

# 3

## Part 3. 수익형 웹 서비스 구현하기(2) - 대출 이자 계산기 만들기

### 3-1. 대출 이자 계산기 미리보기

파트2에서 만든 계산기를 응용하여 이번엔 대출 이자 계산기를 만들어보도록 하겠습니다.

## 대출 이자 계산기

대출금	<input type="text" value="100000000"/>	원	<input type="button" value="+1백만"/>	<input type="button" value="+1천만"/>	<input type="button" value="+1억"/>
대출금리	<input type="text" value="연 3"/>	%	<input type="button" value="3%"/>	<input type="button" value="4%"/>	<input type="button" value="5%"/>
대출기간	<input type="text" value="12"/>	개월	<input type="button" value="1년"/>	<input type="button" value="3년"/>	<input type="button" value="5년"/>

계산하기

고객님의 12개월 동안 월 상환액은 8,469,370원 입니다.

### 3-2. 대출 이자 계산기의 프로그래밍적 원리 이해하기

#### 원리금균등상환 방식

우리는 원리금균등상환 방식의 대출 이자 계산기를 만들겁니다.

💡 대출 원금과 이자를 더한 금액을 만기일까지 균등하게 상환하는 대출 상환 방식입니다. 가장 일반적인 상환 방식으로 매달 내야 하는 납부금이 동일하기에 자금 관리 면에서 유리한 방식이죠!

예시를 하나 들어봅시다. 우리가 1000만원을 연 10%의 금리로 12개월 원리금균등상환으로 대출받았다고 해 봅시다. 단순히 1000만원의 10%이니 이자가 100만원이다!라고 생각할수도 있지만 실제로는 그렇지 않습니다. 매달 지날 수록 원금이 줄어들기 때문에 이자도 조금씩 줄어들기 때문이죠. 따라서 계산하는 방식이 까다롭습니다. 계산 방식이 까다롭기 때문에 사람들이 인터넷에 많이 검색할 것이고, 따라서 이런 계산기를 만들면 사람들이 많이 사용할 가능성이 높겠죠?

$$\text{월상환액} = \frac{\text{원금} \times \text{월금리} \times (1 + \text{월금리})^{\text{기간}}}{(1 + \text{월금리})^{\text{기간}} - 1}$$

좀 복잡하죠? 하지만 하나하나 그대로 따라하면 그렇게 어려울 것은 없습니다.

위의 예로 다시 돌아가서 정리하면:

- 원금 : 10,000,000만원
- 연금리 : 10% (월금리 0.833%)
  - 계산할때는 퍼센티지를 소수점으로 바꾼다. (100% → 1.0)
    - 10% → 0.1
    - 0.833% → 0.00833
- 상환기간 : 12개월

따라서 수식에 따라 넣어보면:

$$\text{월상환액} = \frac{1000\text{만원} \times 0.00833 \times (1 + 0.00833)^{12}}{(1 + 0.00833)^{12} - 1}$$

계산기로 계산해보면 월 상환액은 약 879,158원이라는 것을 알 수 있죠. (계산 순서에 주의하세요!)

## 사용자 입력

우리는 사용자에게서 3개를 입력받을 예정입니다. 이것들을 입력받기위한 텍스트박스 3개가 필요하겠죠? 그리고 계산하기 버튼을 눌러 계산할 수 있습니다.

1. 대출금 (원)
2. 대출금리 (연 %)
3. 대출기간 (개월)
4. 계산하기 (버튼)

## 3-3. HTML로 계산기 틀 만들기

### 보일러플레이트

새로운 프로젝트를 시작할 때는 항상 보일러플레이트 코드부터 시작합니다.

```
<!DOCTYPE html> <html> <head> <meta charset="UTF-8"> <meta name="viewport"
content="width=device-width, initial-scale=1" /> <title>Hello, world!</title> <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css"
rel="stylesheet"> <link rel="stylesheet" href="style.css"> </head> <body> <main
class="container"> <h1>Hello, world!</h1> </main> <script
src="https://code.jquery.com/jquery-3.6.1.min.js"></script> <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/js/bootstrap.bundle.min.js">
</script> <script src="script.js"></script> </body> </html>
```

index.html

```
$(document).ready(function() { });
```

script.js

```
/* 비어있음 */
```

style.css

## Bootstrap에서 표 사용하기

이번 시간에는 표(Table)를 사용하여 예쁘게 계산기를 만들어보도록 하겠습니다. 표를 이용하면 각이 딱딱 맞는 디자인을 만들 수 있죠!

Bootstrap 표 문서: <https://getbootstrap.com/docs/5.2/content/tables/#overview>

main.container안에 아래와 같이 테이블 코드를 복사+붙여넣기 합니다. thead 부분은 필요없으므로 삭제합니다.

```
<table class="table"> <tbody> <tr> <th scope="row">1</th> <td>Mark</td> <td>Otto</td>  
<td>@mdo</td> </tr> <tr> <th scope="row">2</th> <td>Jacob</td> <td>Thornton</td>  
<td>@fat</td> </tr> <tr> <th scope="row">3</th> <td colspan="2">Larry the Bird</td>  
<td>@twitter</td> </tr> </tbody> </table>
```

# 대출 이자 계산기

1	Mark	Otto	@mdo
2	Jacob	Thornton	@fat
3	Larry the Bird		@twitter

## 요소 구현하기

먼저 텍스트박스로 이루어진 기본적인 요소들부터 구현해봅시다.

```
<table class="table"> <tbody> <tr> <th scope="row">대출금</th> <td> <input type="text" id="loan"> 원 </td> </tr> <tr> <th scope="row">대출금리</th> <td> 연 <input type="text" id="interest"> % </td> </tr> <tr> <th scope="row">대출기간</th> <td> <input type="text" id="period"> 개월 </td> </tr> </tbody> </table> <button type="button" id="calculate">계산하기</button> <div id="result"></div>
```

