

教材

検索

所属チーム ▼ 🔔<sup>1</sup> 👤

本文

目次

質問一覧 7件

ホーム

教材

JavaScriptの基礎を学ぼう

関数を理解しよう

11章 関数を理解しよう

関数でたくさんの処理をひとまとめにしたり再利用します。

🕒90分 🏆 - 📖読了

11.1 本章の目標

本章では以下を目標にして学習します。

- 関数とは何か、概要をつかむこと
- 関数の作り方・呼び出し方を知ること
- 関数を実際に使ってみること

例えば以下のように、朝のあいさつを出力する一連の処理があるとします（Webサービスのログイン時にあいさつが表示される場면을イメージしましょう）。

```
1 console.log('おはようございます!');
2 console.log('昨日はよく眠れましたか?');
3 console.log('今日も一日頑張りましょう!');
4
```

プログラミングをしていると、このような一連の処理を何度も実行したくなる場面が必ずやってきます。しかし、その度に毎回同じ処理を記述するのは大変ですし、コードが無駄に長くなってしまいます。

```
1 console.log('おはようございます!');
2 console.log('昨日はよく眠れましたか?');
3 console.log('今日も一日頑張りましょう!');
4
5 console.log('おはようございます!');
6 console.log('昨日はよく眠れましたか?');
7 console.log('今日も一日頑張りましょう!');
8
```

そして何より問題なのは、処理の一部を変更したくなったときにすべてのコードを書き換えなければならないことです。例えば上記の例であれば、朝のあいさつを丁寧体から「おはよう!」「昨日はよく眠れた?」などの普通体に変更したいとき、console.log() 内のすべての文字列を書き換えなければなりません。そこで大活躍するのが**関数**です。関数を使えば以下のように、一度関数を作るだけで何度でも再利用できるようになります（コードの書き方は本章で詳しく解説します）。

