



3章 データの扱い方を理解しよう

数値や文字列など、JavaScriptで扱うデータについて解説します。

🕒 60分 🏆 - 📖 読了

3.1 本章の目標

本章では以下を目標にして学習します。

- JavaScriptで使われるデータの種類（データ型）を知ること
- JavaScriptでデータを出力する方法を知ること
- JavaScriptで簡単な計算を行い、実際に計算結果を確認すること

JavaScriptをはじめとするプログラミング言語では、文字列を表示したり数値を計算したり、条件に当てはまるかどうかで処理を変えたりする際に、さまざまなデータを扱います。

JavaScriptの学習を進めるうえで、データの種類や扱い方に関する知識は欠かせません。これらの知識がないと、今後の学習の理解度が下がってしまいます。

本章でデータの種類や扱い方をしっかりと学び、次章以降の学習をスムーズに進められるようにしておきましょう。

3.2 データの種類（データ型）

JavaScriptをはじめとするプログラミング言語では、以下のようにさまざまなデータを扱います。

- 「こんにちは」という文字列データ
- 「100」という数値データ
- 「2022/12/31」という日付データ

プログラミング言語におけるこのようなデータの種類のことを、**データ型**といいます。

JavaScriptではまず、代表的な以下3つのデータ型を覚えましょう。

- 文字列型
- 数値型
- 論理型

では順番に解説します。なお3節までは座学のみですが、4節からは実際にJavaScriptを書いていきます。

+ 質問する





1. 文字列型



文字列型は「こんにちは」「Hello, world!」など、文字列を表すデータ型のことです。

なお、JavaScriptをはじめとするプログラミング言語では、「果物」などの単語や「私は果物を食べたいです」などの文章のことを、まとめて「**文字列**」と呼びます。普段の生活ではあまり使いませんが、プログラミングではよく使う言葉なので覚えておきましょう。

JavaScriptでは、以下のようにシングルクォーテーションまたはダブルクォーテーションで文字列を囲むことで、文字列型のデータとしてコンピュータに扱われます。

JSファイル（見本）

```
1 console.log(' こんにちは');
2 console.log("今日はいい天気ですね");
3
```

シングルクォーテーションとダブルクォーテーションのどちらで囲んでも構いませんが、どちらかに統一するようにしましょう。本教材ではシングルクォーテーションで統一します。

なお、シングルクォーテーションはWindows、Macともに「shift + 7」で入力できます。

2. 数値型



数値型



数値型は「100」「9999」など、数値を表すデータ型のことです。

JavaScriptでは、以下のように数値をそのまま記述することで数値型になります。なお、数値は**半角**で入力しないとエラーが発生するので注意してください。

JSファイル（見本）

```
1 console.log(100);
2
```

3. 論理型

論理型



論理型は true（真）または false（偽）、つまり「本当」か「嘘」かという論理値（真偽値）を表すデータ型のことです。6、7章で条件分岐を学ぶときに実際に使うので、現時点では「こんなデータ型があるんだな」程度で十分です。

JavaScriptでは、以下のように true または false をそのまま記述することで、論理型のデータとしてコンピュータに扱われます。

JSファイル（見本）



```
1 console.log(true);
2 console.log(false);
3
```

論理型は true または false の2つの値しか存在しません。

なお、シングルクォーテーションまたはダブルクォーテーションで囲んだ場合は、論理型ではなく文字列型になります。

3.3 データを出力する方法を学ぼう

次節から実際にJavaScriptのコードを書いていきますが、その前にまず、データを出力する方法を学びましょう。

「データを出力する」とはということか

「データを出力する」とは、コンピュータの処理結果（データ）をディスプレイ（表示装置）やプリンタなどの周辺装置に送信することです。

なぜデータを出力する必要があるのか

文字や画像を配置するHTMLや、見た目を装飾するCSSであれば、コードを書いたあとにブラウザで開けばすぐに実行結果を確認できます。

しかし、JavaScriptをはじめとするプログラミング言語の場合、そうはいきません。

なぜなら、プログラミング言語による処理（プログラム）はブラウザやコンピュータの**内部**で実行されるものであり、ブラウザ上にわかりやすく実行結果が表示されるものではないからです。

