

🏠

🕒

📖

📄

🔍

✎

🔊

教材

🔍 検索

所属チーム ▼ 🔔¹ 👤

本文 目次 質問一覧 3件

ホーム 教材 JavaScriptの基礎を学ぼう スコープを理解しよう

13章 スコープを理解しよう

変数の扱い方のひとつであるスコープについて解説します。

🕒90分 🏆 - 📖 読了

13.1 本章の目標

本章では以下を目標にして学習します。

■ スコープについて知り、関数の内外で変数や定数を正しく使えるようになること

関数の中で変数や定数を使うときは、**使える範囲（有効範囲）** に注意が必要です。変数や定数の有効範囲を知っておかないと、中身が入っていないかったり想定とは異なる中身が入っていたりして、予期せぬエラーにつながってしまいます。

本章では変数や定数の有効範囲について学び、関数の内外で変数や定数を正しく使えるようになります。

13.2 スコープとは

スコープとは、**変数や定数を使う範囲（有効範囲）** のことです。言い換えれば、変数や定数の値を取得したり、変更したりできる範囲のことです。

JSファイル

const userName = '侍太郎'

const useVariable = () => {

定数userNameが使えるスコープ

}

定数userNameが使えるスコープ

JSファイル

const useVariable = () => {

const userName = '侍太郎'

定数userNameが使えるスコープ

}

定数userNameが使えないスコープ

このスコープを知らなければ、変数や定数が使えずにコードが正しく動作しなくなるおそれがあります。

+ 質問する

https://terakoya.sejuku.net/programs/60/chapters/679

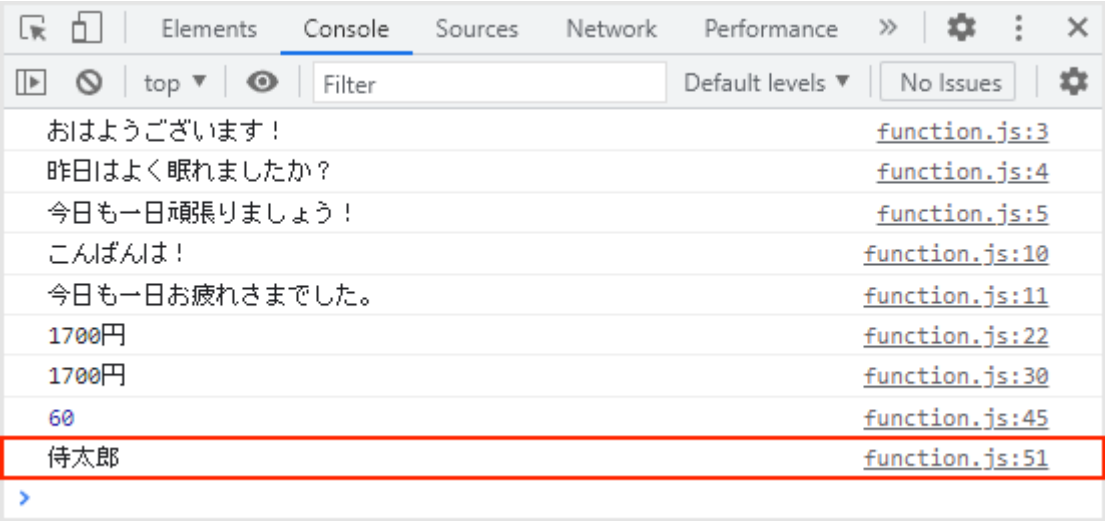
1/5

ではscopeとはどのようなものかを理解するために、実際にやってみましょう。まずは `function.js` を以下のように編集してください。

function.js

```
1 //===== 前略 =====
2
3 // 与えられた引数numを2倍にし、その値を戻り値として返す関数を定義する
4 const double = (num) => {
5   return num * 2;
6 }
7
8 // 関数の戻り値を出力する
9 console.log(double(30));
10
11 + // 定数を宣言する
12 + const userName = '侍太郎';
13 +
14 + // 定数の値を出力する
15 + console.log(userName);
16
```

続いて `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。以下のように、定数の値が問題なく表示されます。つまり、定数 `userName` を問題なく使えているということです。



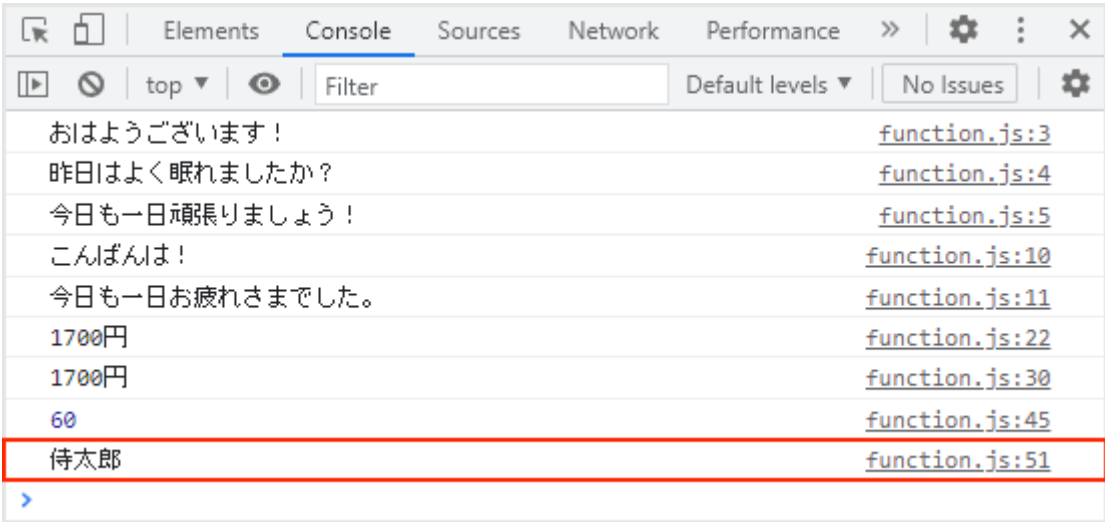
では次に、関数の中で定数 `userName` を使ってみましょう。 `function.js` を以下のように編集してください。