# 대출 이자 계산기

대출금

원

대출금리

연

%

대출기간

개월

계산하기

## 3-4. 원리금균등상환 계산기 만들기

### 사용자 입력 받아오기

먼저 텍스트박스에서 사용자가 입력한 값들을 받아오는 것부터 만들어보죠. 계산하기 버튼을 클릭하면 3개의 텍스트박스에 있는 값을 불러오죠. script.js 파일에 다음과 같이 작성합니다.

```
function calculateHandler() { var loan = $('#loan').val(); var interest =
$('#interest').val(); var period = $('#period').val(); loan = parseFloat(loan); interest
= parseFloat(interest); period = parseFloat(period); } $(document).ready(function() {
$('#calculate').click(calculateHandler); });
```

script.js

1. id가 calculate인 버튼을 클릭하면, calculateHandler() 함수안에 있는 코드가 실행됩니다.
2. calculateHandler() 함수는 id가 loan, interest, period 인 텍스트박스에서 값을 가져와 각 변수에 저장합니다.
3. 마지막으로 각각의 값을 문자에서 숫자로 바꿔 저장합니다.

## 계산 전 필요한 값 정리하기

위에서 월 상환액을 계산했던 식을 다시 한 번 살펴봅시다.

$$\text{월상환액} = \frac{\text{원금} \times \text{월금리} \times (1 + \text{월금리})^{\text{기간}}}{(1 + \text{월금리})^{\text{기간}} - 1}$$

위의 식에서 원금, 연금리, 기간은 이미 받아왔지만 월금리는 우리가 별도로 계산해야 합니다. 월금리를 계산하는 식은 간단하죠. 단순히 연금리에서 12(개월)를 나누면 끝입니다. 따라서 우리는 월금리를 구하는 코드를 만들어 따로 저장합니다.



```
var interestMonth = interest / 12;
```

interestMonth 변수를 만들어 연금리(interest) 에서 12를 나누어 저장합니다.

현재 interestMonth에는 퍼센티지 단위의 값이 저장되어 있습니다. 퍼센티지를 소수점으로 변환하기 위해 100으로 나누어줍니다.

```
var interestMonth = interest / 12 / 100;
```

## 월 상환액 계산하기

수식이 복잡하니 분자와 분모를 먼저 계산하여 나중에 나눠주는 방식으로 진행하겠습니다.

일단 분자를 구하는 식을 코드로 작성하면 아래와 같이 되겠죠?

💡 JavaScript에서 거듭제곱은 \*\* 기호로 표시합니다.

```
var div1 = loan * interestMonth * (1 + interestMonth) ** period;
```

분모를 구하는 식은 아래와 같습니다.

```
var div2 = (1 + interestMonth) ** period - 1;
```

분자에서 분모로 나누어 최종 결과를 완성합니다.

```
var result = div1 / div2;
```

## 월 상환액 결과 표시하기

id가 result인 요소에 우리가 계산한 월 상환액을 표시합니다.

```
$('#result').text(result);
```

## 대출 이자 계산기

대출금

원

대출금리

연

%

대출기간

개월

계산하기

879158.872300099

### 월 상환액 반올림하여 결과 표시하기

결과값에 소수점이 나오니 지저분합니다. 우리는 이것을 반올림하여 사용자가 보기 쉽게 표시하도록 하겠습니다. 이럴 때는 함수를 외워서 하는 것이 아니라 구글을 주로 활용합니다. 구글에 "javascript 반올림"이라고 검색하여 정보를 찾습니다.

```
result = Math.round(result);
```

# 대출 이자 계산기

대출금

원

대출금리

연 

%

대출기간

개월

계산하기

879159

## 사용자 친화적으로 만들기

계산한 결과가 숫자로만 나오니 약간 불친절한 느낌이 듭니다. “고객님의 월 상환액은 00000원입니다.” 등의 메시지를 함께 띄워보죠.

```
var resultText = `고객님의 ${period}개월 동안 월 상환액은 ${result}원 입니다.`;  
$('#result').text(resultText);
```

💡 JavaScript에서 백틱(Backtick, ```)을 사용하면 변수에 저장된 값을 원하는 문구와 함께 출력할 수 있습니다. 변수를 표시할 때는 백틱 안에 `${변수명}` 이렇게 작성해주세요.

## 대출 이자 계산기

대출금  원

대출금리 연  %

대출기간  개월

계산하기

고객님의 12개월 동안 월 상환액은 879159원 입니다.