つまりブラウザを開いただけでは、書いたコード（プログラム）が正しく実行されているかどうかを確認できません。

そこで便利なのがコンソールです。コンソールにデータを出力すれば、プログラムが正しく実行されているかどうかを簡単に確認できます。

- コンソール=コンピュータの入出力を担当する装置のこと



HTML・CSS



ブラウザですぐに確認できる

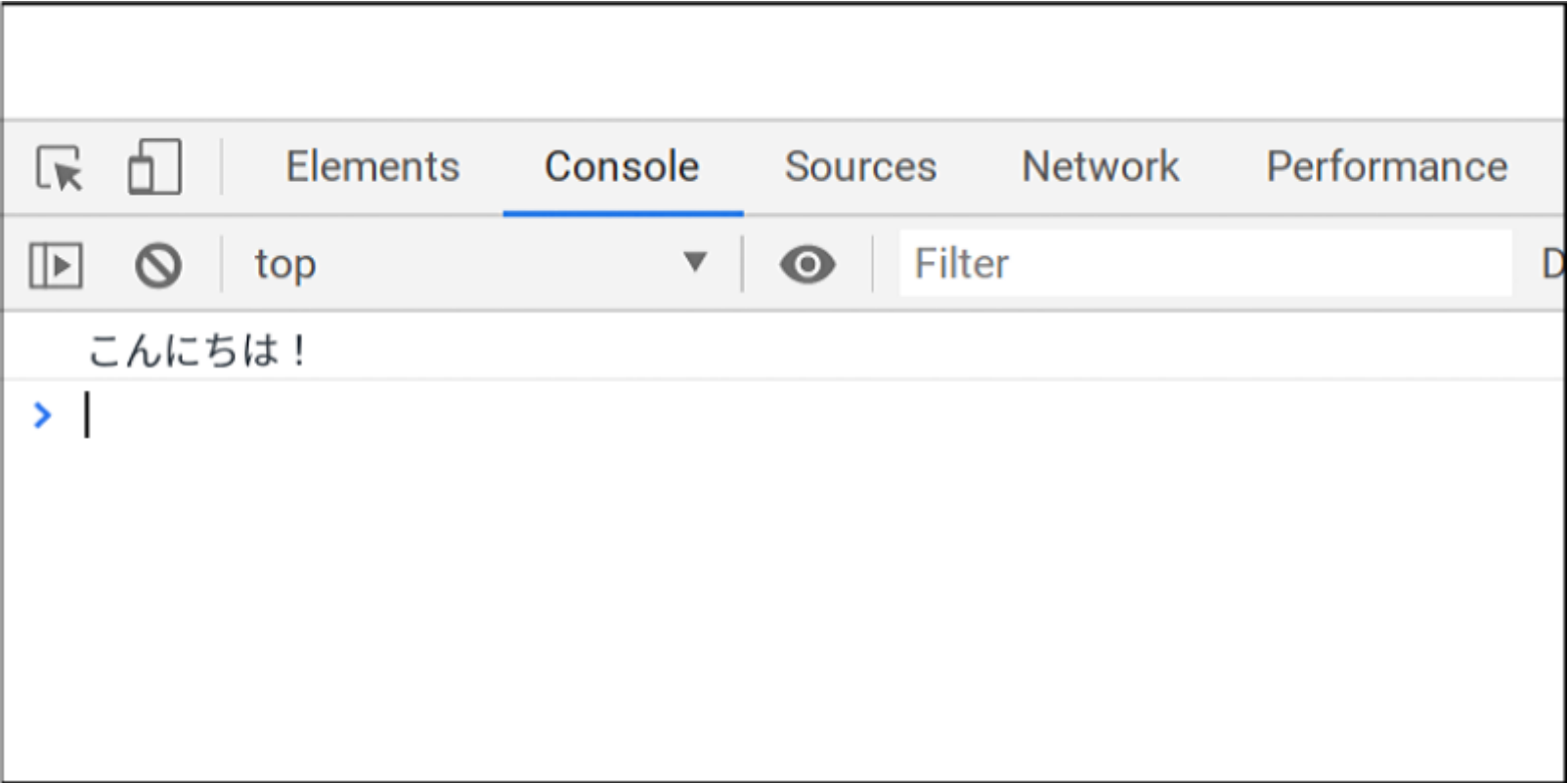
