

3章 計算しよう

2章までは文字をコンソールに表示する処理を作成しました。この章では、小学校で習った、たし算・ひき算などの四則演算を行うプログラムを作成します

ワーク1 四則演算の計算を試みよう

console.logを利用して、簡単な計算を行いましょう。たし算、ひき算、かけ算、わり算を実行するプログラムです。以下のようなプログラムを作成しましょう

```
/js-level1/chapter03/work01.js
```

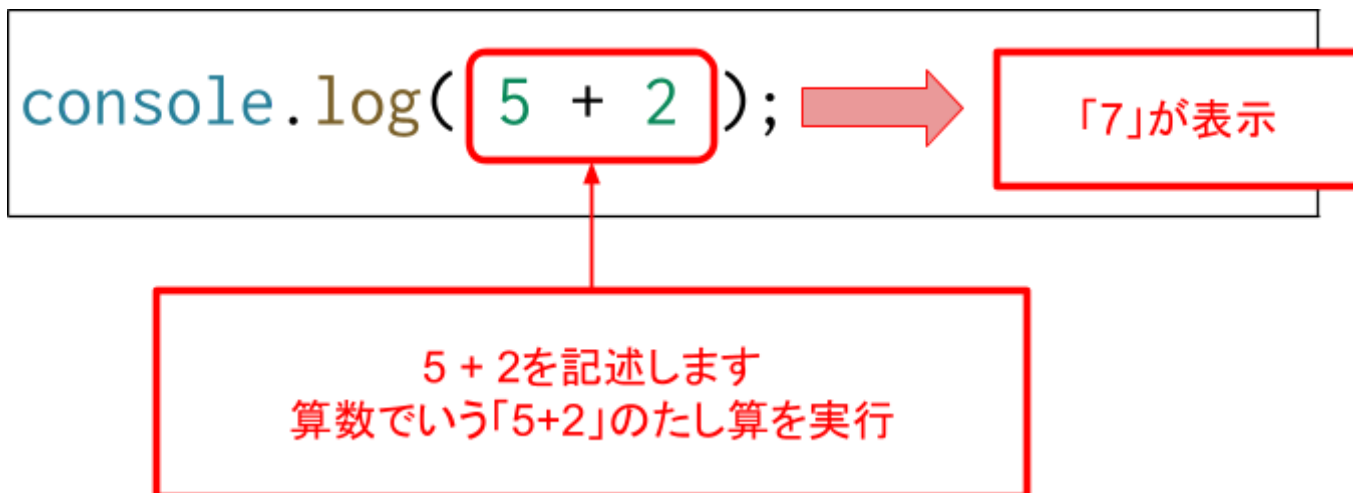
```
console.log(5 + 2);  
console.log(5 - 2);  
console.log(5 * 2);  
console.log(5 / 2);  
console.log(5 % 2);
```

