



15章 DOM操作の仕組みを理解しよう

DOMとは何かを解説します。

🕒 90分 🏆 - 📖 読了

15.1 本章の目標

本章では以下を目標にして学習します。

- DOMとは何か、概要をつかむこと
- DOM操作の仕組みを知ること

1章では、JavaScriptには「Webサイトに動きをつけられる」という特徴があることを学びました。「カルーセルを使ってメインビジュアルを切り替えたい」「ボタンが浮き出るようにアニメーションさせたい」など、Webサイトに動きをつけることを目的としてJavaScriptを学んでいる方も多いかと思います。

しかし、これまでに身につけてきた知識だけではWebサイトに動きをつけることはできません。Webサイトに動きをつけるには、本章で学ぶDOM操作の理解が必須です。

本章でDOM操作の仕組みを学ぶことで、JavaScriptを使ってHTML要素を取得したり、作成したりできるようになります。

本教材もあともう一息なので、頑張っていきましょう。

15.2 DOMとは

DOMはDocument Object Modelの略称で、**HTMLドキュメントをオブジェクトのように扱える仕組み**のことです。DOMを使えば、JavaScriptからHTML要素を取得・作成・更新・削除するといった操作ができるようになります。

例えば、AmazonのようなWebサービスにログインする場면을イメージしてみてください。パスワードの入力を間違えると以下のようなエラーメッセージが表示されるはずです。