プログラミング言語



コンピューターの内部で実行されるので
コンソールで確認する

デベロッパーツールを使ってコンソールを表示する

ブラウザのデベロッパーツールを使えば、以下のようにコンソールを表示できます。



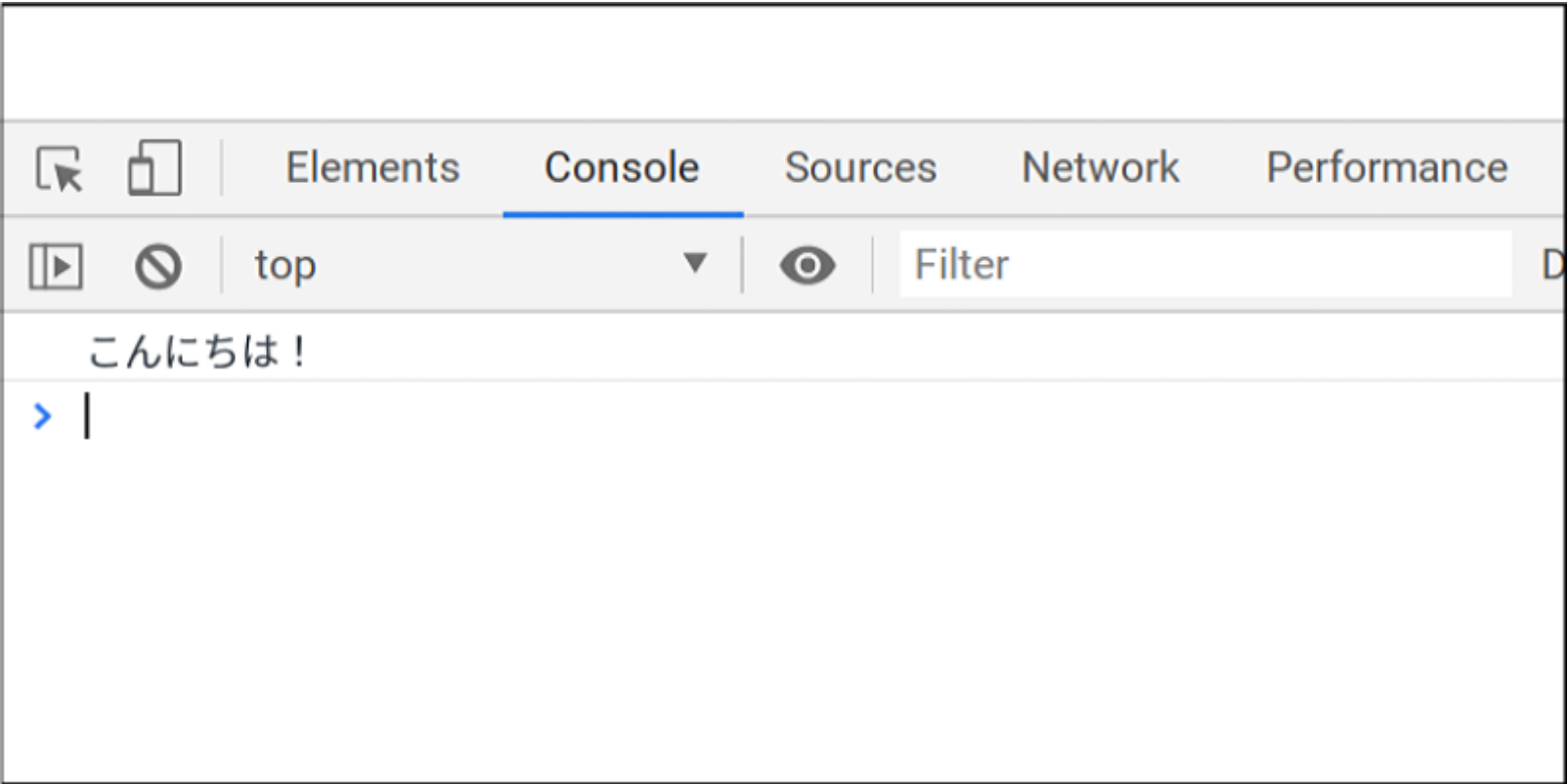
本教材では、デベロッパーツールのコンソールを利用します。デベロッパーツールの開き方は、以下の教材を参考にしてください。