実行結果

```
7  
3  
10  
2.5  
1
```

プログラムの解説

今回のプログラムでは計算した結果を画面に表示します。「console.log」に計算式を記述します



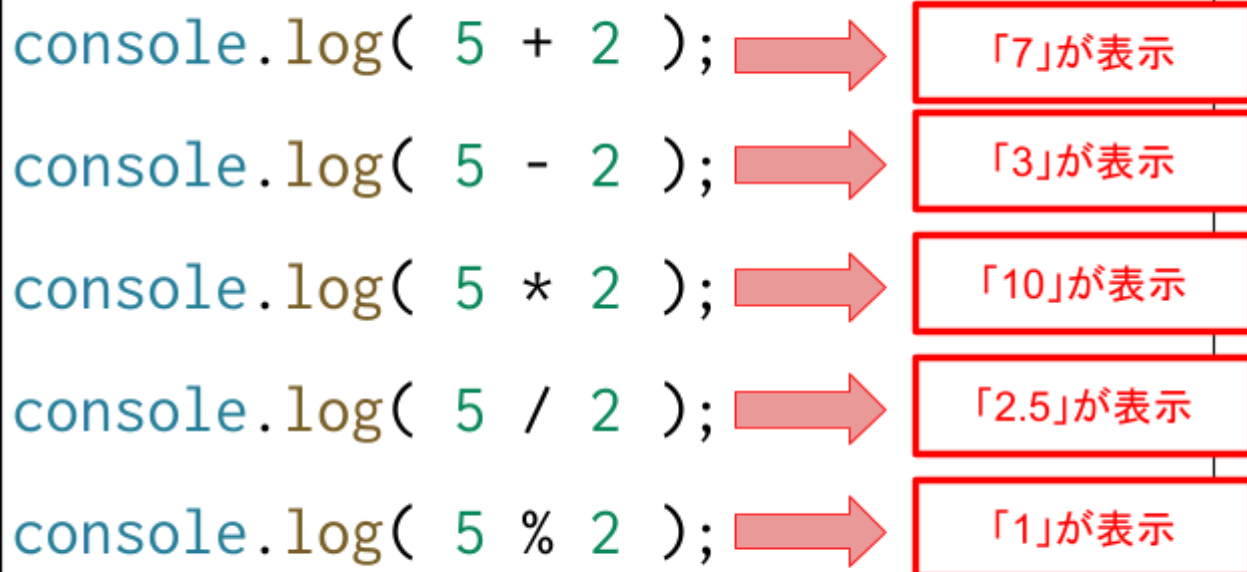
たし算以外にも計算することができます

記号	意味	実行例	結果
+	たし算	5 + 2	7
-	ひき算	5 - 2	3
*	かけ算	5 * 2	10
/	わり算	5 / 2	2.5
%	剰余	5 % 2	1

「+」「-」は算数と同じですが、かけ算は「*」（アスタリスク）という記号を使います。また、わり算は「/」（スラッシュ）という記号を使うことに注意しましょう

剰余とは、わり算の「余り」を計算する記号です。「5 % 2」とは、「5 / 2」の計算を行い、余りである「1」を答えとして返します

プログラム全体は以下のように計算されます



たし算、ひき算、かけ算、わり算、剰余の計算を行う事ができます

ワーク2 簡単な計算を試みよう (2)

先ほどの例よりも、すこしだけ複雑な計算を行います。数学と同じように「()」を利用して計算の優先順位を変更します。以下のプログラムを作成して実行しましょう

```
/js-level1/chapter03/work02.js
```

```
console.log(2 + 2 * 2);  
console.log((2 + 2) * 2);  
  
console.log(6 - 2 * 2);  
console.log((6 - 2) * 2);
```

実行結果

```
6  
8  
2  
8
```

計算の優先順位を変更する

計算を行うための記号には優先順位があります。通常は数学と同じように、かけ算、わり算、たし算、ひき算となります。しかし、数学と同様に、() で囲むと計算の優先順位が上がります

丸カッコ () がない

「 2×2 」が行われて「4」、その後、「 $2 + 4$ 」が実行されて「6」となります

```
console.log(2 + 2 * 2);
```

丸カッコ () が優先して計算される

() 内が先に計算されます。($2 + 2$)が行われて「4」、「 4×2 」が実行されて「8」となります

```
console.log((2 + 2) * 2);
```

丸カッコ () がない

「 2×2 」が行われて「4」、その後、「 $6 - 4$ 」が実行されて「2」となります

```
console.log(6 - 2 * 2);
```

丸カッコ () が優先して計算される

() が先に計算されます。 $(6 - 2)$ が行われて「4」、 4×2 が実行されて「8」となります

```
console.log((6 - 2) * 2);
```