+ 質問する

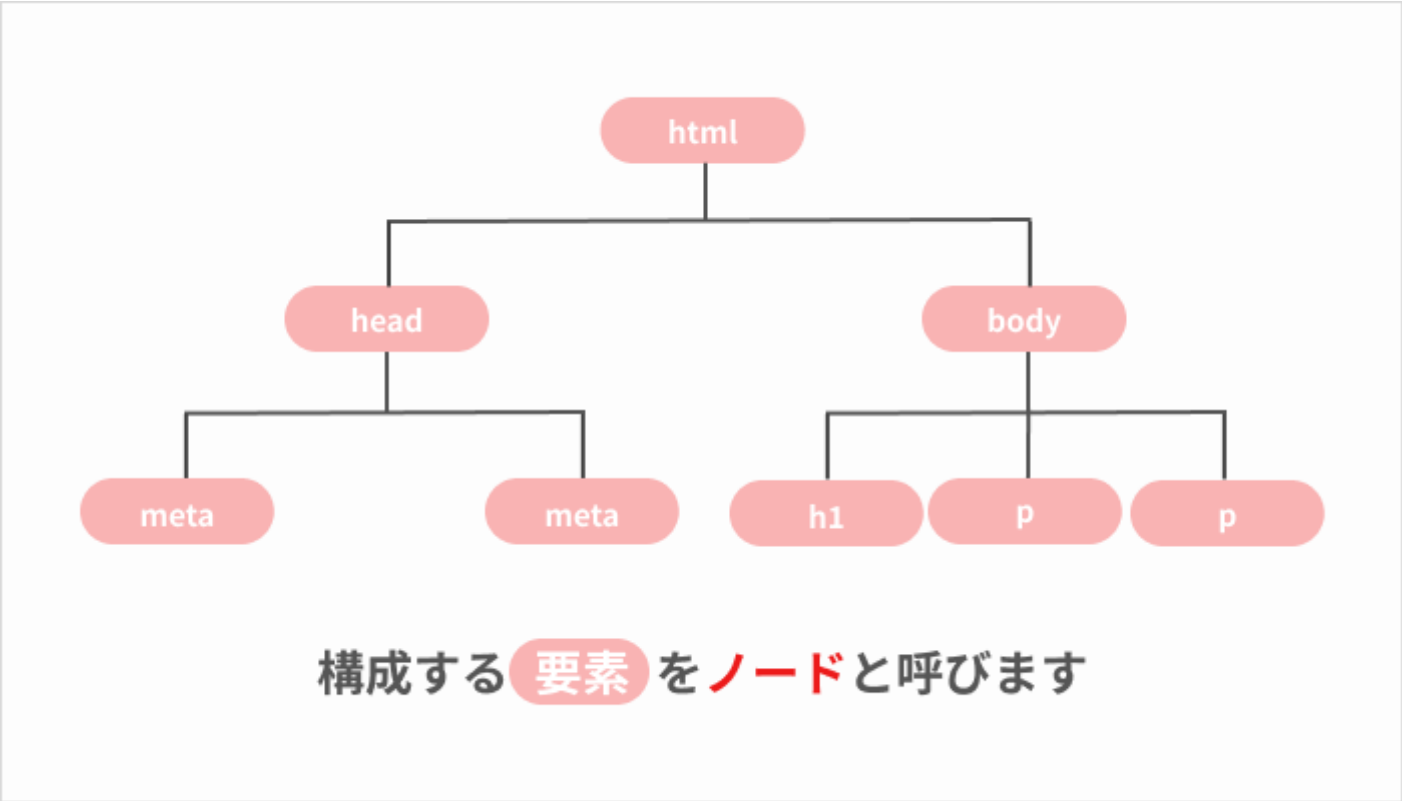


ここでは、JavaScriptからエラーメッセージ用のHTML要素を生成して画面に表示しています。これが**DOM操作**です。

DOMを使えば他にも、テキストの文字数をリアルタイムにカウントしたり、アニメーションを作成したり、スクロールにあわせて追尾するボタンを開発したりできるようになります（これらの機能を作る方法は16章で解説します）。