전체 코드

```
<!DOCTYPE html> <html> <head> <meta charset="UTF-8"> <meta name="viewport"
content="width=device-width, initial-scale=1" /> <title>Hello, world!</title> <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css"
rel="stylesheet"> <link rel="stylesheet" href="style.css"> </head> <body> <main
class="container"> <h1>대출 이자 계산기</h1> <table class="table"> <tbody> <tr> <th
scope="row">대출금</th> <td> <input type="text" id="loan"> 원 </td> </tr> <tr> <th
scope="row">대출금리</th> <td> 연 <input type="text" id="interest"> % </td> </tr> <tr>
<th scope="row">대출기간</th> <td> <input type="text" id="period"> 개월 </td> </tr>
</tbody> </table> <button type="button" id="calculate">계산하기</button> <div
id="result"></div> </main> <script src="https://code.jquery.com/jquery-3.6.1.min.js">
</script> <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/js/bootstrap.bundle.min.js">
</script> <script src="script.js"></script> </body> </html>
```

index.html

```
function calculateHandler() { var loan = $('#loan').val(); var interest =
$('#interest').val(); var period = $('#period').val(); loan = parseFloat(loan); interest
= parseFloat(interest); period = parseFloat(period); var interestMonth = interest / 12 /
100; var div1 = loan * interestMonth * (1 + interestMonth) ** period; var div2 = (1 +
interestMonth) ** period - 1; var result = div1 / div2; result = Math.round(result); var
resultText = `고객님의 ${period}개월 동안 월 상환액은 ${result}원 입니다.`;
$('#result').text(resultText); } $(document).ready(function() {
$('#calculate').click(calculateHandler); });
```

script.js

### 3-5. 사용자 편의 기능 만들기

사용자들이 모든 숫자를 직접 입력하려면 힘들겠죠? 우리는 사용자들을 더 편하게해서 방문율을 높이기 위해 약간의 편의 기능들을 만들겁니다. 예를 들면 이런거죠.

금액을 입력하세요.

+1만 +5만 +10만 +100만 전액

1,000,000 원

1 2 3

4 5 6

7 8 9

00 0

확인

기업은행 인터넷뱅킹에서 송금을 하려고 금액을 입력했을 때 뜨는 팝업입니다. 금액을 직접 입력할 수도 있지만 +10만 +100만 버튼을 사용해서 더 간단하게 입력할수도 있습니다.

### 버튼 만들기

먼저 대출금액부터 손봐줄게요. td를 하나 더 만들어 열을 추가합니다. 그 안에 3개의 버튼을 추가할게요.

```
<td> <button type="button" id="loan-100">+백만</button> <button type="button" id="loan-1000">+천만</button> <button type="button" id="loan-10000">+일억</button> </td>
```

## 대출 이자 계산기

대출금	<input type="text"/>	원	<button>+백만</button>	<button>+천만</button>	<button>+일억</button>
대출금리	연 <input type="text"/>	%			
대출기간	<input type="text"/>	개월			
<button>계산하기</button>					

각 버튼의 id를 loan-100, loan-1000, loan-10000 으로 지정하였습니다. 먼저 loan-100을 클릭하면 100만원이 더해지도록 하죠.

### 100만원 더하기

script.js 파일에 loan-100 버튼을 클릭했을 때 loan100Handler() 함수가 실행되도록 만들어줍니다.



```
$('#loan-100').click(loan100Handler);
```

loan100Handler() 에서 해야하는 일은 다음과 같습니다.

1. 현재 입력되어 있는 대출금 값을 가져온다.
2. 현재 입력값에 100만원을 더한다.
3. 대출금 텍스트박스에 더한 값을 넣는다.

이것을 코드로 작성하면:

```
function loan100Handler() { var loan = $('#loan').val(); loan = parseFloat(loan); loan = loan + 1000000; $('#loan').val(loan); }
```

💡 val() 함수는 괄호 안쪽이 비어있을 때는 해당 id 요소의 값을 가져오고, 안쪽에 값 또는 변수를 넣었을 때는 해당 id 요소에 값을 채워넣습니다.

하지만 이대로는 잘 동작하지 않습니다. [+백만] 버튼을 누르면 NaN이라는 글자가 뜨죠. NaN은 Not a Number 의 약자로 "숫자가 아니다"라는 뜻입니다. 이 오류가 발생하는 이유는 현재 loan 텍스트박스에 아무 것도 입력되어 있지 않아서 그런건데요. 우리는 이 에러를 해결하기 위해 예외처리를 해주어야 합니다.

```
function loan100Handler() { var loan = $('#loan').val(); if (loan == "") { loan = 0; } loan = parseFloat(loan); loan = loan + 1000000; $('#loan').val(loan); }
```

loan 텍스트박스에서 값을 받아온 후 loan에 저장된 값이 빈 칸("")인지 확인하여 빈 칸이라면 0을 넣어주는 코드를 작성합니다. 이제 정상적으로 동작합니다.

## 다른 버튼 구현하기

마찬가지로 숫자만 변경하여 천만, 일억 버튼도 만들어줍니다.

```
function loan1000Handler() { var loan = $('#loan').val(); if (loan == "") { loan = 0; }  
loan = parseFloat(loan); loan = loan + 10000000; $('#loan').val(loan); }
```

```
function loan10000Handler() { var loan = $('#loan').val(); if (loan == "") { loan = 0; }  
loan = parseFloat(loan); loan = loan + 100000000; $('#loan').val(loan); }
```

## 전체 코드

```
<!DOCTYPE html> <html> <head> <meta charset="UTF-8"> <meta name="viewport"
content="width=device-width, initial-scale=1" /> <title>Hello, world!</title> <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css"
rel="stylesheet"> <link rel="stylesheet" href="style.css"> </head> <body> <main
class="container"> <h1>대출 이자 계산기</h1> <table class="table"> <tbody> <tr> <th
scope="row">대출금</th> <td> <input type="text" id="loan"> 원 </td> <td> <button
type="button" id="loan-100">+백만</button> <button type="button" id="loan-1000">+천만
</button> <button type="button" id="loan-10000">+일억</button> </td> </tr> <tr> <th
scope="row">대출금리</th> <td> 연 <input type="text" id="interest"> % </td> </tr> <tr>
<th scope="row">대출기간</th> <td> <input type="text" id="period"> 개월 </td> </tr>
</tbody> </table> <button type="button" id="calculate">계산하기</button> <div
id="result"></div> </main> <script src="https://code.jquery.com/jquery-3.6.1.min.js">
</script> <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/js/bootstrap.bundle.min.js">
</script> <script src="script.js"></script> </body> </html>
```