ワーク3 練習問題

以下のような計算を実行するプログラムを作成しましょう

console.logで計算式のプログラムを作成します。計算結果として、「結果」を表示します

ファイル名	/js-level1/chapter03/work03.js
-------	--------------------------------

計算	$4 \times 3 + 2$
結果	14

計算	$(18 \div 3) - 1$
結果	5

計算	$(4 - 2) \times (8 - 3)$
結果	10

ワーク4 練習問題

以下のような計算を実行するプログラムを作成しましょう

ファイル名	/js-level1/chapter03/work04.js
-------	--------------------------------

計算	$5 \times (5 - 2)$
結果	15

計算	$10 / (4 - 2)$
結果	5

計算	$(5 \times 3) / (3 + 2)$
結果	3

ワーク5 HTMLとJavaScriptで画面のある処理を作成しよう（たし算）

これまで作成したプログラムは画面下部に「文字」として表示されるプログラムでした。ここではHTMLを利用して「ウィンドウ」で表示されるプログラムを作成します。今回は、「HTML」で画面を表示する処理を作成します。その上で、「JavaScript」で計算を行う処理を作成します

まずは、「/js-level1/chapter03/work05.html」のファイルを開きましょう。ほとんどの部分が作られています

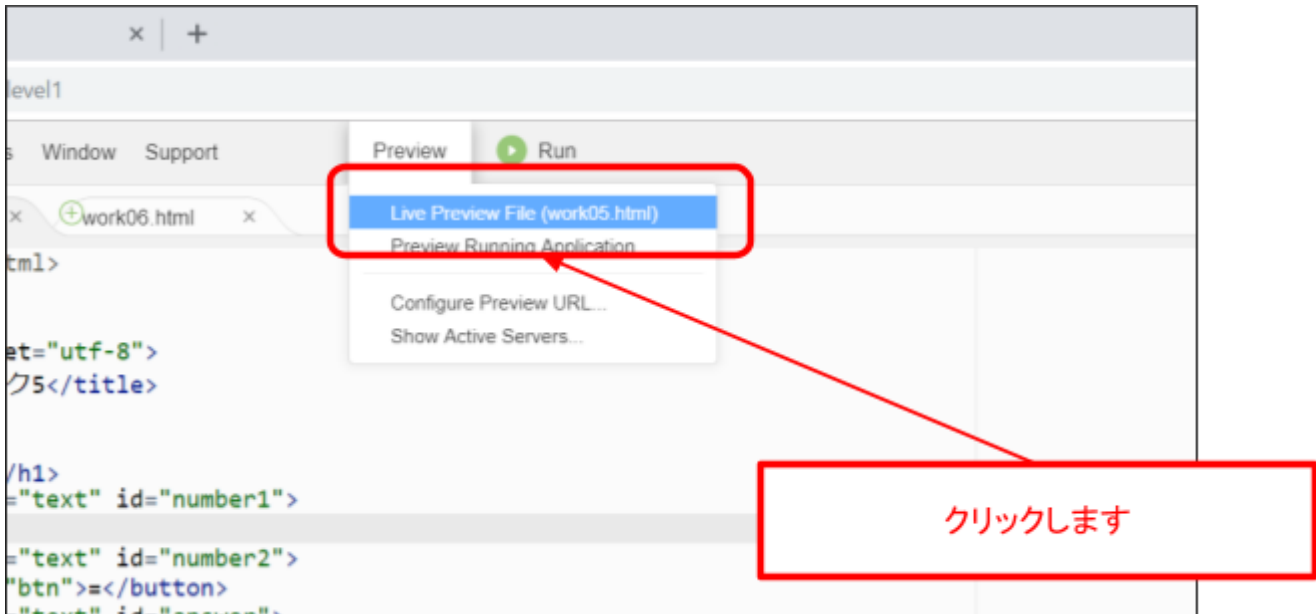
/js-level1/chapter03/work05.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>ワーク5</title>
</head>
<body>
<h1>たし算</h1>
<input type="text" id="number1">
+
<input type="text" id="number2">
<button id="btn">=</button>
<input type="text" id="answer">