+ 質問する

https://terakoya.sejuku.net/programs/60/chapters/677

1/8



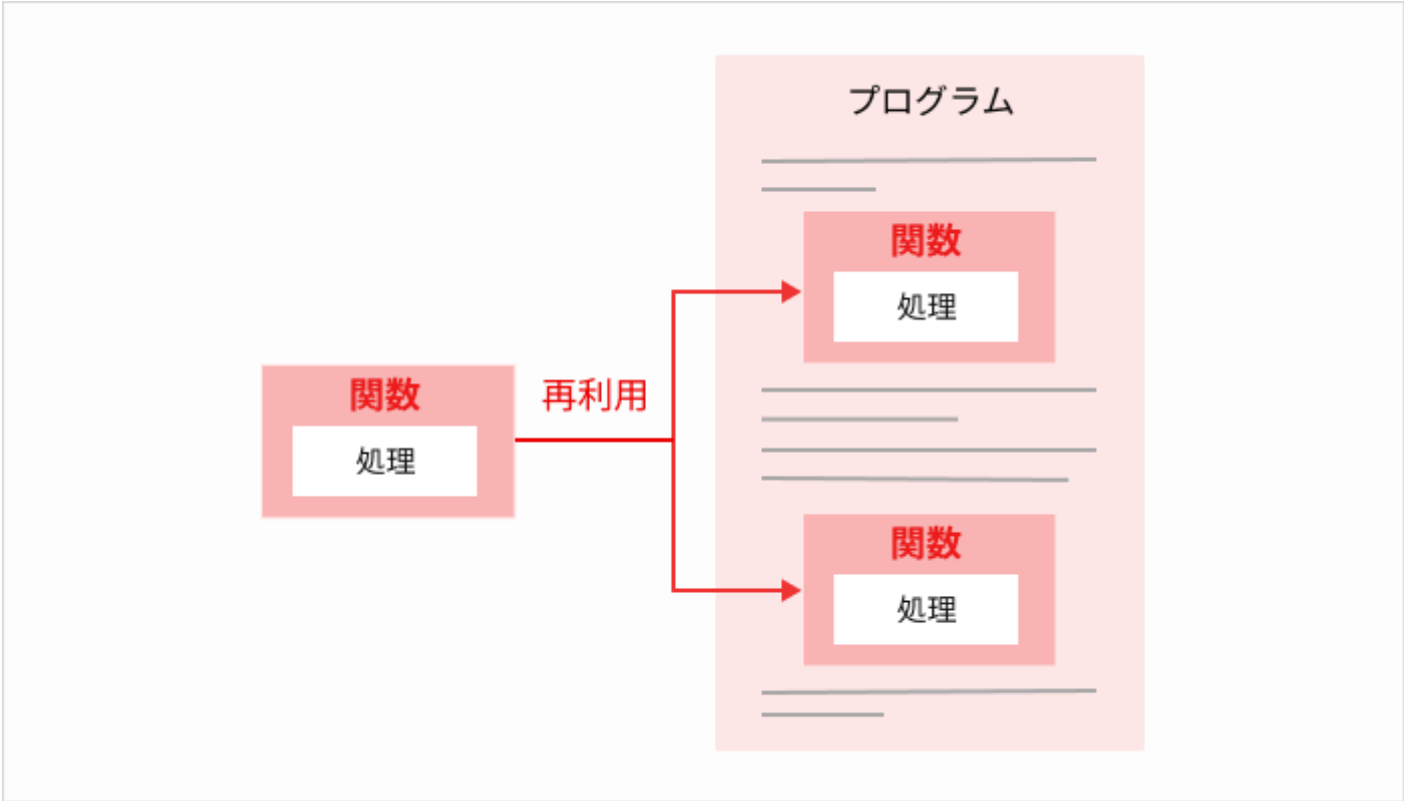
```
1 // 朝のあいさつを出力する関数を作成する
2 const sayGoodMorning = () => {
3   console.log('おはようございます!');
4   console.log('昨日はよく眠れましたか?');
5   console.log('今日も一日頑張りましょう!');
6 }
7
8 // 関数を呼び出す (1回目)
9 sayGoodMorning();
10
11 // 関数を呼び出す (2回目)
12 sayGoodMorning();
13
```