index.html

```
function calculateHandler() { var loan = $('#loan').val(); var interest =
$('#interest').val(); var period = $('#period').val(); loan = parseFloat(loan); interest
= parseFloat(interest); period = parseFloat(period); var interestMonth = interest / 12 /
100; var div1 = loan * interestMonth * (1 + interestMonth) ** period; var div2 = (1 +
interestMonth) ** period - 1; var result = div1 / div2; result = Math.round(result); var
resultText = `고객님의 ${period}개월 동안 월 상환액은 ${result}원 입니다.`;
$('#result').text(resultText); } function loan100Handler() { var loan = $('#loan').val();
if (loan == "") { loan = 0; } loan = parseFloat(loan); loan = loan + 1000000;
$('#loan').val(loan); } function loan1000Handler() { var loan = $('#loan').val(); if
(loan == "") { loan = 0; } loan = parseFloat(loan); loan = loan + 10000000;
$('#loan').val(loan); } function loan10000Handler() { var loan = $('#loan').val(); if
(loan == "") { loan = 0; } loan = parseFloat(loan); loan = loan + 100000000;
$('#loan').val(loan); } $(document).ready(function() {
$('#calculate').click(calculateHandler); $('#loan-100').click(loan100Handler); $('#loan-
1000').click(loan1000Handler); $('#loan-10000').click(loan10000Handler); });
```

script.js

## 대출금리와 대출기간 버튼 만들기

대출금리와 대출기간도 위와 비슷한 기능을 하는 버튼을 만들어주겠습니다.

```
<tr> <th scope="row">대출금리</th> <td> 연 <input type="text" id="interest"> % </td> <td>
<button type="button" id="interest-3">3%</button> <button type="button" id="interest-
4">4%</button> <button type="button" id="interest-5">5%</button> </td> </tr> <tr> <th
scope="row">대출기간</th> <td> <input type="text" id="period"> 개월 </td> <td> <button
type="button" id="period-1">1년</button> <button type="button" id="period-3">3년</button>
<button type="button" id="period-5">5년</button> </td> </tr>
```

이 버튼들은 더하기 버튼이 아니어서 현재 입력된 값에 더하는 형태가 아니라 새로운 값을 채워주는 형태로 만들어줄게요.

```
function interest3Handler() { $('#interest').val(3); } function interest4Handler() {
$('#interest').val(4); } function interest5Handler() { $('#interest').val(5); } function
period1Handler() { $('#period').val(12); } function period3Handler() {
$('#period').val(36); } function period5Handler() { $('#period').val(60); }
$(document).ready(function() { $('#calculate').click(calculateHandler); $('#loan-
100').click(loan100Handler); $('#loan-1000').click(loan1000Handler); $('#loan-
10000').click(loan10000Handler); $('#interest-3').click(interest3Handler); $('#interest-
4').click(interest4Handler); $('#interest-5').click(interest5Handler); $('#period-
1').click(period1Handler); $('#period-3').click(period3Handler); $('#period-
5').click(period5Handler); });
```

# 대출 이자 계산기

대출금	<input type="text" value="100000000"/>	원	<input type="button" value="+백만"/>	<input type="button" value="+천만"/>	<input type="button" value="+일억"/>
대출금리	연 <input type="text" value="3"/>	%	<input type="button" value="3%"/>	<input type="button" value="4%"/>	<input type="button" value="5%"/>
대출기간	<input type="text" value="60"/>	개월	<input type="button" value="1년"/>	<input type="button" value="3년"/>	<input type="button" value="5년"/>

고객님의 60개월 동안 월 상환액은 1796869원 입니다.

전체 코드

```
<!DOCTYPE html> <html> <head> <meta charset="UTF-8"> <meta name="viewport"
content="width=device-width, initial-scale=1" /> <title>Hello, world!</title> <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css"
rel="stylesheet"> <link rel="stylesheet" href="style.css"> </head> <body> <main
class="container"> <h1>대출 이자 계산기</h1> <table class="table"> <tbody> <tr> <th
scope="row">대출금</th> <td> <input type="text" id="loan"> 원 </td> <td> <button
type="button" id="loan-100">+백만</button> <button type="button" id="loan-1000">+천만
</button> <button type="button" id="loan-10000">+일억</button> </td> </tr> <tr> <th
scope="row">대출금리</th> <td> 연 <input type="text" id="interest"> % </td> <td> <button
type="button" id="interest-3">3%</button> <button type="button" id="interest-4">4%
</button> <button type="button" id="interest-5">5%</button> </td> </tr> <tr> <th
scope="row">대출기간</th> <td> <input type="text" id="period"> 개월 </td> <td> <button
type="button" id="period-1">1년</button> <button type="button" id="period-3">3년</button>
<button type="button" id="period-5">5년</button> </td> </tr> </tbody> </table> <button
type="button" id="calculate">계산하기</button> <div id="result"></div> </main> <script
src="https://code.jquery.com/jquery-3.6.1.min.js"></script> <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/js/bootstrap.bundle.min.js">
</script> <script src="script.js"></script> </body> </html>
```