<script>
// ここからJavaScriptのプログラム
var btn = document.getElementById("btn");
btn.onclick = function() {
    // ここにプログラムを作成します
}
</script>
</body>
</html>
```


まずは、プログラムを作成する前に現在の状態で動かしてみましよう

画面内の「Preview」をクリックします。表示された項目内の「Live Preview File(work05.html)」をクリックします



画面右側に表示されると成功です



このアプリケーションは、四角のボックスに数字を入力して「=」のボタンをクリックすると、たし算を行います。たし算の「和」を右側のボックスに表示します。ただし、現時点では、プログラムは未完成なので、動作しません

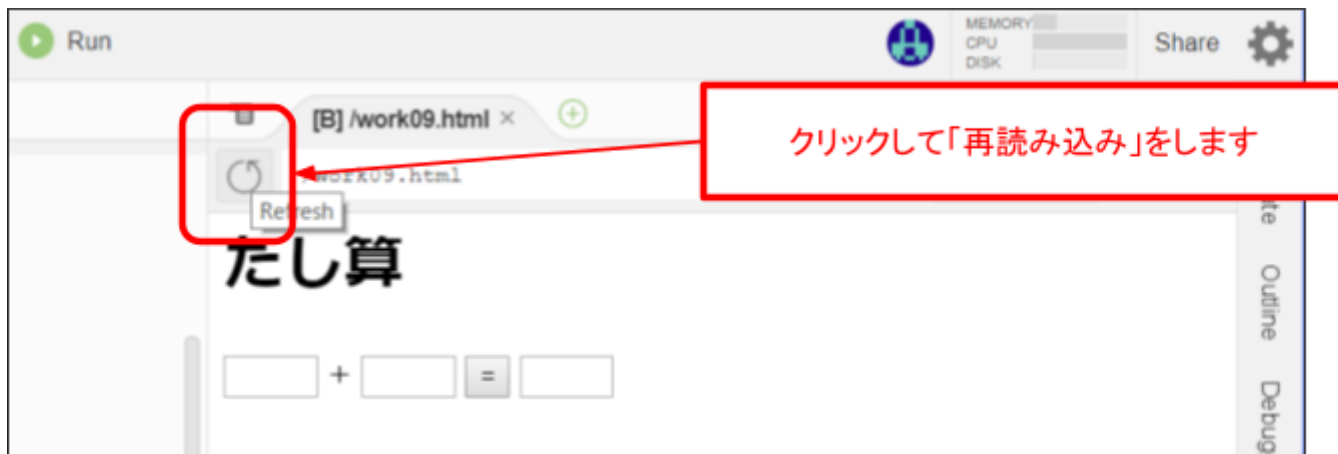


それでは、プログラムを作成しましょう。基本的な部分は作成済みです。赤字の箇所のみが追加する部分です

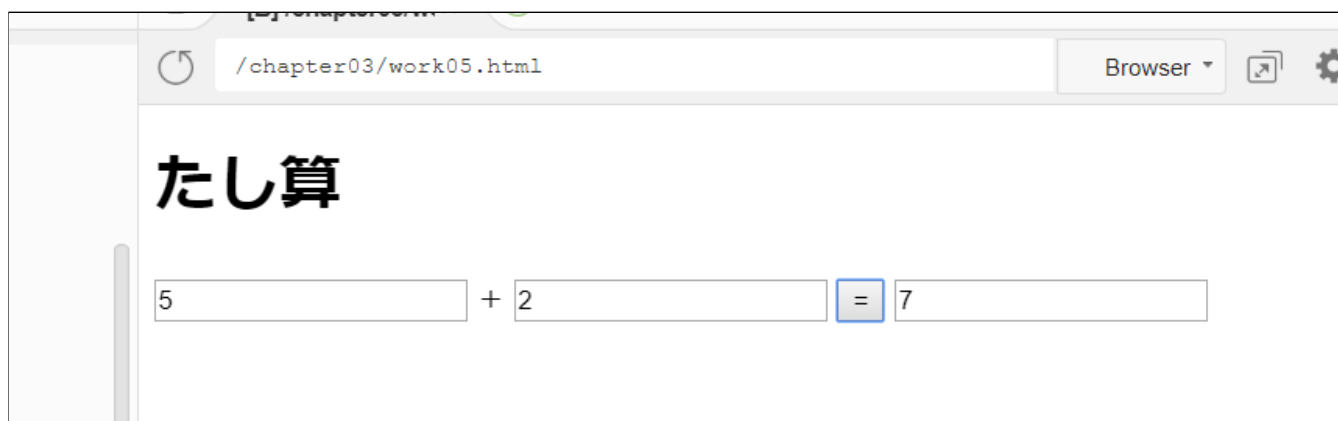
/js-level1/chapter03/work05.html