15.3 DOM操作の仕組み

ブラウザはHTMLファイルを読み込むと、自動的に以下のような**DOMツリー**を生成します。



まるで逆さになった木のように、階層が深くなるほどHTML要素が枝分かれしているのがわかりいただけると思います。

なお、DOMツリーを構成する各要素は**ノード**と呼ばれています。

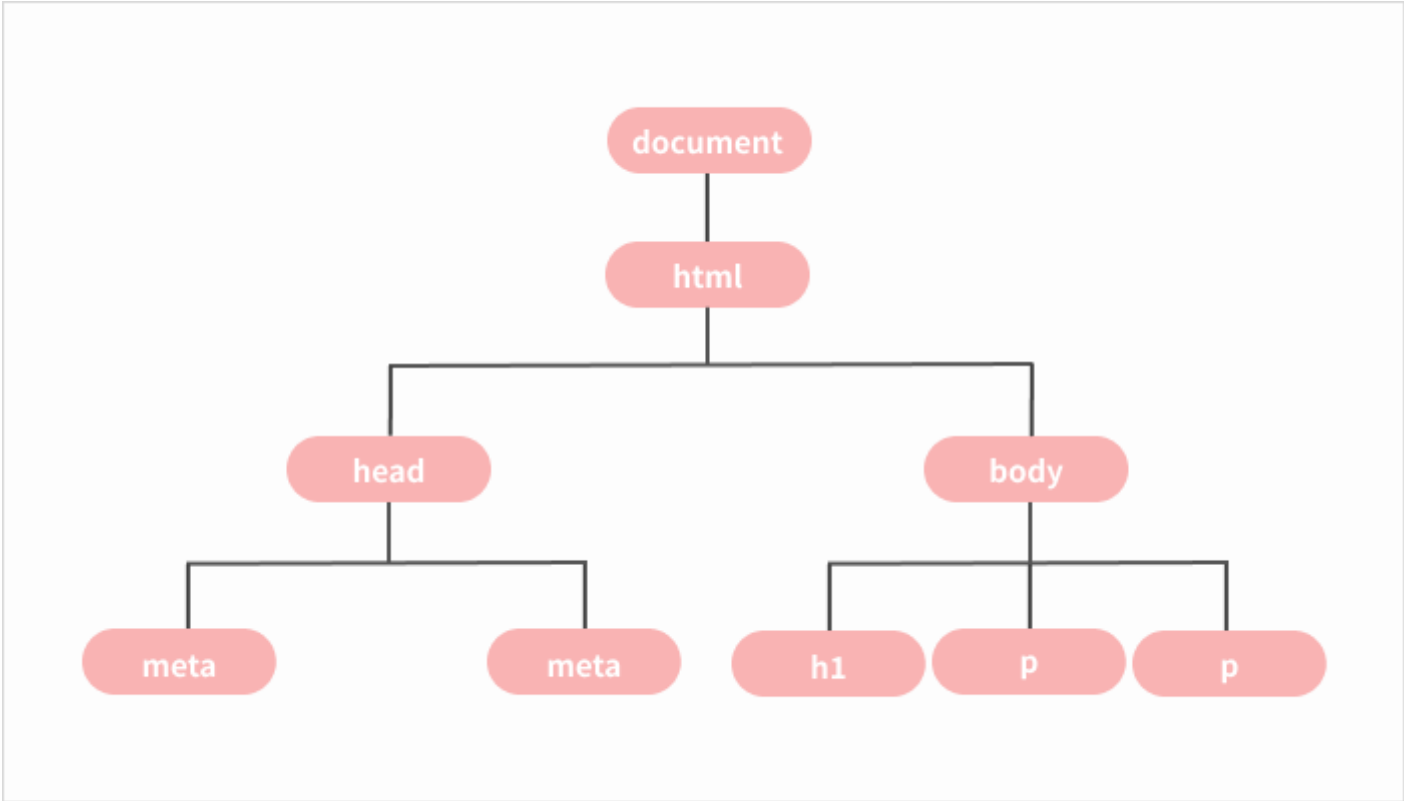
- ノード（node）＝結び目、こぶ

JavaScriptは各種メソッドを使ってこれらのノードにアクセスすることで、HTML要素の取得・作成・更新・削除などを行うことができます。これがDOM操作の仕組みです。



documentオブジェクトについて

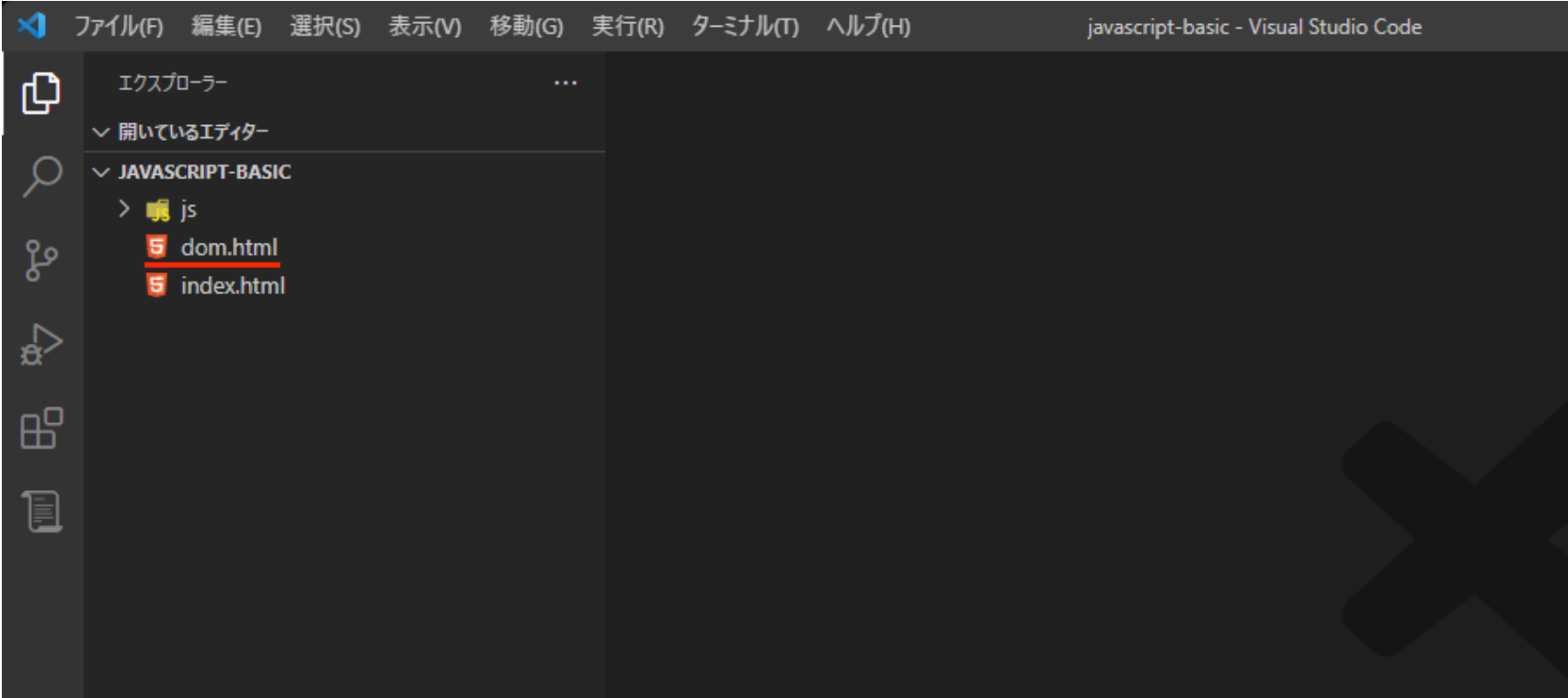
実は、DOMツリーの先頭には `document` という名前のオブジェクトがあります。DOMツリーが生成されるとこの `document` オブジェクトが使えるようになり、さまざまなDOM操作を行えるようになります。