index.html

```
function calculateHandler() { var loan = $('#loan').val(); var interest =
$('#interest').val(); var period = $('#period').val(); loan = parseFloat(loan); interest
= parseFloat(interest); period = parseFloat(period); var interestMonth = interest / 12 /
100; var div1 = loan * interestMonth * (1 + interestMonth) ** period; var div2 = (1 +
interestMonth) ** period - 1; var result = div1 / div2; result = Math.round(result); var
resultText = `고객님의 ${period}개월 동안 월 상환액은 ${result}원 입니다.`;
$('#result').text(resultText); } function loan100Handler() { var loan = $('#loan').val();
if (loan == "") { loan = 0; } loan = parseFloat(loan); loan = loan + 1000000;
$('#loan').val(loan); } function loan1000Handler() { var loan = $('#loan').val(); if
(loan == "") { loan = 0; } loan = parseFloat(loan); loan = loan + 10000000;
$('#loan').val(loan); } function loan10000Handler() { var loan = $('#loan').val(); if
(loan == "") { loan = 0; } loan = parseFloat(loan); loan = loan + 100000000;
$('#loan').val(loan); } function interest3Handler() { $('#interest').val(3); } function
interest4Handler() { $('#interest').val(4); } function interest5Handler() {
$('#interest').val(5); } function period1Handler() { $('#period').val(12); } function
period3Handler() { $('#period').val(36); } function period5Handler() {
$('#period').val(60); } $(document).ready(function() {
$('#calculate').click(calculateHandler); $('#loan-100').click(loan100Handler); $('#loan-
1000').click(loan1000Handler); $('#loan-10000').click(loan10000Handler); $('#interest-
3').click(interest3Handler); $('#interest-4').click(interest4Handler); $('#interest-
5').click(interest5Handler); $('#period-1').click(period1Handler); $('#period-
3').click(period3Handler); $('#period-5').click(period5Handler); });
```

script.js

### 3-6. Bootstrap을 활용한 계산기 디자인 꾸미기



## 텍스트박스 꾸미기

이번에도 Bootstrap을 사용하여 계산기를 예쁘게 꾸며봅시다. Bootstrap의 인풋 그룹 문서를 보겠습니다.

Bootstrap 인풋 그룹 문서: <https://getbootstrap.com/docs/5.2/forms/input-group/>

여러분들이 맘에 드는 디자인으로 복사 붙여넣기 해도 괜찮습니다. 저는 아래와 같은 코드를 사용하여 텍스트 박스를 꾸며보겠습니다. 꾸미고 난 후에는 동작이 잘되는지 꼭 확인합시다!

```
<div class="input-group mb-3"> <span class="input-group-text">${</span> <input type="text"
class="form-control" aria-label="Amount (to the nearest dollar)"> <span class="input-
group-text">.00</span> </div>
```

디자인 적용한 코드

```
<tr> <th scope="row">대출금</th> <td> <div class="input-group"> <input type="text"
class="form-control" id="loan"> <span class="input-group-text">원</span> </div> </td>
<td> <button type="button" id="loan-100">+백만</button> <button type="button" id="loan-
1000">+천만</button> <button type="button" id="loan-10000">+일억</button> </td> </tr>
```

## 대출 이자 계산기

대출금	<input type="text" value="30000000"/>	원	+백만	+천만	+일억
대출금리	<div>연</div> <input type="text" value="3"/> <div>%</div>		3%	4%	5%
대출기간	<input type="text" value="12"/>	개월	1년	3년	5년

계산하기

고객님의 12개월 동안 월 상환액은 2540811원 입니다.

### 편의 기능 버튼 꾸미기

버튼은 버튼 그룹으로 묶어주겠습니다.

버튼 그룹 문서: <https://getbootstrap.com/docs/5.2/components/button-group/>

```
<tr> <th scope="row">대출금</th> <td> <div class="input-group"> <input type="text"
class="form-control" id="loan"> <span class="input-group-text">원</span> </div> </td>
<td> <div class="btn-group" role="group"> <button type="button" class="btn btn-outline-
primary" id="loan-100">+1백만</button> <button type="button" class="btn btn-outline-
primary" id="loan-1000">+1천만</button> <button type="button" class="btn btn-outline-
primary" id="loan-10000">+1억</button> </div> </td> </tr>
```

## 대출 이자 계산기

대출금	<input type="text"/>	원	+1백만	+1천만	+1억
대출금리	연 <input type="text"/>	%	3%	4%	5%
대출기간	<input type="text"/>	개월	1년	3년	5년

계산하기

### 계산하기 버튼 꾸미기

버튼 문서를 참고하여 계산하기 버튼도 꾸며봅시다. 저는 계산하기 버튼에 `class="btn btn-primary"` 를 추가하여 파란색 버튼을 만들겠습니다.

버튼 문서: <https://getbootstrap.com/docs/5.2/components/buttons/>

## 대출 이자 계산기

대출금	<input type="text" value="1000000000"/>	원	<input type="button" value="+1백만"/>	<input type="button" value="+1천만"/>	<input type="button" value="+1억"/>
대출금리	<input type="text" value="3"/>	%	<input type="button" value="3%"/>	<input type="button" value="4%"/>	<input type="button" value="5%"/>
대출기간	<input type="text" value="12"/>	개월	<input type="button" value="1년"/>	<input type="button" value="3년"/>	<input type="button" value="5년"/>

계산하기

고객님의 12개월 동안 월 상환액은 8469370원 입니다.

### 결과 텍스트 꾸미기

Bootstrap의 타이포그래피 문서 중에 텍스트의 일부를 하이라이트할 수 있는 방법이 있습니다.

타이포그래피 문서: <https://getbootstrap.com/docs/5.2/content/typography/#inline-text-elements>

## Inline text elements #

Styling for common inline HTML5 elements.