```
<script>
var btn = document.getElementById("btn");
btn.onclick = function() {
  // ここにプログラムを作成します
  var number1 = document.getElementById("number1").value;
  var number2 = document.getElementById("number2").value;
  var answer = parseInt(number1) + parseInt(number2);
  document.getElementById("answer").value = answer;
}
</script>
```

プログラムの作成が完了したら、以下の部分をクリックして再読み込みをします



左側のボックスに数値を入力して、「=」ボタンをクリックします。右側のボックスに「たし算」の答えが表示されている場合、正しく動作しています



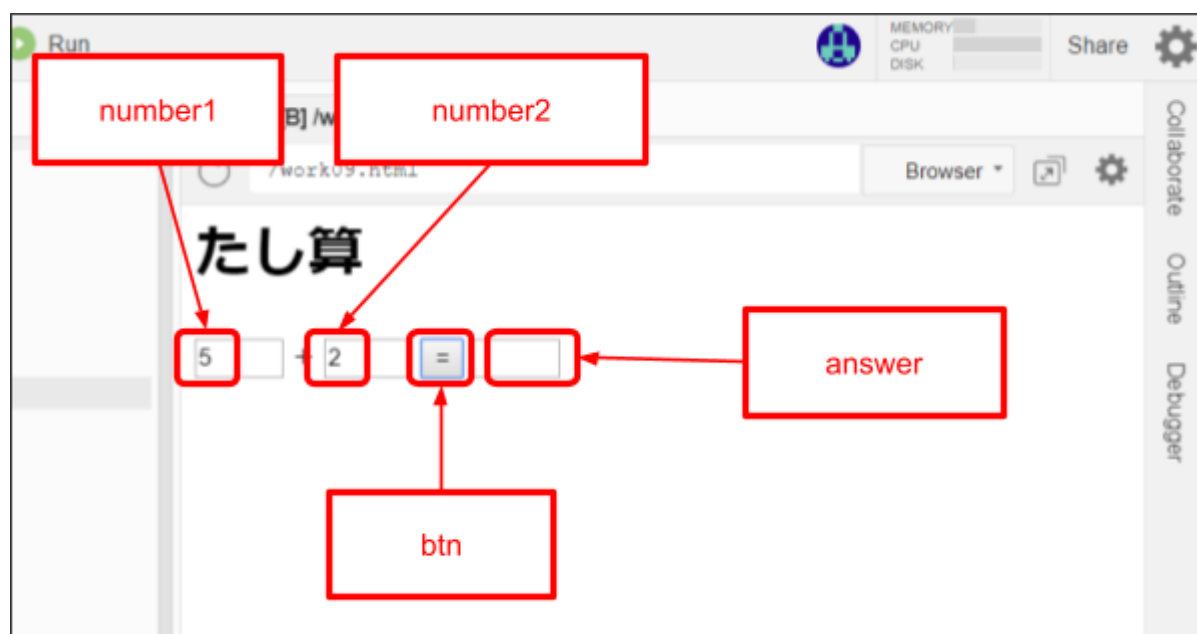
プログラムの解説

事前に作られているプログラムでは、HTMLで入力ボックスやボタンの表示部分が作られています。それぞれの入力ボックスやボタンには「名前」がつけられています

HTMLパーツ

以下の部分です

```
<input type="text" id="number1">
+
<input type="text" id="number2">
<button id="btn">=</button>
<input type="text" id="answer">
```



左側の入力ボックスには、「number1」「number2」、「=」は「btn」、答えが表示されるボックスには「answer」という名前がつけられています

「=」 ボタンを取得する

まずは、「=」 ボタンを取り出す処理を行います。「=」 ボタンにはHTMLで以下のように設定されています。<button>のid属性で「btn」という値が割り当てられています

HTML部分