DOM操作の仕組みをまとめると、以下のとおりです。

1. ブラウザがHTMLファイルを読み込むと、自動的にDOMツリーを生成する
2. DOMツリーが生成されると、 `document` オブジェクトが使えるようになる
3. `document` オブジェクトを使うと、JavaScriptでDOM操作ができる

では、実際に `document` オブジェクトを使ってみましょう。まずはVisual Studio Codeを開き、 `javascript-basic` フォルダ直下（`index.html` と同じ階層）に `dom.html` というファイルを作成してください。



続いて、 `dom.html` を以下のように編集しましょう。 `script` タグ内で読み込んでいる `dom.js` は、このあと作成します。

`dom.html`



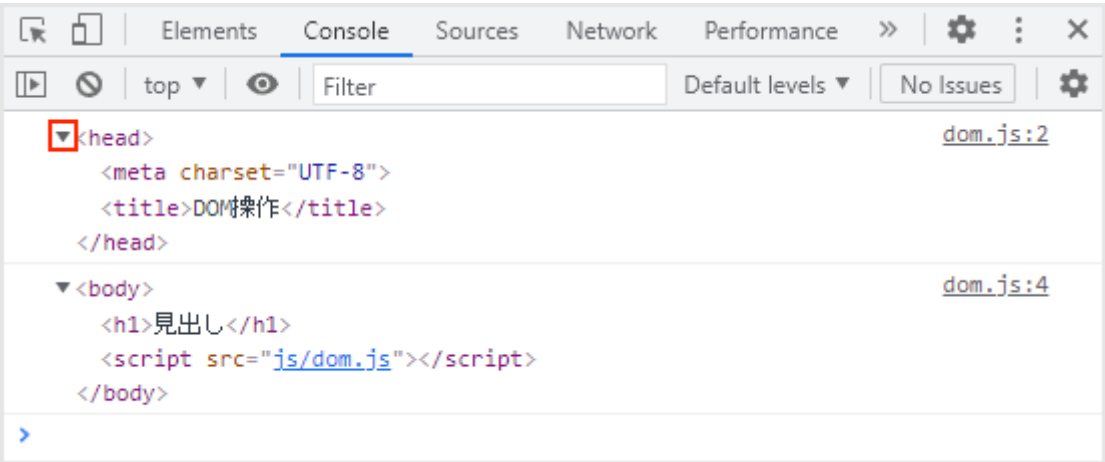
```
1 + <!DOCTYPE html>
2 + <html lang="ja">
3 +
4 + <head>
5 +   <meta charset="UTF-8">
6 +   <title>DOM操作</title>
7 + </head>
8 +
9 + <body>
10 +   <h1>見出し</h1>
11 +   <script src="js/dom.js"></script>
12 + </body>
13 +
14 + </html>
15
```

次に、js フォルダ内に新しく dom.js というファイルを作成してください。dom.js を作成したら、以下のように編集しましょう。以下のように document.HTML要素名 と記述することで、そのHTML要素の中身を取得できます。

dom.js

```
1 + // head要素を取得し、中身を出力する
2 + console.log(document.head);
3 + // body要素を取得し、中身を出力する
4 + console.log(document.body);
5
```