- 『ITリテラシーを身につけ学習を効率的に進めよう 11章 Google Chromeのデベロッパーツールを使いこなそう』

データの出力方法

JavaScriptでデータを出力するには、`console.log()` を記述すればOKです。丸括弧 `()` の中に出力したいデータを記述します。

例えば `console.log('こんにちは！');` と記述すると、コンソールには以下のように表示されます。



座学が続いたので大変だったと思います。お疲れさまでした。

次からはいよいよJavaScriptを書いていくので、ここで一度休憩しておきましょう。

3.4 簡単な計算（四則演算）をしてみよう

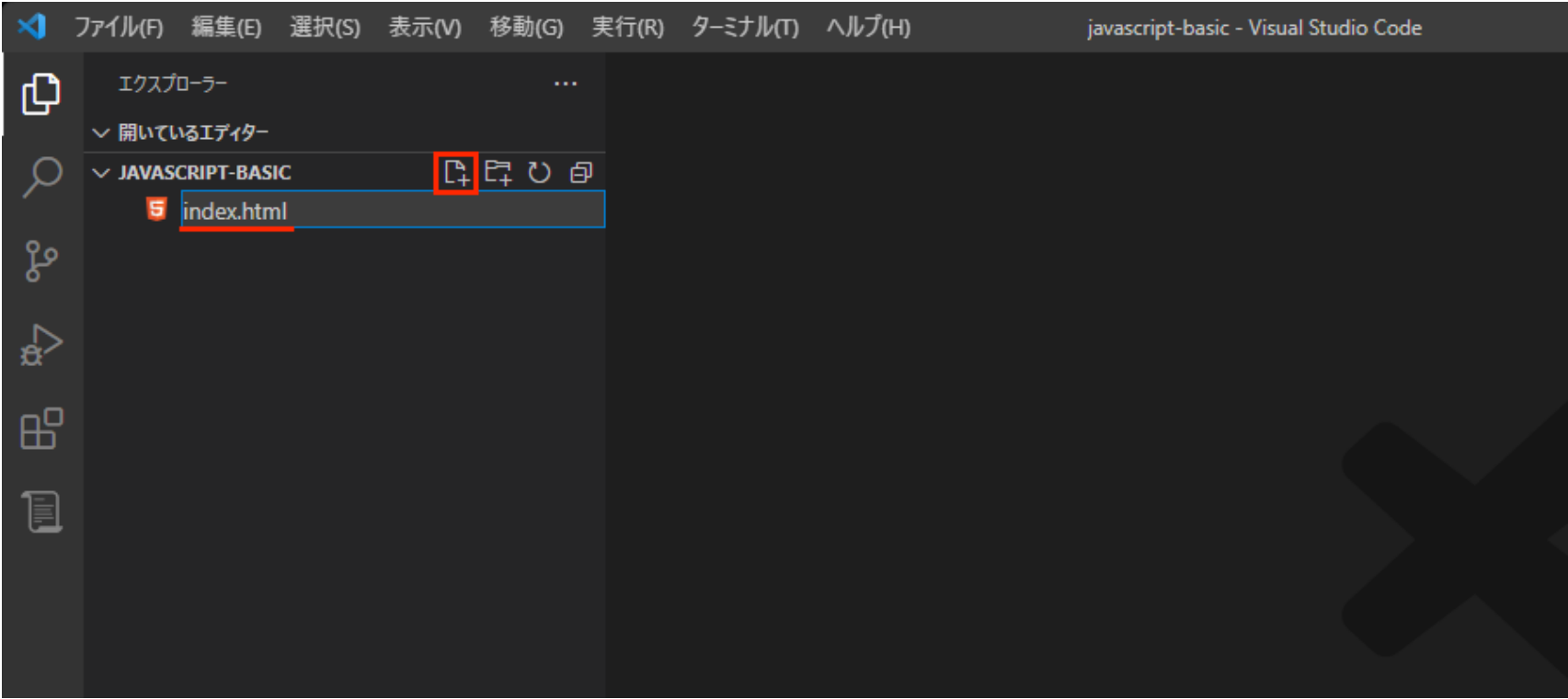
数値型のデータを使い、四則演算（足し算・引き算・掛け算・割り算）をしてみましょう。