```
<button id="btn">=</button>
```

そのため、「id」が「btn」のタグを取得する処理を行います。「document.getElementById」という文で、「btn」という名前のボタンを取得します。変数（へんすう）「btn」で「=」のボタンを表します。変数については、次の章で説明します。現時点では、「変数というものにデータが格納されている」という理解で構いません

JavaScript部分

```
var btn = document.getElementById("btn");
```

「=」 ボタンがクリックされたときの動作

次に、「=」 ボタンがクリックされたときの動作を作成します。以下のようにすると、「btn」がクリックされたときの動作を作成することができます。{}の内側にボタンがクリックされた時の処理を作成します

JavaScript部分

```
btn.onclick = function() {  
    // ボタンがクリックされたときの処理をここに記述します  
}
```

ボタンがクリックされたときの処理手順は以下のようになります

1. 「number1」のタグに入力された値を取り出す
2. 「number2」のタグに入力された値を取り出す
3. 「number1」の値と「number2」の値を足して、「答え」を計算する
4. 「answer」のタグに「答え」を代入する

「number1」のタグに入力された値を取り出す

「number1」に入力されている値を取り出して、変数「number1」に代入します

JavaScript部分

```
var number1 = document.getElementById("number1").value;
```

「number2」のタグに入力された値を取り出す

「number2」に入力されている値を取り出して、変数「number2」に代入します

JavaScript部分

```
var number2 = document.getElementById("number2").value;
```

「number1」の値と「number2」の値を足して、「答え」を計算する

「parseInt」は文字列を数字に変換する処理です。入力ボックスから取得した値は「文字列」です。そのため、「parseInt」を利用して、「文字列」を「数字」に変換します。その上で、「number1」と「number2」を合計します。合計した結果を変数「answer」に代入します

JavaScript部分

```
var answer = parseInt(number1) + parseInt(number2);
```

「answer」のタグに「答え」を代入する

「answer」の値を、テキストボックス「answer」に代入します

JavaScript部分