出力内容を変えたいときも、一度コードを書き換えれば済みます。

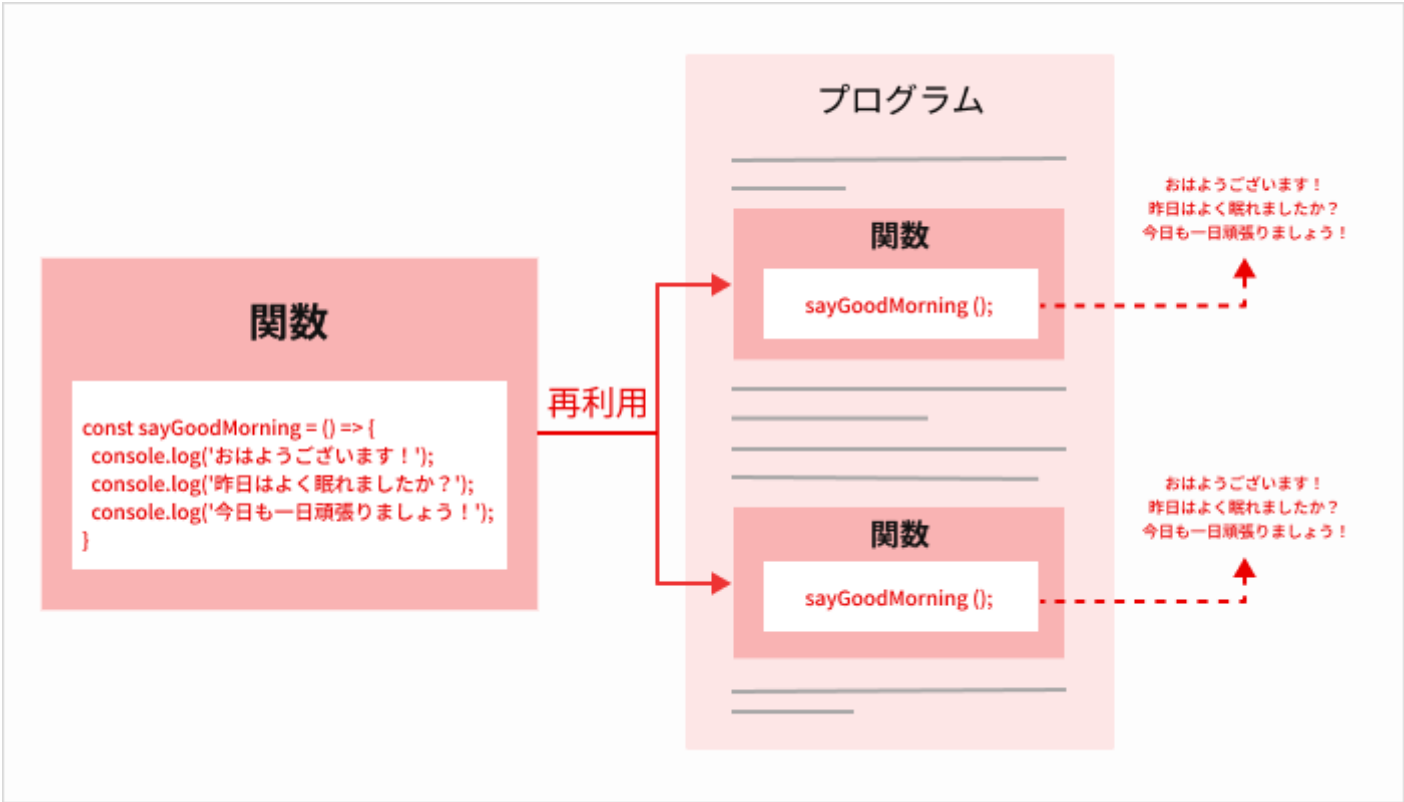
関数を使いこなせればより楽にコードを書けるようになるので、本章で一緒に使いながら感覚をつかみましょう。

## 11.2 関数とは

関数とは、一連の処理をひとまとめにして、何度でも再利用できるようにする仕組みのことです。



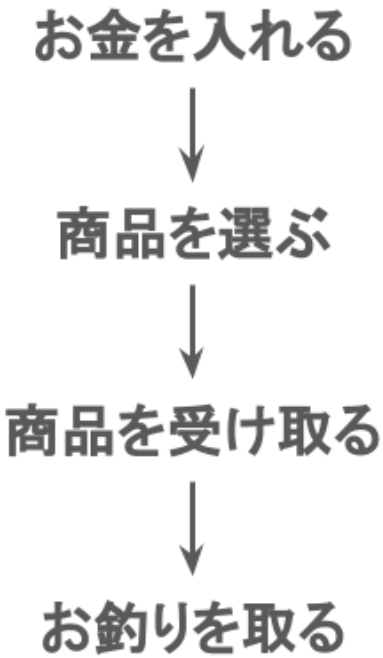
1節の例の場合は以下ようになります。



関数を使えば短いコードで再利用できるので、コード全体がスッキリして見やすくなります。関数の基本的な考え方を理解するために、まずは自動販売機を例に見ていきましょう。

## 自動販売機の例

一般的に、自動販売機を使う流れは以下のとおりです。



お金を入れ、商品を選んで受け取り、お釣りがあれば排出するというシンプルな処理です。この流れは、基本的に誰が利用しても同じです。

そこで、これらの処理を関数として1つにまとめ、再利用しやすい形にしておきます。そうしておく、例えばカップラーメンの自動販売機を作ることになっても、関数をそのまま再利用すれば簡単に開発ができるわけです。