JavaScriptでは、以下の算術演算子を使うことで簡単に四則演算ができます。

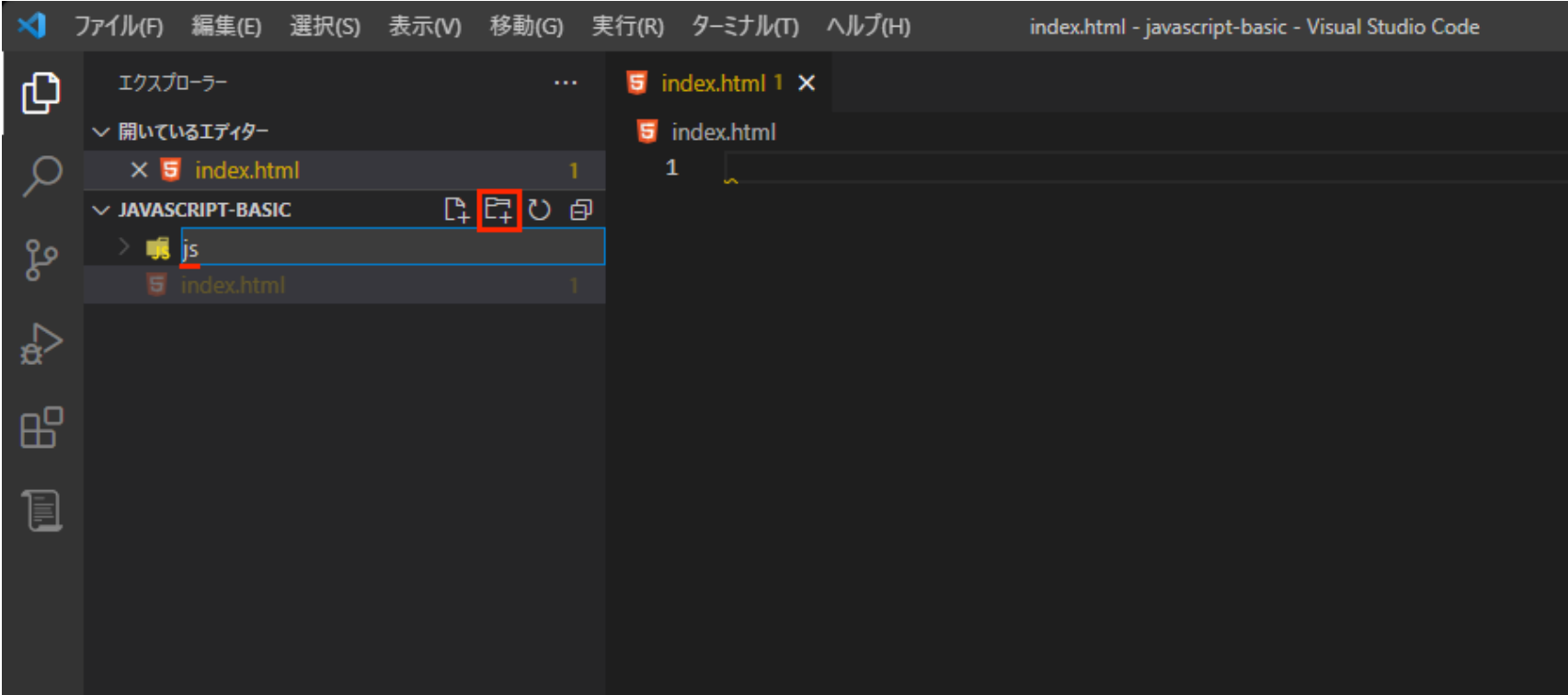
算術演算子	処理の内容	例（出力結果）
+	左の値と右の値を 足す 。	45 + 18 （ 63 ）
-	左の値から右の値を 引く 。	30 - 12 （ 18 ）
*	左の値と右の値を 掛ける 。	15 * 6 （ 90 ）
/	左の値を右の値で 割る 。	30 / 3 （ 10 ）
%	左の値を右の値で割った 余り を求める。	45 % 8 （ 5 ）

ではまず、任意の場所（特にこだわりがなければデスクトップなど）に javascript-basic というフォルダを作成してください。続いて Visual Studio Codeを開き、画面上部の「ファイル > フォルダーを開く」から javascript-basic フォルダを開きましょう。