You can use the mark tag to highlight text.

~~This line of text is meant to be treated as deleted text.~~

~~This line of text is meant to be treated as no longer accurate.~~

This line of text is meant to be treated as an addition to the document.

This line of text will render as underlined.

This line of text is meant to be treated as fine print.

**This line rendered as bold text.**

*This line rendered as italicized text.*

HTML



```
<p>You can use the mark tag to <mark>highlight</mark> text.</p>
<p><del>This line of text is meant to be treated as deleted text.</del></p>
<p><s>This line of text is meant to be treated as no longer accurate.</s></p>
<p><ins>This line of text is meant to be treated as an addition to the document.</ins><
<p><u>This line of text will render as underlined.</u></p>
<p><small>This line of text is meant to be treated as fine print.</small></p>
<p><strong>This line rendered as bold text.</strong></p>
<p><em>This line rendered as italicized text.</em></p>
```

강조하고 싶은 텍스트 사이에 <mark>강조</mark>를 씌우면 된다고 설명하고 있습니다. 결과 텍스트에 mark 태그를 삽입하려면 calculateHandler() 함수에 resultText 변수에 넣어주면 될 것 같습니다.

```
var resultText = `고객님의 ${period}개월 동안 월 상환액은 <mark>${result}원</mark> 입니다.`; $('#result').text(resultText);
```

## 대출 이자 계산기

대출금	<input type="text" value="100000000"/>	원	<input type="button" value="+1백만"/>	<input type="button" value="+1천만"/>	<input type="button" value="+1억"/>
대출금리	<input type="text" value="3"/>	%	<input type="button" value="3%"/>	<input type="button" value="4%"/>	<input type="button" value="5%"/>
대출기간	<input type="text" value="12"/>	개월	<input type="button" value="1년"/>	<input type="button" value="3년"/>	<input type="button" value="5년"/>

계산하기

고객님의 12개월 동안 월 상환액은 <mark>8469370원</mark> 입니다.

이상하게도 <mark> 태그가 텍스트 그대로 출력됩니다. 왜일까요?

<mark> 태그는 HTML 마크업 언어입니다. 근데 우리는 text()라는 함수를 사용해서 텍스트를 출력하라고 명령을 내렸기 때문입니다. HTML 마크업 언어를 출력하기 위해서 text() 대신 html()이라는 함수를 사용합니다.

```
var resultText = `고객님의 ${period}개월 동안 월 상환액은 <mark>${result}원</mark> 입니다.`; $('#result').html(resultText);
```

## 대출 이자 계산기

대출금	100000000	원	+1백만	+1천만	+1억
대출금리	연 3	%	3%	4%	5%
대출기간	12	개월	1년	3년	5년

계산하기

고객님의 12개월 동안 월 상환액은 8469370원 입니다.

### 결과 텍스트 꾸미기 (심화)

결과 텍스트가 금액이 커질 경우 읽기 힘들니 콤마를 붙여서 가독성을 좋게 만들어봅시다. (예: 1000000만원, 1,000,000만원)

구글에 javascript comma number 라고 검색하여 가장 첫 번째 나오는 링크를 클릭합니다.

💡 StackOverflow는 개발자들의 성지로 코딩에 대해 모르는 것을 질문하고 답변하는 세계에서 가장 유명한 서비스입니다.

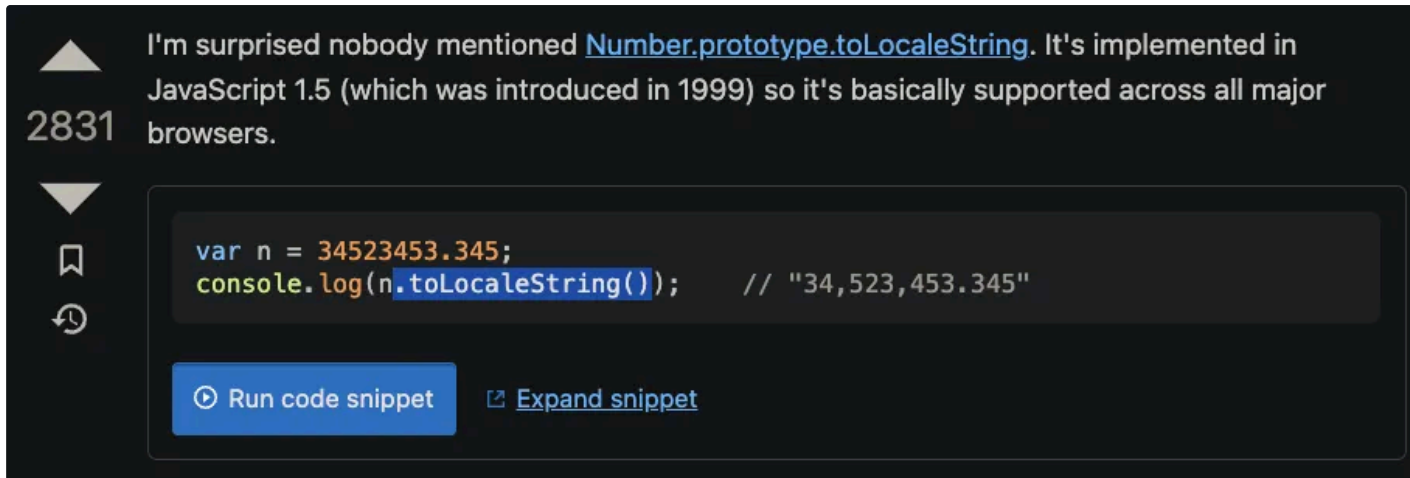
The screenshot shows a Stack Overflow page with a dark theme. The question is 'How to print a number with commas as thousands separators in JavaScript', asked 12 years, 5 months ago, modified 1 month ago, and viewed 1.9m times. The question text is: 'I am trying to print an integer in [JavaScript](#) with commas as thousands separators. For example, I want to show the number 1234567 as "1,234,567". How would I go about doing this?'. The answer, which has 2483 upvotes, is: 'Here is how I am doing it:'. The code provided is: 

```
function numberWithCommas(x) {
  x = x.toString();
  var pattern = /(-?\d+)(\d{3})/;
  while (pattern.test(x))
    x = x.replace(pattern, "$1,$2");
  return x;
}
```

