



7章 条件分岐のswitch文を理解しよう

条件分岐はifだけではなくswitchでもできます。本章でその使い方の違いを学びましょう。

🕒 120分 🏆 - 📖 読了

7.1 本章の目標

本章では以下を目標にして学習します。

- switch文の書き方を知り、実際にコードを書いてみる

本章では前章に引き続き、条件分岐について学んでいきましょう。前章ではif文の書き方を学びましたが、本章ではswitch文の書き方を学びます。

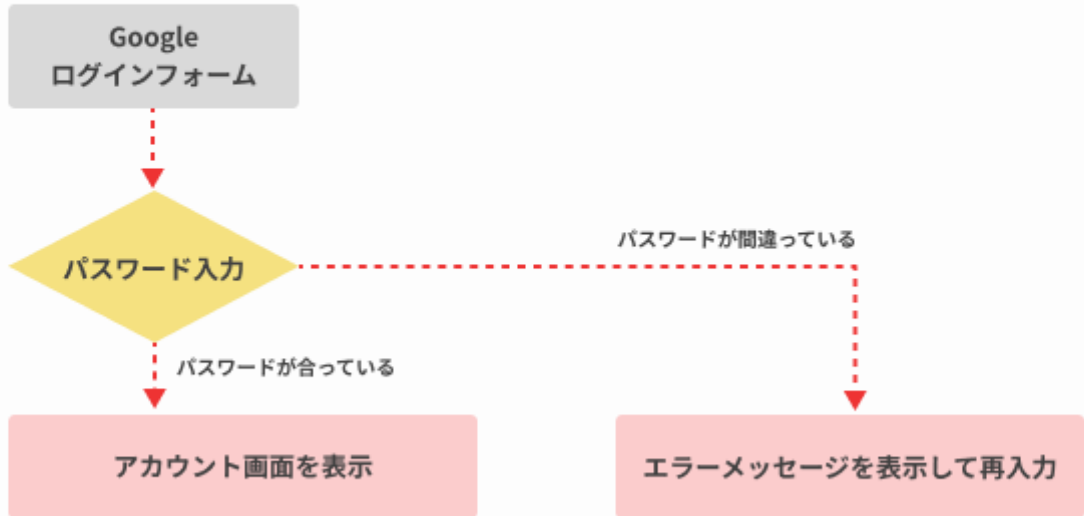
7.2 条件分岐とif文について復習しよう