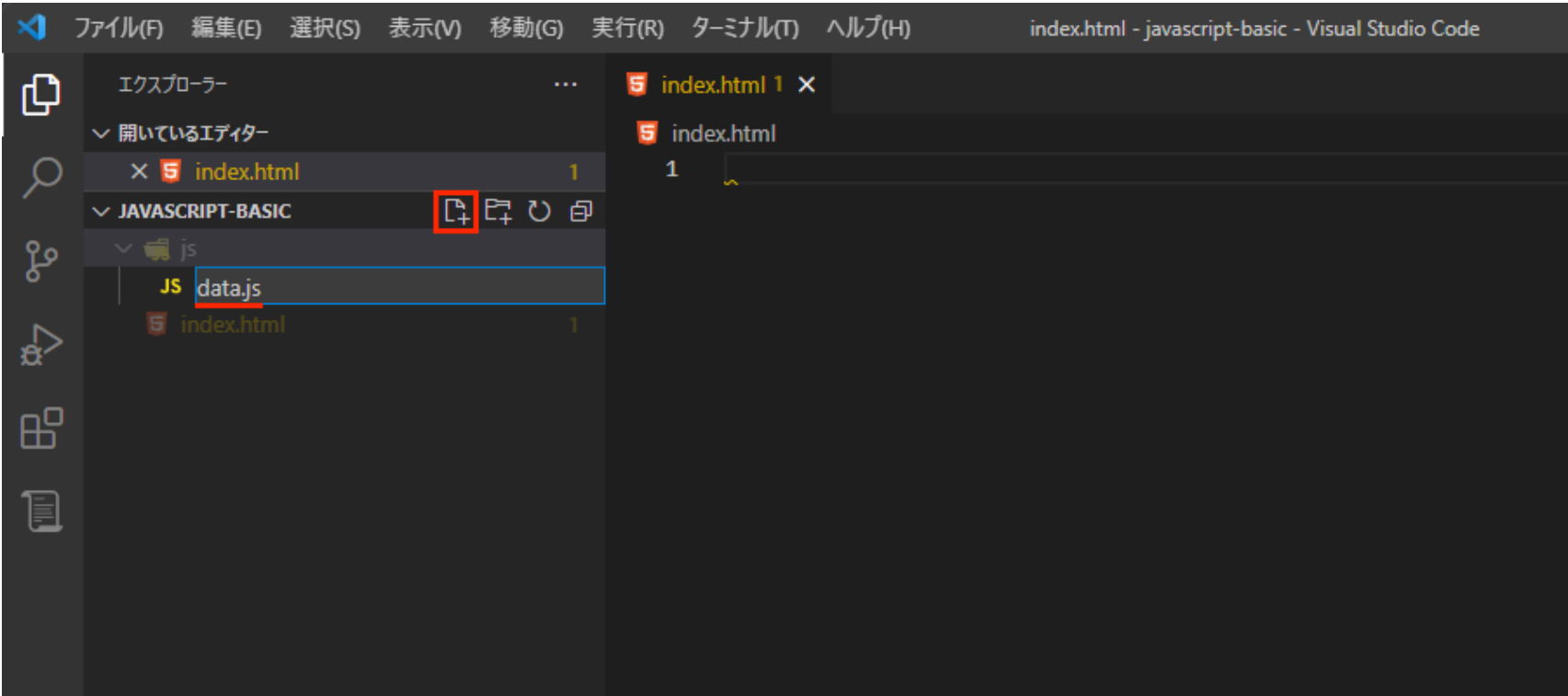
javascript-basic フォルダを開いたら、まず以下のように index.html を作成してください。



続いて、js という名前のフォルダを作成しましょう。



次に、js フォルダの中に data.js という名前のファイルを作成してください。js フォルダを選択した状態で「新しいファイル」ボタンをクリックすれば、js フォルダの中に新規ファイルを作成できます。



最終的に、以下のディレクトリ構成になっていればOKです。



```
1 javascript-basicフォルダ
2 |   └─ jsフォルダ
3 |       └─ data.js
4 |   └─ index.html
5
```

では data.js を以下のように編集し、四則演算の結果をコンソールに出力しましょう。

data.js

```
1 + // 四則演算
2 + console.log(45 + 18);
3 + console.log(30 - 12);
4 + console.log(15 * 6);
5 + console.log(30 / 3);
6 + console.log(45 % 8);
7
```

なお、今後は編集したらファイルを保存する習慣を身につけてください。Windowは「Ctrl + S」キー、Macは「command + S」キーでファイルを保存できます。