```
document.getElementById("answer").value = answer;
```

このワークでは以上のような流れで「たし算の処理」を行います

HTMLとJavaScriptを利用して画面で動作するプログラムを作成できます

ワーク6 練習問題 かけ算

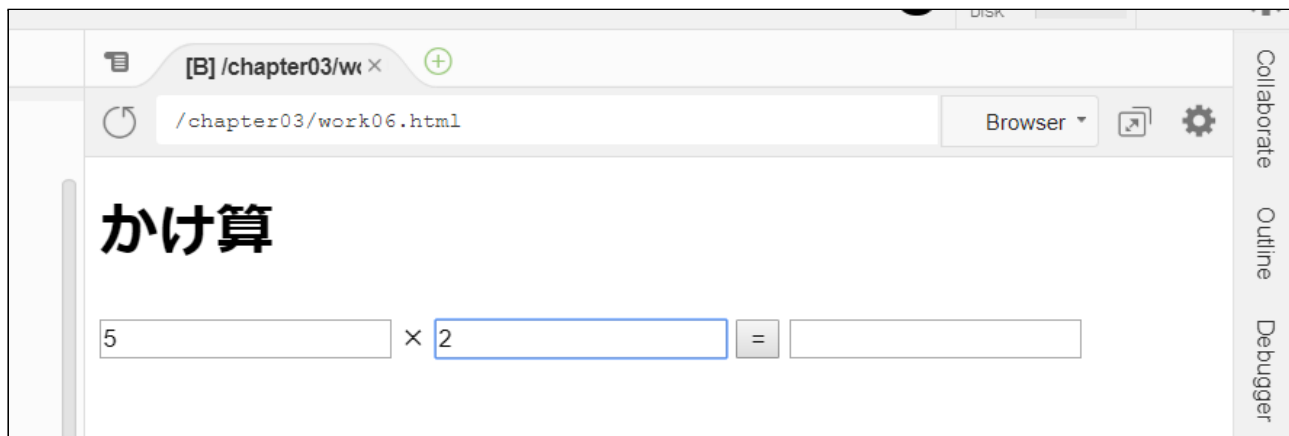
最後に、「かけ算」を行うプログラムを[自分で作成してみましょう](#)。まずは完成後の実行結果を確認します

実行結果

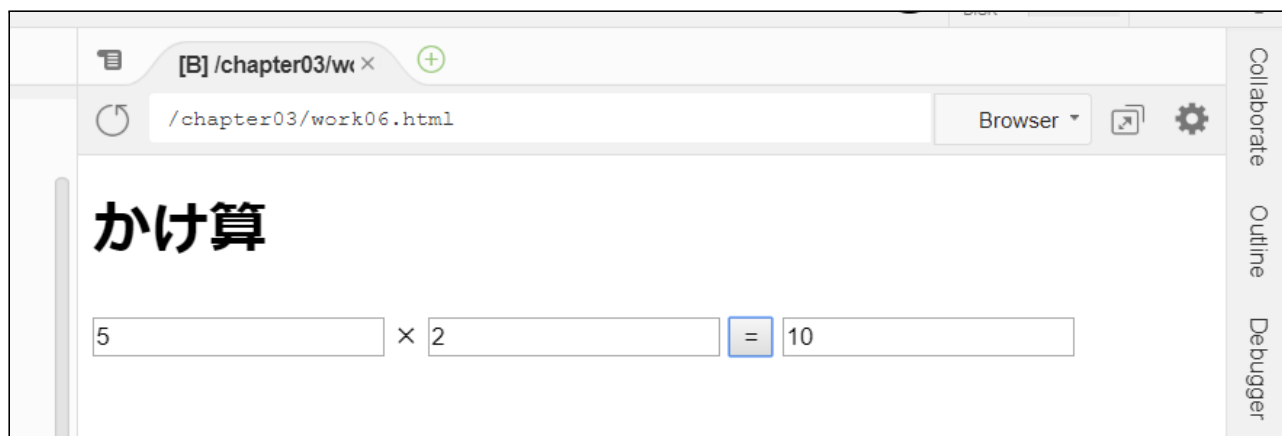
画面を開いた状態



数字を入力して「=」ボタンをクリックします



「=」 をクリックすると、かけ算の答えが右側のボックスに表示されます



基本的な部分は提供されています。「ここにプログラムを作成します」よりも下の部分にプログラムを作成します。「ワーク5」で作成した「たし算」のプログラムを参考にしながら進めていきましょう

「ワーク5」に作成した処理は「たし算」でした。ここで作成する処理は「かけ算」です。基本的な処理の流れは同じです。たし算では「+」記号で計算しました。かけ算にするためには、「一文字」だけ記号を変更する必要があります

/js-level1/chapter03/work06.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>ワーク6</title>
</head>
<body>
<h1>かけ算</h1>
<input type="text" id="number1">
x
<input type="text" id="number2">
<button id="btn">=</button>
<input type="text" id="answer">
<script>
var btn = document.getElementById("btn");
btn.onclick = function() {
    // ここにプログラムを作成します
}
</script>
</body>
</html>
```

次ページに続く

作成するためのポイント

「ワーク5」で作成したのは、

ボタンがクリックされると「number1」と「number2」の「たし算」の答えを「answer」代入する

でした。今回作成するのはほとんど同じです

ボタンがクリックされると「number1」と「number2」の「かけ算」の答えを「answer」代入する

です。ボタンがクリックされたときの処理手順は以下のようになります

「number1」のタグに入力された値を取り出す

「number2」のタグに入力された値を取り出す

「number1」の値と「number2」の値をかけ算した「答え」を計算する

「answer」のタグに「答え」を代入する

「ワーク5」で作成した処理を参考にしながら、「たし算」を「かけ算」に変更するだけで完成します