まずは、前章で学んだ条件分岐とif文について復習しましょう。

条件分岐

条件分岐とは、「必須事項が入力されていたらフォーム内容を送信し、未入力であればエラーメッセージを表示する」など、**条件によって処理を分ける**ことでした。

条件分岐の例

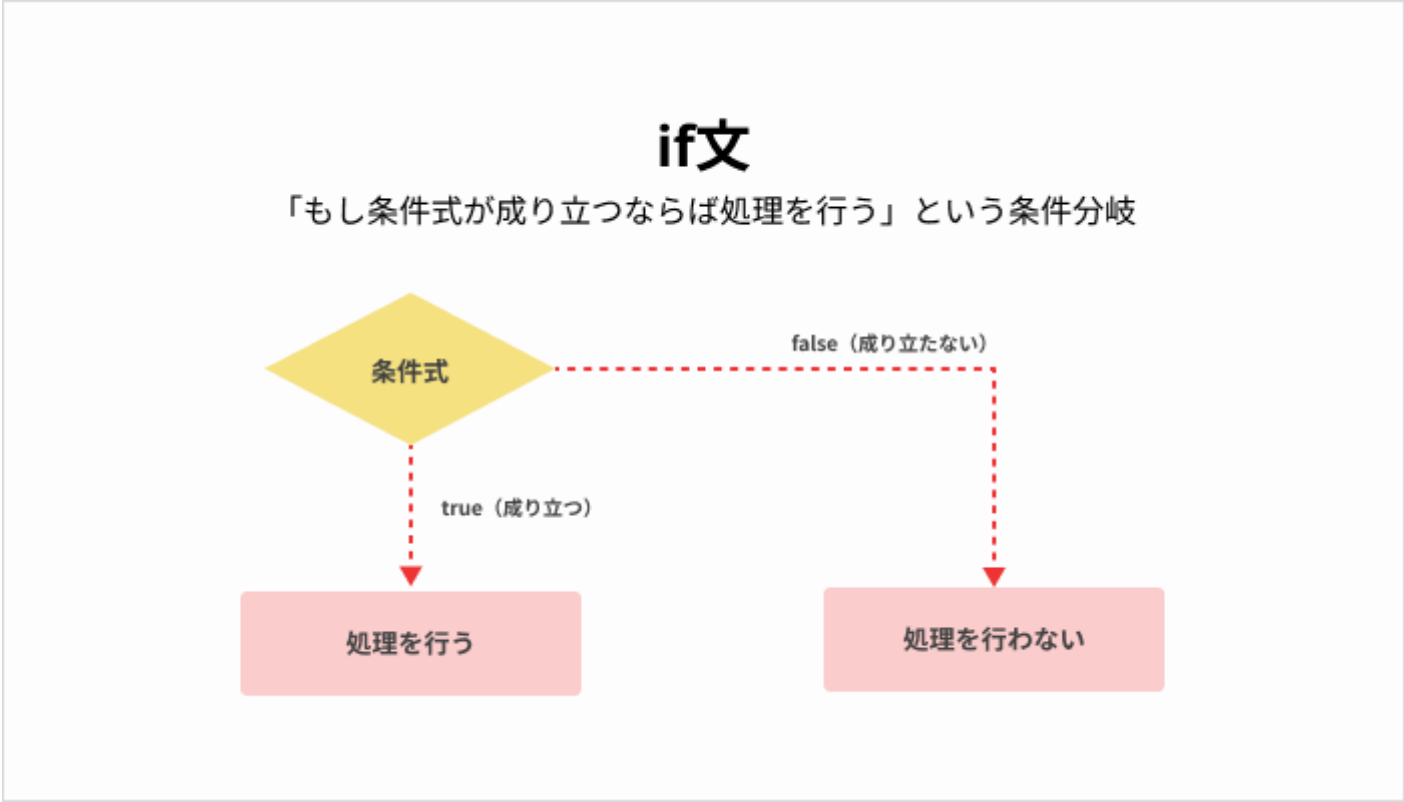


+ 質問する



if文

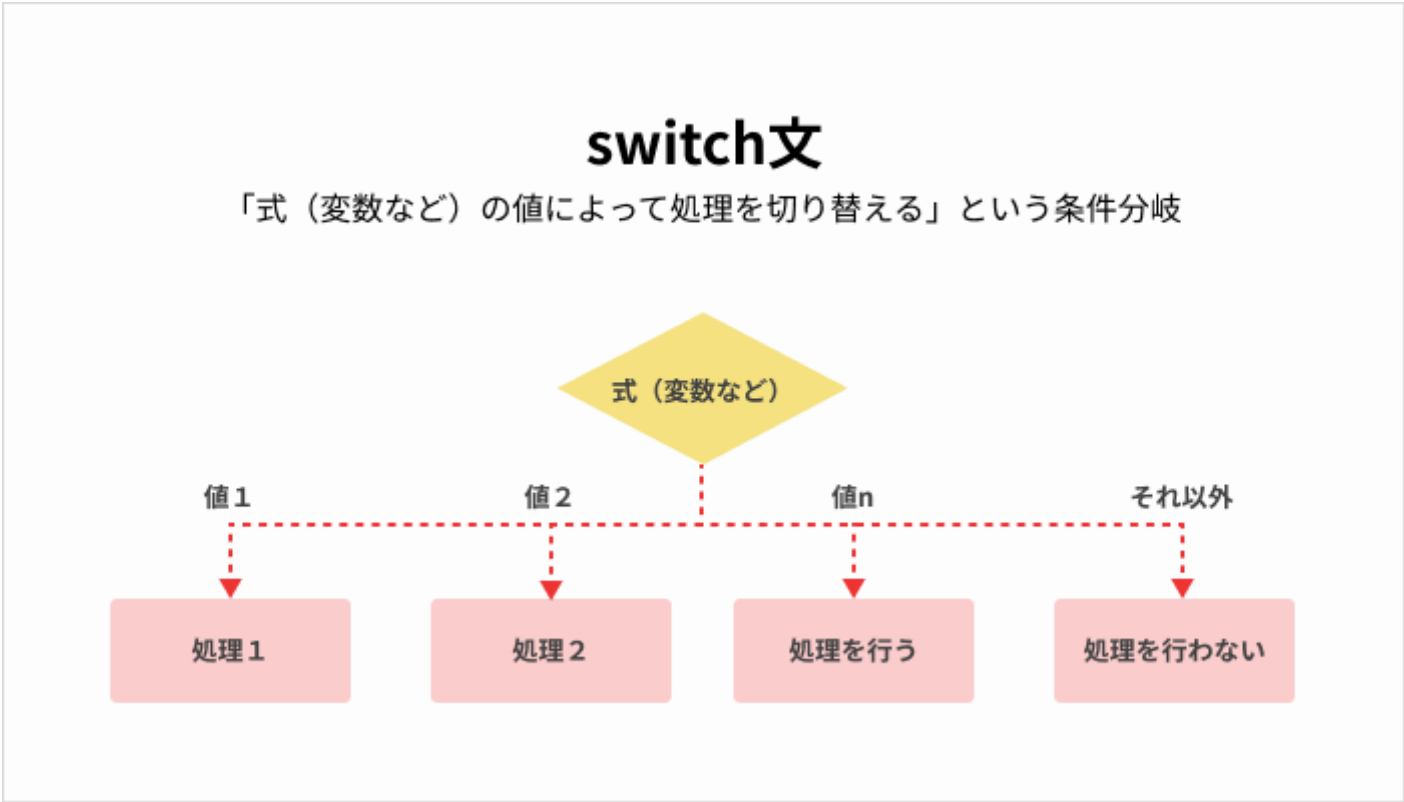
if文とは、「もし条件式が成り立つならば、処理を行う」という条件分岐のことでした。