## ショッピングサイトにおける商品検索の例

🔴

🏠

🕒

📖

📄

🔍

✎

🗣️

Webの世界においても同じようなケースはたくさんあります。例えばAmazonや楽天のようなショッピングサイトで商品を検索するシーンをイメージしてください。

一般的に、検索を利用する流れは以下のとおりです。



キーワードを入力

↓

データベースにアクセス

↓

該当商品を検索

↓

結果を表示する

検索したい商品のキーワードを入力すると、データベースにアクセスして結果を表示してくれます。この流れも、基本的に誰が利用しても同じです。

そこで、この一連の処理を関数として1つにまとめておけば、例えば新しくレストランを検索する機能が欲しいときにも、関数を再利用して素早く開発できるようになります。

関数のメリットをまとめると以下ようになります。

- 複雑なコードを1つにまとめられる
- 同じ処理を行うときに何度でも再利用できる
- プログラミングの生産性を高め、素早い開発ができるようになる

### 11.3 関数の作り方

関数を作るには以下のように記述します。実は、JavaScriptにおける関数は文字列や数値と同じくデータ型の一種なので、関数自体を変数や定数に代入できます（関数を再代入するケースはほとんどないため、一般的には定数に代入します）。

```
1  const 定数名 = () => {
2    一連の処理
3  }
4
```

なお、これは**アロー関数**と呼ばれる書き方です。アロー関数という名称は、丸括弧 ( ) の直後にある矢印 => が由来です（arrow=矢印）。他にも関数の作り方はありますが（このあと簡単に解説します）、現在ではアロー関数が主流なので基本的にはこちらを使いましょう。

https://terakoya.sejuku.net/programs/60/chapters/677

4/8





本章の冒頭で挙げた朝のあいさつの例であれば、以下のように記述します。

JSファイル（見本）

```
1  const sayGoodMorning = () => {
2    console.log('おはようございます!');
3    console.log('昨日はよく眠れましたか?');
4    console.log('今日も一日頑張りましょう!');
5  }
6
```

なお、JavaScriptをはじめとするプログラミング言語では、関数を作成することを「関数を定義する」というので頭の片隅に入れておいてください。

## 補足：アロー関数以外の記法（関数宣言）

アロー関数以外にも、「関数宣言」と呼ばれる記法があります。関数宣言は以下のように、 `function` に続けて任意の関数名を記述する方法です。

```
1  function 関数名() {
2    一連の処理
3  }
4
```

本章の冒頭で挙げた朝のあいさつの例であれば、以下のように記述します。

JSファイル（見本）

```
1  function sayGoodMorning() {
2    console.log('おはようございます!');
3    console.log('昨日はよく眠れましたか?');
4    console.log('今日も一日頑張りましょう!');
5  }
6
```

アロー関数はJavaScriptのバージョンES6で新しく追加された記法ですが、関数宣言はそれ以前から存在していました。そのため、古いバージョンのJavaScriptでよく目にします。

繰り返しになりますが、基本的にはアロー関数を使えばOKです。ただし、上記のような関数宣言もまだ現役で使われているので、頭の片隅に入れておきましょう。

## 関数名のつけ方

関数名は変数と同じように、内容がわかる名前にすべきです。

基本的に、プログラミング言語で関数名をつけるときは、「 `addProduct` （商品を追加する）」のように「動詞＋目的語」の形をとることが多いです。そうすることで、関数が実行する処理の内容がわかりやすくなります。

以下に代表的なパターンを掲載するので、イメージをつかんでおきましょう。



処理の内容	関数名	例
〇〇を追加する	add〇〇	addProduct （商品を追加する）
〇〇を削除する	remove〇〇	removeProduct （商品を削除する）
〇〇が存在するかどうかチェックする	has〇〇	hasProduct （商品が存在するかチェックする）
〇〇の状態になっているかどうかチェックする	is〇〇	isPurchased （購入されたかどうかチェックする）
〇〇が起こったときに処理を行う	on〇〇	onPushedButton （ボタンが押されたときに処理を行う）