. The answer also includes a comment: 'Is there a simpler or more elegant way to do it? It would be nice if it works with floats also, but that is not necessary. It does not need to be locale-specific to decide between periods and commas.' The right sidebar shows 'The Overflow Blog' with articles like 'Fighting to balance anonymity on the internet' and 'Hashgraph: The next step to blockchain', 'Upcoming Events' including '2022 Community ends November', and 'Featured on Meta' with 'The 2022 Community begun!'.

아래로 내리다보면 우리에게 적절한 함수를 찾을 수 있습니다. Up vote를 2831개나 받은 답변이네요. 이 답변에서는 toLocaleString() 함수를 쓰면 된다고 써있습니다.





▲  
2831 ▼

I'm surprised nobody mentioned [Number.prototype.toLocaleString](#). It's implemented in JavaScript 1.5 (which was introduced in 1999) so it's basically supported across all major browsers.

```
var n = 34523453.345;  
console.log(n.toLocaleString()); // "34,523,453.345"
```

🏠 ⌂ ↺

⌂ Run code snippet 📄 Expand snippet

이 답변을 참고하여 calculateHandler() 함수 내의 result 코드를 다음과 같이 수정합니다. 이제 결과가 깔끔하게 나옵니다.

```
result = Math.round(result).toLocaleString();
```

## 대출 이자 계산기

대출금	<input type="text" value="100000000"/>	원	<input type="button" value="+1백만"/>	<input type="button" value="+1천만"/>	<input type="button" value="+1억"/>
대출금리	<input type="text" value="연 3"/>	%	<input type="button" value="3%"/>	<input type="button" value="4%"/>	<input type="button" value="5%"/>
대출기간	<input type="text" value="12"/>	개월	<input type="button" value="1년"/>	<input type="button" value="3년"/>	<input type="button" value="5년"/>

계산하기

고객님의 12개월 동안 월 상환액은 8,469,370원 입니다.

전체 코드

```

<!DOCTYPE html> <html> <head> <meta charset="UTF-8"> <meta name="viewport"
content="width=device-width, initial-scale=1" /> <title>Hello, world!</title> <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css"
rel="stylesheet"> <link rel="stylesheet" href="style.css"> </head> <body> <main
class="container"> <h1>대출 이자 계산기</h1> <table class="table"> <tbody> <tr> <th
scope="row">대출금</th> <td> <div class="input-group"> <input type="text" class="form-
control" id="loan"> <span class="input-group-text">원</span> </div> </td> <td> <div
class="btn-group" role="group"> <button type="button" class="btn btn-outline-primary"
id="loan-100">+1백만</button> <button type="button" class="btn btn-outline-primary"
id="loan-1000">+1천만</button> <button type="button" class="btn btn-outline-primary"
id="loan-10000">+1억</button> </div> </td> </tr> <tr> <th scope="row">대출금리</th> <td>
<div class="input-group"> <span class="input-group-text">연</span> <input type="text"
class="form-control" id="interest"> <span class="input-group-text">%</span> </div> </td>
<td> <div class="btn-group" role="group"> <button type="button" class="btn btn-outline-
primary" id="interest-3">3%</button> <button type="button" class="btn btn-outline-
primary" id="interest-4">4%</button> <button type="button" class="btn btn-outline-
primary" id="interest-5">5%</button> </div> </td> </tr> <tr> <th scope="row">대출기간
</th> <td> <div class="input-group"> <input type="text" class="form-control" id="period">
<span class="input-group-text">개월</span> </div> </td> <td> <div class="btn-group"
role="group"> <button type="button" class="btn btn-outline-primary" id="period-1">1년
</button> <button type="button" class="btn btn-outline-primary" id="period-3">3년
</button> <button type="button" class="btn btn-outline-primary" id="period-5">5년
</button> </div> </td> </tr> </tbody> </table> <button type="button" id="calculate"
class="btn btn-primary">계산하기</button> <div id="result"></div> </main> <script
src="https://code.jquery.com/jquery-3.6.1.min.js"></script> <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/js/bootstrap.bundle.min.js">
</script> <script src="script.js"></script> </body> </html>

