامل فایل های basic.html ، basic.js و style.css می باشد. فایل style.css برطبق یک تمپلیت خاص طراحی شده است و برای تنظیم صحیح برخی پارامتر های آن ، از هوش مصنوعی کمک گرفته شده است. در ادامه نیز فایل html اصلی که شامل تمام موارد موردنظر در تمرین می باشد در پوشه ی Main HW1 قرار دارد و شامل فایل های warmup.html ، warmup.js و style.css است. نکته ی اصلی که در اینجا پیاده سازی شده است evaluate شدن المان formula برای هر فرمول دلخواهی است. برای روشن تر شدن موضوع ، توجه کنید که در basic.html و basic.js تنها یک فرمول از پیش مشخص شده قرار دارد و برای محاسبه شدن اش نیز ، در فایل basic.js به طور مستقیم از آن فرمول استفاده میکنم. اما در warmup.js ، طبق خواسته ی تمرین ، المان formula به نحوی پیاده سازی شده است که هر فرمول دلخواهی را برحسب id های المان های دیگر می توان در attribute متناظر evaluator نوشت و warmup.js به درستی آن فرمول را محاسبه می کند. به طور خلاصه ، خواسته ی تمرین در پوشه Main HW1 پیاده سازی شده است و پوشه Basic Example صرفا برای مشخص شدن پایه ی کار است.

### • Basic Example

کد های basic.html و style.css نما و ساختار کلی صفحه را مشخص کرده اند. در اینجا از کلاس های outer-box و inner-box به صورت nested برای قرار دادن ورودی ها و نتیجه ی فرمول استفاده شده است و در آنها المان های input و formula قرار دارند. اصل کار در فایل basic.js قرار دارد که ورودی ها را از المان های مخصوص شان میگیرد و طبق فرمول  $Fee \times count - discount$  نتیجه را محاسبه می کند. در زیر میتوانید فایل basic.js را مشاهده کنید.

```

Basic Example > JS basic.js > ...
1 document.addEventListener("DOMContentLoaded", () => {
2   const fee = document.getElementById("fee");
3   const count = document.getElementById("count");
4   const discount = document.getElementById("discount");
5   const result = document.getElementById("result");
6
7   const valid = (number) => {
8     return /^[0-9]*\.[0-9]+$/.test(number);
9   }
10
11  const evaluate = () => {
12    const feeValue = parseFloat(fee.value);
13    const countValue = parseFloat(count.value);
14    const discountValue = parseFloat(discount.value);
15
16    if(!valid(fee.value) || !valid(count.value) || !valid(discount.value) || feeValue < 0 || countValue < 0 || discountValue < 0){
17      result.textContent = 'Invalid Formula';
18      return;
19    }
20
21    const resultValue = feeValue * countValue - discountValue;
22    result.textContent = resultValue.toFixed(2);
23  }
24
25  fee.addEventListener("input", evaluate);
26  count.addEventListener("input", evaluate);
27  discount.addEventListener("input", evaluate);
28 });

```

در اینجا به کمک id المان های ورودی ، input ها مشخص می شوند. از تابع valid برای چک کردن معتبر بودن ورودی ها استفاده می شود و برای معتبر بودن ، باید مطابق regex داده شده که متناظر اعداد اعشاری یا صحیح مثبت می باشد ، باشند. حال value ی المان های ورودی تبدیل به float می شوند و اگر کمتر از 0 باشند و یا ورودی ها معتبر نباشند عبارت 'Invalid Formula' نشان داده خواهد شد. سپس مقدار نتیجه مطابق فرمولی که مشاهده می کنید محاسبه می شود. تمام این کار ها در تابع evaluate انجام می شوند. برای تمام المان های ورودی ، یک eventListener گذاشته شده است که با تغییر آنها ، تابع evaluate را اجرا می کند. با اینکار ، هر کدام از ورودی ها که تغییر بکنند مقدار Result آپدیت می شود. همانطور که گفته شد ، در اینجا تنها یک فرمول مشخص اجرا میشود.

توجه کنید که فایل style.css در هر دو پوشه یکسان می باشد و در انتهای این فایل نیز responsive بودن صفحه برحسب سایز اسکرین تضمین شده است. نمای صفحه را می توانید در زیر مشاهده کنید

## Welcome to My Simple Web Page

[Home](#) [About](#) [Contact](#)

### Home

This is a webpage for the first warm-up exercise.