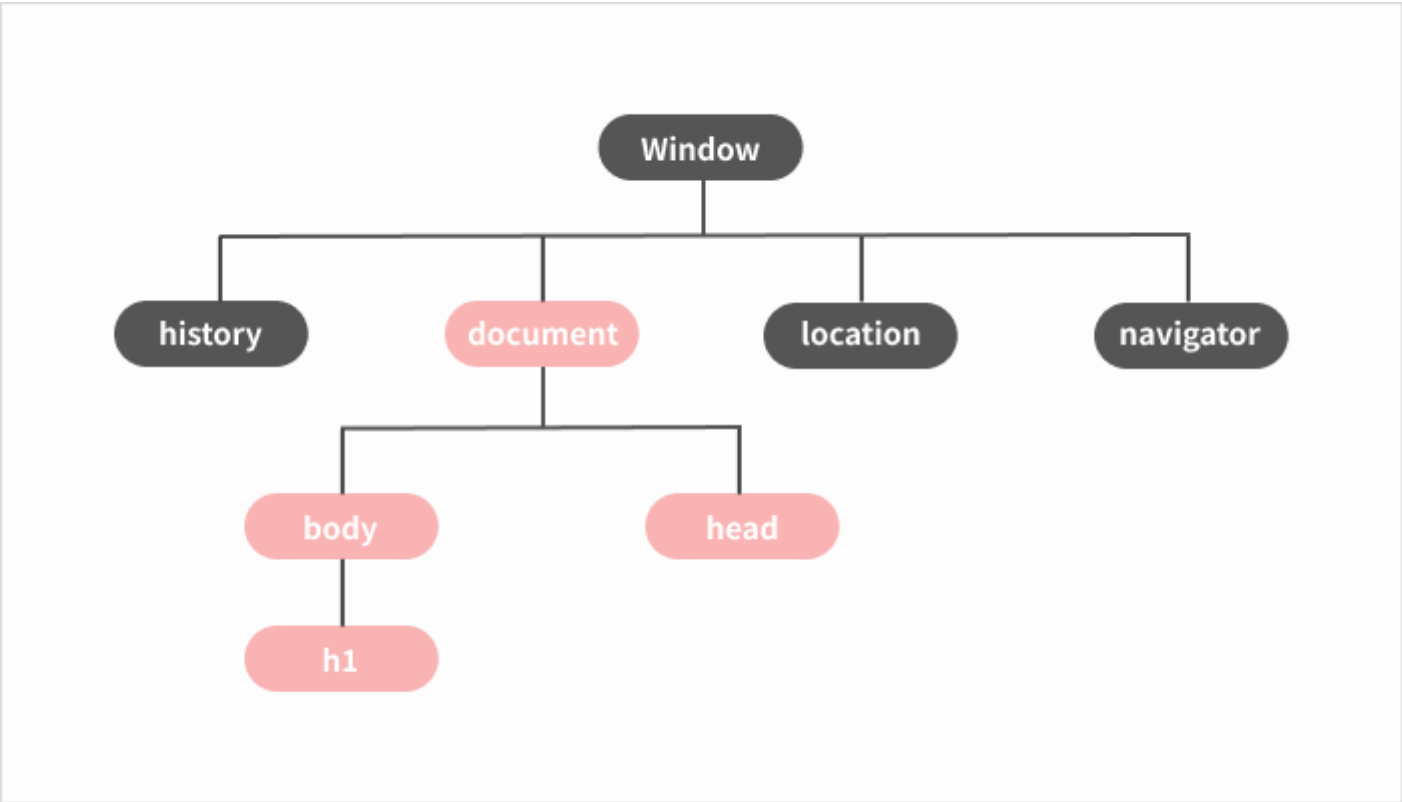
では dom.html をブラウザで開き、デベロッパーツールのコンソールを確認してみましょう。以下のように、 head 要素と body 要素の中身が出力されていればOKです（左側の三角ボタンをクリックすると中身を開閉できます）。



このように、 document オブジェクトを使うことで、JavaScriptからHTML要素にアクセスすることができます。そして、あらゆるHTML要素をJavaScriptから操作できるようになります。

補足 : windowオブジェクトについて

続いて、 document オブジェクトよりもさらに上の階層に存在する window オブジェクトについて補足しておきます。実は、先ほど使った document オブジェクトもこの window オブジェクトの中に存在しています。