function.js

```
1 //===== 前略 =====
2
3 // 定数を宣言する
4 const userName = '侍太郎'
5
6 - // 定数の値を出力する
7 - console.log(userName);
8 + // 関数の中で定数を使う
9 + const useVariable = () => {
10 +   console.log(userName);
11 + }
12 +
13 + // 関数を呼び出す
14 + useVariable();
15
```

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。関数の中で定数を使った場合でも、以下のよう

に問題なく表示されます。

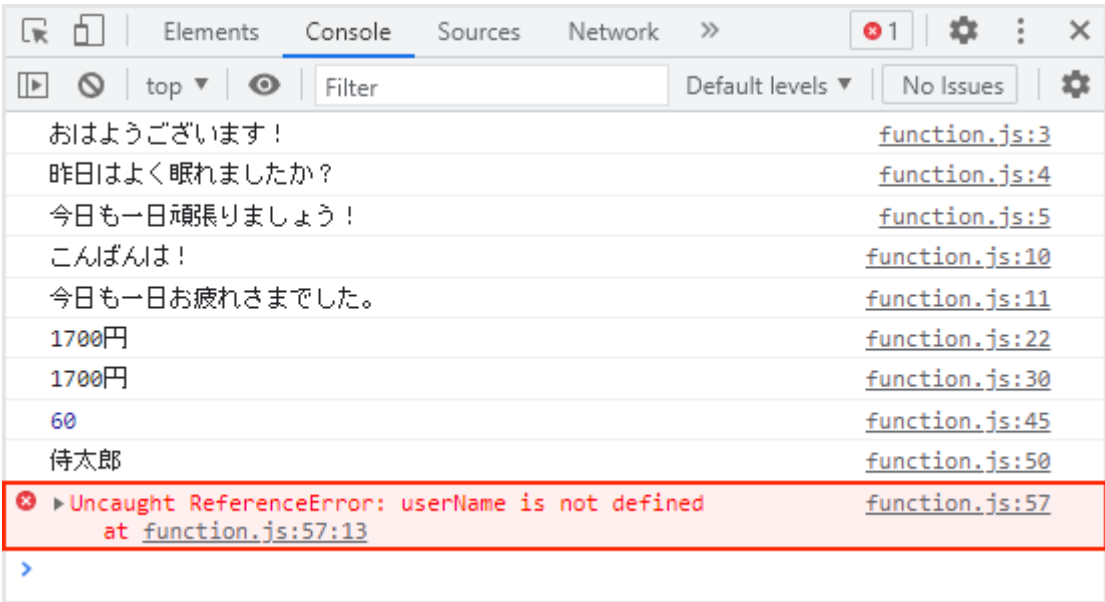


では逆に、関数の中で宣言した定数を関数の外で使った場合はどうなるか試してみましょう。function.js を以下のように編集してください。

function.js

```
1 //===== 前略 =====
2
3 - // 定数を宣言する
4 - const userName = '侍太郎'
5
6 - // 関数の中で定数を使う
7 + // 関数の中で定数を宣言し、関数の中で定数を使う
8 const useVariable = () => {
9 +   const userName = '侍太郎'
10   console.log(userName);
11 }
12
13 // 関数を呼び出す
14 useVariable();
15
16 + // 関数の中で宣言した定数を関数の外で使う
17 + console.log(userName);
18
```

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。関数の中で宣言した定数を関数の外で使った場合、以下のようにエラーが発生してしまいます。



このように、変数や定数は使える範囲（有効範囲）が決まっています。これがスコープです。

JSファイル

```
const userName = '侍太郎'
const useVariable = () => {
  定数userNameが使えるスコープ
}
定数userNameが使えるスコープ
```

JSファイル

```
const useVariable = () => {
  const userName = '侍太郎'
  定数userNameが使えるスコープ
}
定数userNameが使えないスコープ
```

なお、関数以外にもif文やwhile文など、 `{ }` を使う構文であれば同じようにスコープが決まります。

JSファイル

```
if (条件式) {
  const userName = '侍太郎';
  定数userNameが使えるスコープ
}
定数userNameが使えないスコープ
```

JSファイル

```
while (条件式) {
  const userName = '侍太郎';
  定数userNameが使えるスコープ
}
定数userNameが使えないスコープ
```

まとめ

本章では以下の内容を学習しました。

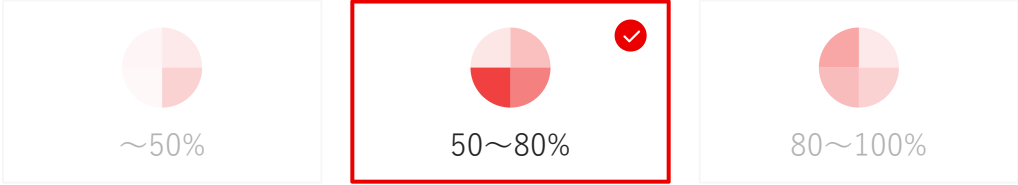
- スコープとは、 **変数や定数**が**使える範囲（有効範囲）**のことである
- スコープ外で変数や定数を使った場合、エラーが発生する

次章では、クラスについて学びます。



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

16 / 26 ページ

次に進む

く 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。