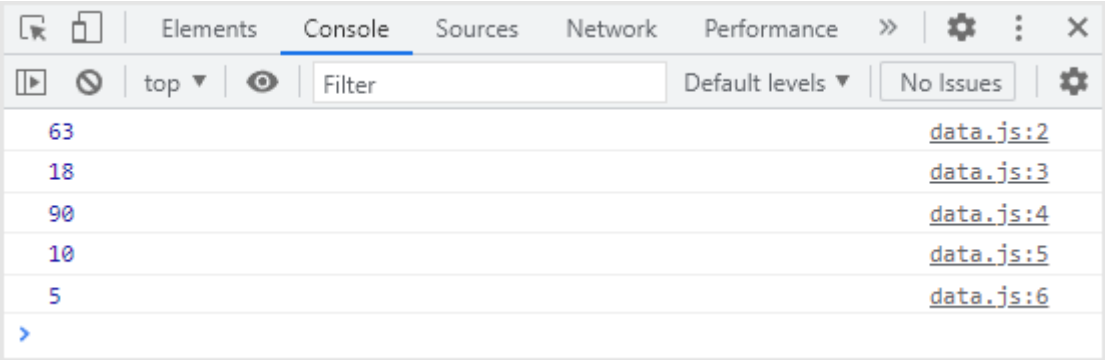
では続いて index.html を以下のように編集し、 data.js を読み込みましょう。

index.html

```
1 + <!DOCTYPE html>
2 + <html lang="ja">
3 +
4 + <head>
5 +   <meta charset="UTF-8">
6 +   <title>JavaScript基礎編</title>
7 + </head>
8 +
9 + <body>
10 +   <script src="js/data.js"></script>
11 + </body>
12 +
13 + </html>
14
```

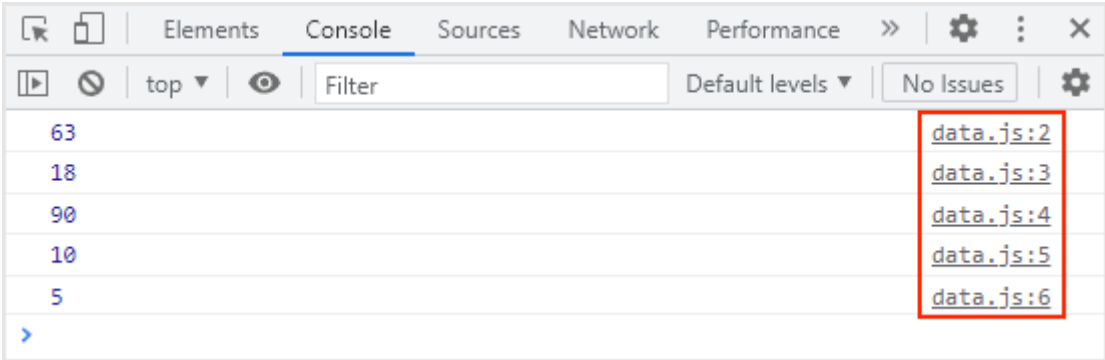
index.html をブラウザで開き、デベロッパーツールのコンソールを確認してみましょう。なお、デベロッパーツールの開き方は[前述の教材](#)を参考にしてください。

以下のように、四則演算の実行結果が表示されていればOKです。



HTML要素は記述していないのでブラウザには白い画面が表示されるだけですが、コンソールを確認すると、きちんとJavaScriptの処理が実行されていることがわかります。