```

index.html

```

function calculateHandler() { var loan = $('#loan').val(); var interest =
$('#interest').val(); var period = $('#period').val(); loan = parseFloat(loan); interest
= parseFloat(interest); period = parseFloat(period); var interestMonth = interest / 12 /
100; var div1 = loan * interestMonth * (1 + interestMonth) ** period; var div2 = (1 +
interestMonth) ** period - 1; var result = div1 / div2; result =
Math.round(result).toLocaleString(); var resultText = `고객님의 ${period}개월 동안 월 상환
액은 <mark>${result}원</mark> 입니다.`; $('#result').html(resultText); } function
loan100Handler() { var loan = $('#loan').val(); if (loan == "") { loan = 0; } loan =
parseFloat(loan); loan = loan + 1000000; $('#loan').val(loan); } function
loan1000Handler() { var loan = $('#loan').val(); if (loan == "") { loan = 0; } loan =
parseFloat(loan); loan = loan + 10000000; $('#loan').val(loan); } function
loan10000Handler() { var loan = $('#loan').val(); if (loan == "") { loan = 0; } loan =
parseFloat(loan); loan = loan + 100000000; $('#loan').val(loan); } function
interest3Handler() { $('#interest').val(3); } function interest4Handler() {
$('#interest').val(4); } function interest5Handler() { $('#interest').val(5); } function
period1Handler() { $('#period').val(12); } function period3Handler() {
$('#period').val(36); } function period5Handler() { $('#period').val(60); }
$(document).ready(function() { $('#calculate').click(calculateHandler); $('#loan-
100').click(loan100Handler); $('#loan-1000').click(loan1000Handler); $('#loan-
10000').click(loan10000Handler); $('#interest-3').click(interest3Handler); $('#interest-
4').click(interest4Handler); $('#interest-5').click(interest5Handler); $('#period-
1').click(period1Handler); $('#period-3').click(period3Handler); $('#period-
5').click(period5Handler); });

```

script.js

## 3-7. 웹 사이트 기본 기능 만들기 - 상단 메뉴, 홈버튼 구현

### 원하는 템플릿 골라서 사용하기

우리는 지금까지 2개의 서비스를 만들었습니다. 이 2개의 서비스는 독립적으로 사용자들에게 배포할 수도 있지만 한 웹사이트 안에 여러가지 서비스가 함께 있을 경우 더 많은 방문자가 더 오래 체류할 가능성이 높습니다. 우리는 사칙연산 계산기와 대출 이자 계산기를 하나의 사이트에서 메뉴로 이동할 수 있게끔 구현해보겠습니다.

만약 여러분들이 강의를 잘 따라오셨다면 VSCode 사이드 메뉴 파일탐색기에 다음과 같이 part1, part2, part3라는 폴더가 있을거예요. 폴더 가장 바깥쪽에 main.html 파일을 새로 만들어 홈페이지로 사용하도록 하겠습니다.

```

  ✓ part1
    <> index.html
    JS script.js
    # style.css
  ✓ part2
    <> boilerplate.html
    <> index.html
    JS script.js
    # style.css
  ✓ part3
    <> index.html
    JS script.js
    # style.css
  <> main.html
```

main.html 파일 안에 채워넣을 템플릿을 찾아봅시다. Bootstrap 예제 중에서 골라볼게요. Navbars 섹션에 있는 Navbar fixed를 사용해보겠습니다.

- Bootstrap 예제: <https://getbootstrap.com/docs/5.2/examples/>
- Navbar fixed 예제 사이트: <https://getbootstrap.com/docs/5.2/examples/navbar-fixed/>

먼저 보일러플레이트 코드를 복사+붙여넣기합니다.

```
<!DOCTYPE html> <html> <head> <meta charset="UTF-8"> <meta name="viewport"
content="width=device-width, initial-scale=1" /> <title>Hello, world!</title> <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css"
rel="stylesheet"> <link rel="stylesheet" href="style.css"> </head> <body> <main
class="container"> <h1>Hello, world!</h1> </main> <script
src="https://code.jquery.com/jquery-3.6.1.min.js"></script> <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/js/bootstrap.bundle.min.js">
</script> <script src="script.js"></script> </body> </html>
```

main.html

```
$(document).ready(function() { });
```

script.js

```
/* 비어있음 */
```

style.css

Navbar fixed 예제에서 오른쪽 클릭 페이지 소스 보기 또는 보기 - 개발자 정보 - 소스 보기 메뉴를 클릭하여 소스코드를 확인합니다. 먼저 body 부분의 nav, main 태그를 전부 복사하여 main.html 파일의 body 부분에 붙여넣습니다.

```

<nav class="navbar navbar-expand-md navbar-dark fixed-top bg-dark"> <div
class="container-fluid"> <a class="navbar-brand" href="#">Fixed navbar</a> <button
class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target="#navbarCollapse" aria-controls="navbarCollapse" aria-expanded="false" aria-
label="Toggle navigation"> <span class="navbar-toggler-icon"></span> </button> <div
class="collapse navbar-collapse" id="navbarCollapse"> <ul class="navbar-nav me-auto mb-2
mb-md-0"> <li class="nav-item"> <a class="nav-link active" aria-current="page"
href="#">Home</a> </li> <li class="nav-item"> <a class="nav-link" href="#">Link</a> </li>
<li class="nav-item"> <a class="nav-link disabled">Disabled</a> </li> </ul> <form
class="d-flex" role="search"> <input class="form-control me-2" type="search"
placeholder="Search" aria-label="Search"> <button class="btn btn-outline-success"
type="submit">Search</button> </form> </div> </div> </nav> <main class="container"> <div
class="bg-light p-5 rounded"> <h1>Navbar example</h1> <p class="lead">This example is a
quick exercise to illustrate how fixed to top navbar works. As you scroll, it will remain
fixed to the top of your browser's viewport.</p> <a class="btn btn-lg btn-primary"
href="/docs/5.2/components/navbar/" role="button">View navbar docs &raquo;</a> </div>
</main>

```

Fixed navbar

Home

Link

Disabled

Search

Search

## Navbar example

This example is a quick exercise to illustrate how fixed to top navbar works. As you scroll, it will remain fixed to the top of your browser's viewport.