## 11.4 関数の呼び出し方

関数を呼び出す（実行する）方法は簡単で、関数名を記述するだけです。

実際にコードを書いてみましょう。まずはVisual Studio Codeを開き、 `js` フォルダ内に新しく `function.js` というファイルを作成してください（`function`=関数）。

続いて、 `function.js` を以下のように編集しましょう。まずは2つの関数を定義し、そのあと関数名を記述して関数を呼び出します。

function.js

```
1 + // 朝のあいさつを出力する関数を定義する
2 + const sayGoodMorning = () => {
3 +   console.log('おはようございます！');
4 +   console.log('昨日はよく眠れましたか？');
5 +   console.log('今日も一日頑張りましょう！');
6 + }
7 +
8 + // 夜のあいさつを出力する関数を定義する
9 + const sayGoodEvening = () => {
10 +   console.log('こんばんは！');
11 +   console.log('今日も一日お疲れさまでした。');
12 + }
13 +
14 + // 朝のあいさつを出力する関数を呼び出す
15 + sayGoodMorning();
16 +
17 + // 夜のあいさつを出力する関数を呼び出す
18 + sayGoodEvening();
19
```

次に `index.html` を以下のように編集し、読み込むJSファイルを `function.js` に変更してください。

index.html

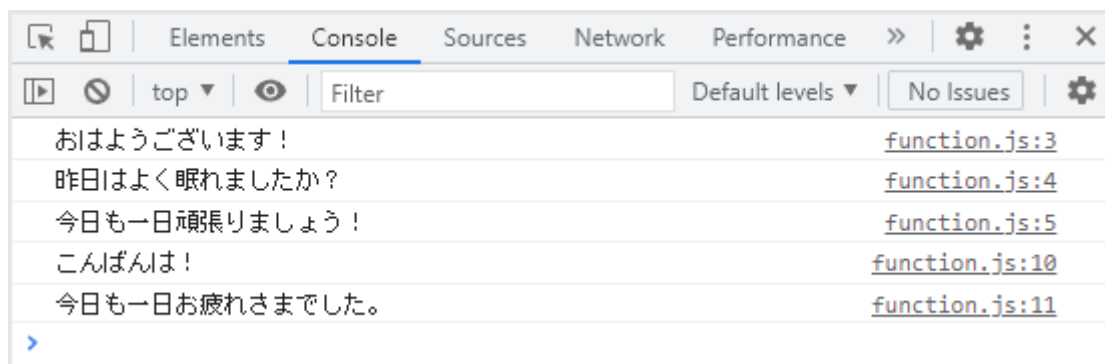


```

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/object.js"></script>
11 +     <script src="js/function.js"></script>
12 </body>
13
14 </html>
15

```

では 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、実行された結果が出力されていればOKです。



なお、「関数は定義しただけでは実行されない」という点に注意してください。関数は呼び出して初めて実行されます。実際に上記の例でも、呼び出した回数分しか実行されていません。


## まとめ








本章では以下の内容を学習しました。

- 関数とは、一連の処理をひとまとめにして、何度でも再利用できるようにする仕組みのことである
- 関数を呼び出す（実行する）には、関数名を記述するだけでよい（例：`sayGoodMorning();`）
- 関数は呼び出して初めて実行される

```
1 // 関数の作り方
2 const 定数名 = () => {
3     一連の処理
4 }
5
```

次章では、引数・戻り値・スコープについて学びます。





## 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

～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

14 / 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。

© SAMURAI Inc.

[利用規約](#)

[法人会員利用規約](#)

[プライバシーポリシー](#)

[運営会社](#)

>

https://terakoya.sejuku.net/programs/60/chapters/677

8/8