なお、コンソールの右側に表示されているのは処理を実行しているファイル名とその行数です。



3.5 文字列を連結してみよう

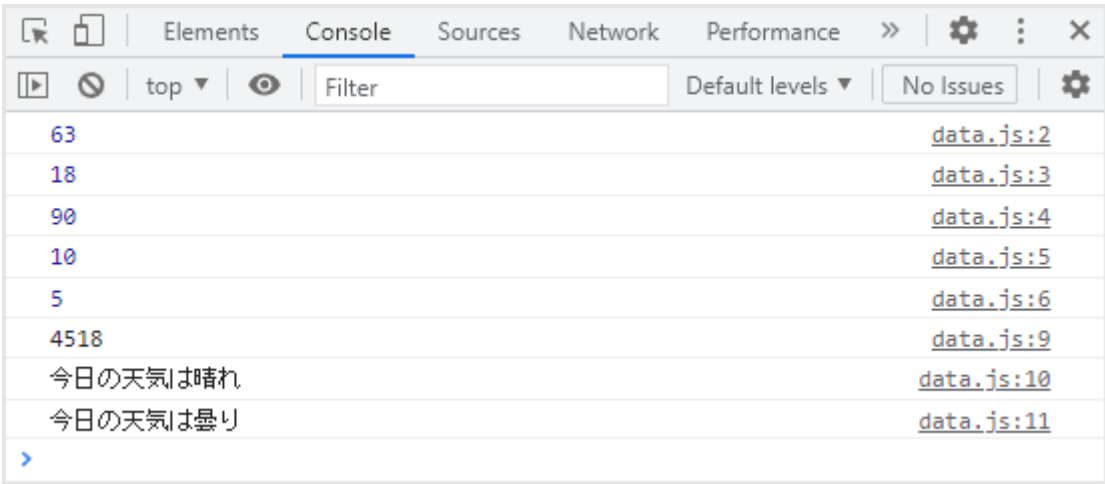
数値型に算術演算子 + を使った場合は足し算になりますが、文字列型に算術演算子 + を使えば文字列を連結することができます。

実際にやってみましょう。data.js を以下のように編集してください。

data.js

```
1 // 四則演算
2 console.log(45 + 18);
3 console.log(30 - 12);
4 console.log(15 * 6);
5 console.log(30 / 3);
6 console.log(45 % 8);
7
8 + // 文字列の連結
9 + console.log('45' + '18');
10 + console.log('今日の天気は' + '晴れ');
11 + console.log('今日の天気は' + '曇り');
12
```

index.html をブラウザで開き、デベロッパーツールのコンソールを確認してみましょう（すでに開いている場合は「F5」キーなどで再読み込み）。以下のように、文字列同士が連結されていればOKです。



お気づきかもしれませんが、 console.log(45 + 18); と console.log('45' + '18'); の実行結果が以下のように異なっていることがわかります。

- console.log(45 + 18); → 63
- console.log('45' + '18'); → 4518



この理由は前者が数値型で、後者が文字列型だからです（2節参照）。数値型の場合は足し算が行われ、文字列型の場合は連結が行われています。

同じ数字でも、`'`（シングルクォーテーション）で囲むかどうかによってデータ型が変わり、実行結果も変わることがわかりいただけたかと思います。

まとめ

本章では以下の内容を学習しました。

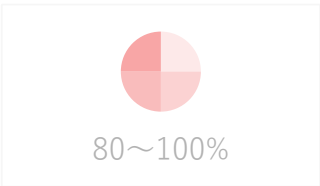
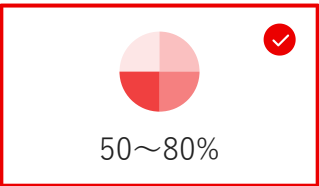
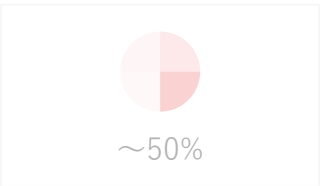
- JavaScriptにおける主なデータの種類（データ型）
 - 文字列型
 - 数値型
 - 論理型
- JavaScriptをはじめとするプログラミング言語では、コンソールにデータを出力することで実行結果を確認できる
- デベロッパーツールを使えばコンソールを表示できる
- 数値型に算術演算子を使えば四則演算できる
- 文字列型に算術演算子 `+` を使えば文字列を連結できる

算術演算子	処理の内容	例（出力結果）
<code>+</code>	左の値と右の値を 足す 。	<code>45 + 18</code> （ 63 ）
<code>-</code>	左の値から右の値を 引く 。	<code>30 - 12</code> （ 18 ）
<code>*</code>	左の値と右の値を 掛ける 。	<code>15 * 6</code> （ 90 ）
<code>/</code>	左の値を右の値で 割る 。	<code>30 / 3</code> （ 10 ）
<code>%</code>	左の値を右の値で割った 余り を求める。	<code>45 % 8</code> （ 5 ）

次章では、変数について学びます。

理解度を選択して次に進みましょう

ボタンを押していただくと次の章に進むことができます



最後に確認テストを行いましょう

下のボタンを押すとテストが始まります。

教材をみなおす

テストをはじめる

前に戻る

3 / 26 ページ

次に進む

◀ 一覧に戻る

! 改善点のご指摘、誤字脱字、その他ご要望はこちらからご連絡ください。