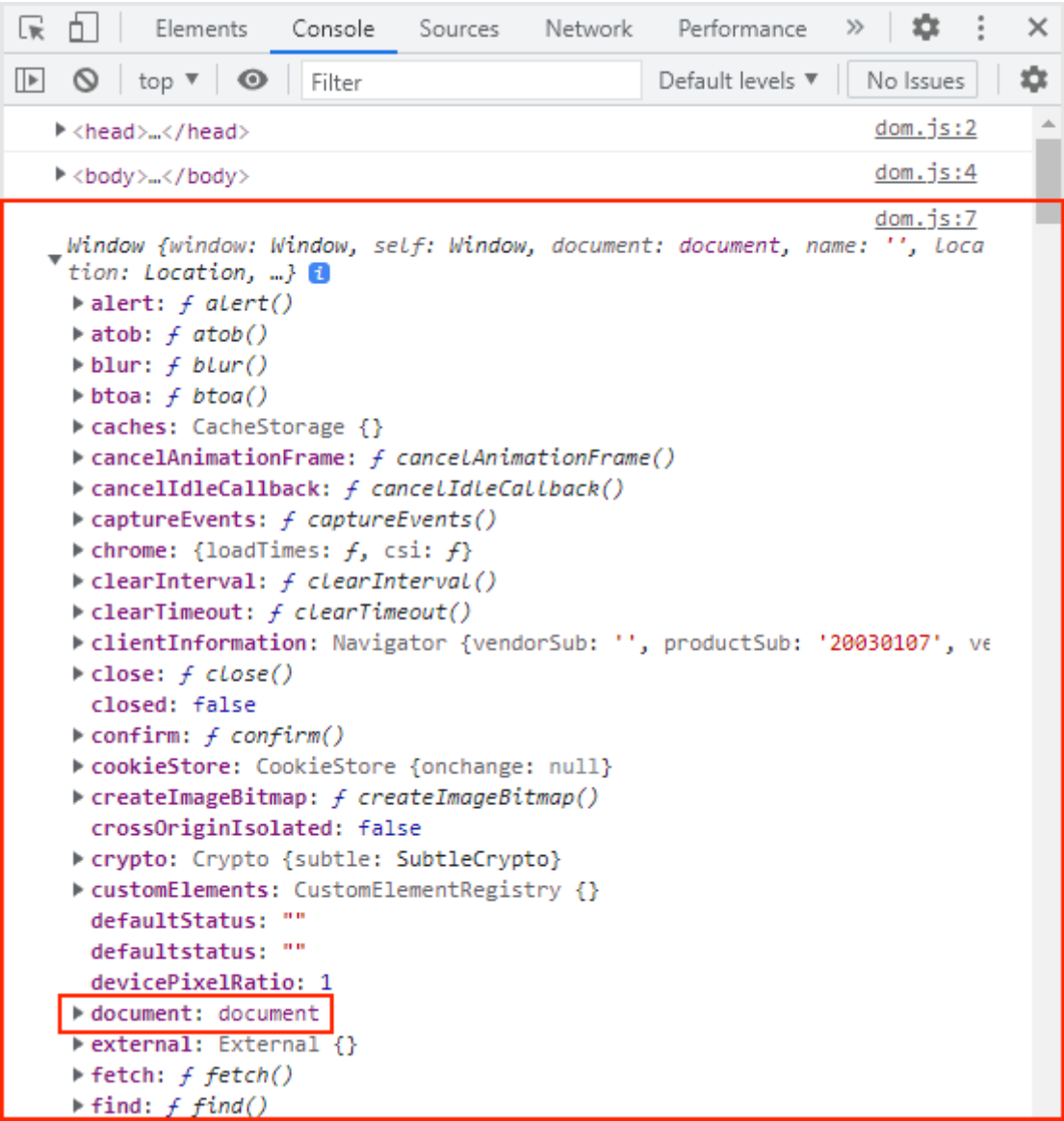
window オブジェクトとは、ブラウザが自動的に生成するオブジェクトのことです。 window オブジェクトを使うことで、JavaScriptからブラウザのさまざまな情報にアクセスできるようになります。

試しに dom.js を以下のように編集したあと、 dom.html をブラウザで開いて確認してみてください。

dom.js

```
1 // head要素を取得し、中身を出力する
2 console.log(document.head);
3 // body要素を取得し、中身を出力する
4 console.log(document.body);
5
6 + // windowオブジェクトの中身を出力する
7 + console.log(window);
8
```

以下のように、 window オブジェクトには大量のプロパティ（キーと値のセット）が存在することがわかりいただけると思います。
document オブジェクトの存在も確認できます。



なお、プロパティによっては値に `f` と表示されていますが、これは関数を意味するfunctionの頭文字です。つまり、メソッドが定義されています。

このように、 `document` オブジェクトは `window` オブジェクトの中に入っているため、以下2つのコードは同じ動作になります。

JSファイル（見本）

```
1 window.document.head;
2 document.head;
3
```

正確には `window.document.head` なのですが、 `window` の記述は省略できるため、 `document.head` と記述しても正しく実行できます。

実は、これまで何度も登場してきた `console.log()` メソッドも、 `window` オブジェクトの中に存在しています。よって、正確には `window.console.log()` となります。

このように、 `window` オブジェクトはJavaScriptが扱うオブジェクトの最上位に位置しており、中にはさまざまなオブジェクトやメソッドが入っています。これらのオブジェクトやメソッドを使うことで、さまざまな操作を行えるということを頭の片隅に入れておいてください。

まとめ

本章では以下の内容を学習しました。

- DOM（Document Object Model）とは、HTMLドキュメントをオブジェクトのように扱える仕組みのことである
- `document` オブジェクトを使うことで、さまざまなDOM操作を行える



次章では、実際にDOM操作を行います。



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

～50%

50～80%

80～100%

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

19 / 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。