#### Simple Formula Evaluator

Fee:

Count:

Discount:

Result: Invalid Formula

### About

Sharif University of Technology, Web Programming course, Spring 2025

### Contact

Mobin Razavi -- 401110267

© 2025 Simple Web Page. All rights reserved.

## Main HW1 •

فایل `warmup.html` کاملاً مشابه `html` قبلی است با این تفاوت که در اینجا سه تا `inner-box` با سه فرمول مختلف و دلخواه قرار داده شده است. تفاوت اصلی در فایل `warmup.js` است که عبارت نوشته شده در بخش `evaluator` المان `formula` را میخواند و فرمول متناظر آن را `extract` می کند و سپس این فرمول را روی ورودی ها اجرا میکند. در زیر فایل `warmup.js` را مشاهده می کنید. قابل ذکر است که برای پیاده سازی منطق این بخش نیز از هوش مصنوعی کمک گرفته شده است.

```

Main HW1 > JS Warmup.js > ...
1 document.addEventListener("DOMContentLoaded", () => {
2   const formulaElements = document.querySelectorAll("formula");
3
4   const valid = (number) => {
5     return /^[0-9]*\.[0-9]+$/.test(number);
6   }
7
8   formulaElements.forEach((formulaElement) => {
9     const evaluator = formulaElement.getAttribute("evaluator");
10    // const result = document.getElementById("result");
11    const result = formulaElement.querySelector("span");
12
13    const inputIds = evaluator.match(/\\b\\w+\\b/g);
14
15    const evaluate = () => {
16      let formula = evaluator;
17      var isValid = true;
18      inputIds.forEach(id => {
19        const inputElement = document.getElementById(id);
20        const elementValue = parseFloat(inputElement.value);
21        if(!valid(inputElement.value) || elementValue < 0){
22          result.textContent = `Invalid Formula`;
23          isValid = false;
24          return;
25        }
26        formula = formula.replace(new RegExp(`\\b${id}\\b`, "g"), elementValue);
27      });
28
29      if(!isValid){return;}
30      const resultValue = eval(formula);
31      result.textContent = resultValue.toFixed(2);
32    };
33
34    inputIds.forEach(id => {
35      const inputElement = document.getElementById(id);
36      if (inputElement) {
37        inputElement.addEventListener("input", evaluate);
38      }
39    });
40  });

```

برای هر المان formula ، آرایه ای از id ورودی ها ، در خط 13 محاسبه می شود. در تابع evaluate بالا ، تمام این id ها در formula ی داده شده ، یکی یکی با مقدارشان جایگزین می شوند ( خط 26 ) و توجه کنید که اگر مقدار یکی از آنها valid نباشد، بولین isValid برابر false قرار داده می شود تا در نهایت عبارت 'Invalid Formula' نوشته بشود. توجه کنید که متغیر isValid به صورت var تعریف شده است زیرا باید در اسکوپ تابع evaluate قابل استفاده باشد. در نهایت نیز نتیجه در خط 30 محاسبه شده و سپس نشان داده می شود . توجه کنید که برای تمام ورودی هایی که در آرایه ی inputIds قرار دارند یک eventlistener گذاشته شده تا با تغییر آنها تابع evaluate اجرا بشود.

در زیر می توانید نتیجه ی نهایی را ببینید :

## Welcome to My Simple Web Page

[Home](#) [About](#) [Contact](#)

### Home

This is a webpage for the first warm-up exercise.

#### Simple Formula Evaluator

Net Worth:

Profit:

Loss:

Result: 0

#### Purchase Cost Evaluator

Fee:

Count:

Discount:

Result: 0

#### Sales Tax Calculator

Price:

Quantity:

Tax Rate:

Result: 0

### About

Sharif University of Technology. Web Programming course. Spring 2025

### Contact

Mobin Razavi -- 401110267

© 2025 Simple Web Page. All rights reserved.

در اینجا سه فرمول متفاوت در warmup.html قرار داده شده است که متناظر سه جعبه ی بالا می باشند. از چپ به راست فرمول ها بدین صورت اند :

$$NetWorth + Profit - Loss$$

$$Fee * Count - Discount$$

$$Price * Quantity * (1 + TaxRate)$$