if文の書き方は以下のとおりです。

```
1  if (条件式A) {
2    条件Aが成り立つときの処理
3  }
4  else if (条件式B) {
5    条件Bが成り立つときの処理
6  }
7  else {
8    どの条件も成り立たないときの処理
9  }
10
```

7.3 switch文の書き方





switch文は「式（変数など）の値によって処理を切り替える」という条件分岐です。式（変数など）が3つ以上の値をとり、値によって処理を切り替えたいときによく使われます。

例えば映画館において、曜日によって入場料の割引対象を変えたいときなどです。

- 月曜日であれば、成人男性の入場料を割引く
- 金曜日であれば、成人女性の入場料を割引く
- 日曜日であれば、子供の入場料を割引く

if文でも同じ条件分岐を実現できるのですが、このようにパターン数が多い場合はswitch文を使うことでより簡潔に記述できます。

switch文は以下のように書きます。5節で実際にコードを書くので、現時点ではイメージだけつかめればOKです。

```
1 switch (式) {  
2   case 値1:  
3     式が値1のときの処理  
4     break;  
5   case 値2:  
6     式が値2のときの処理  
7     break;  
8   case 値3:  
9     式が値3のときの処理  
10    break;  
11 }  
12
```

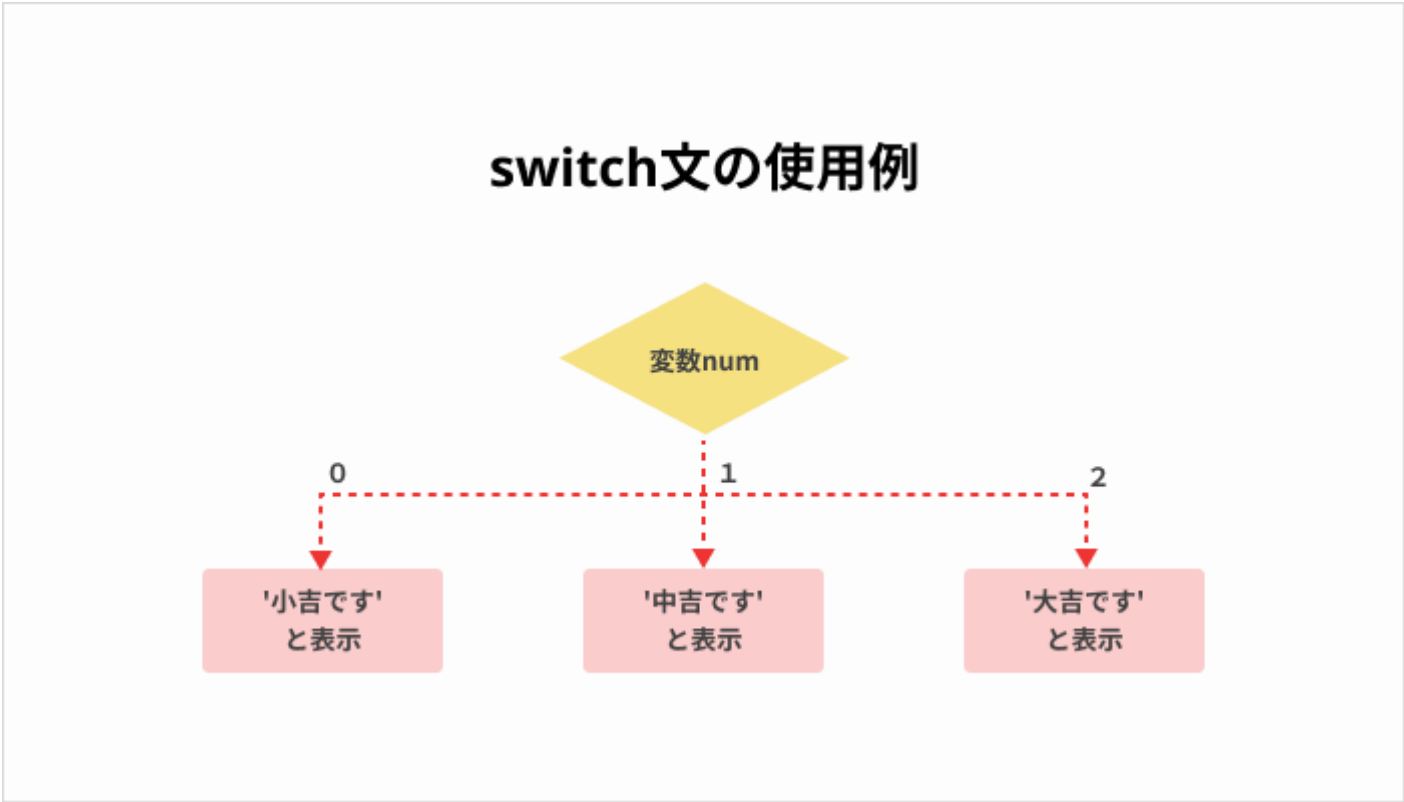
丸括弧 () の中に式（変数など）を記述し、波括弧 { } の中に分けたい処理の数だけ case を記述します。caseは「場合」という意味で、case のあとに条件となる値を記述します。値の直後にはコロン : が必要なので、忘れないようにしましょう。

また、break; はswitch文を終了するためのコードです（※ break の直後は**セミコロン ;**）。break; を記述しないと、最初に値が一致した case 以降の処理がすべて実行されてしまいます。case と break; をセットで記述する習慣をつけておきましょう。

break; を記述しない場合にどうなるかは4節の最後に確かめます。

switch文の使用例

switch文の使用例は以下のとおりです。以下の例では、変数 num に代入された値によって出力される文字列が切り替わります。





JSファイル（見本）

```
1 // 変数numに0〜2までのランダムな整数を代入する
2 let num = Math.floor(Math.random() * 3);
3
4 // 変数numの値によって出力する文字列を切り替える
5 switch (num) {
6   case 0:
7     console.log('小吉です')
8     break;
9   case 1:
10    console.log('中吉です')
11    break;
12   case 2:
13    console.log('大吉です')
14    break;
15 }
16
```

前章でも登場しましたが、`Math.floor(Math.random() * n)` は、`0 ~ n - 1` までのランダムな整数を取得するコードです（覚える必要はありません）。

7.4 switch文を書いてみよう

では、実際にswitch文を書いてみましょう。以下の2パターンに分けて、簡単なプログラムを作成します（`default` については後述します）。

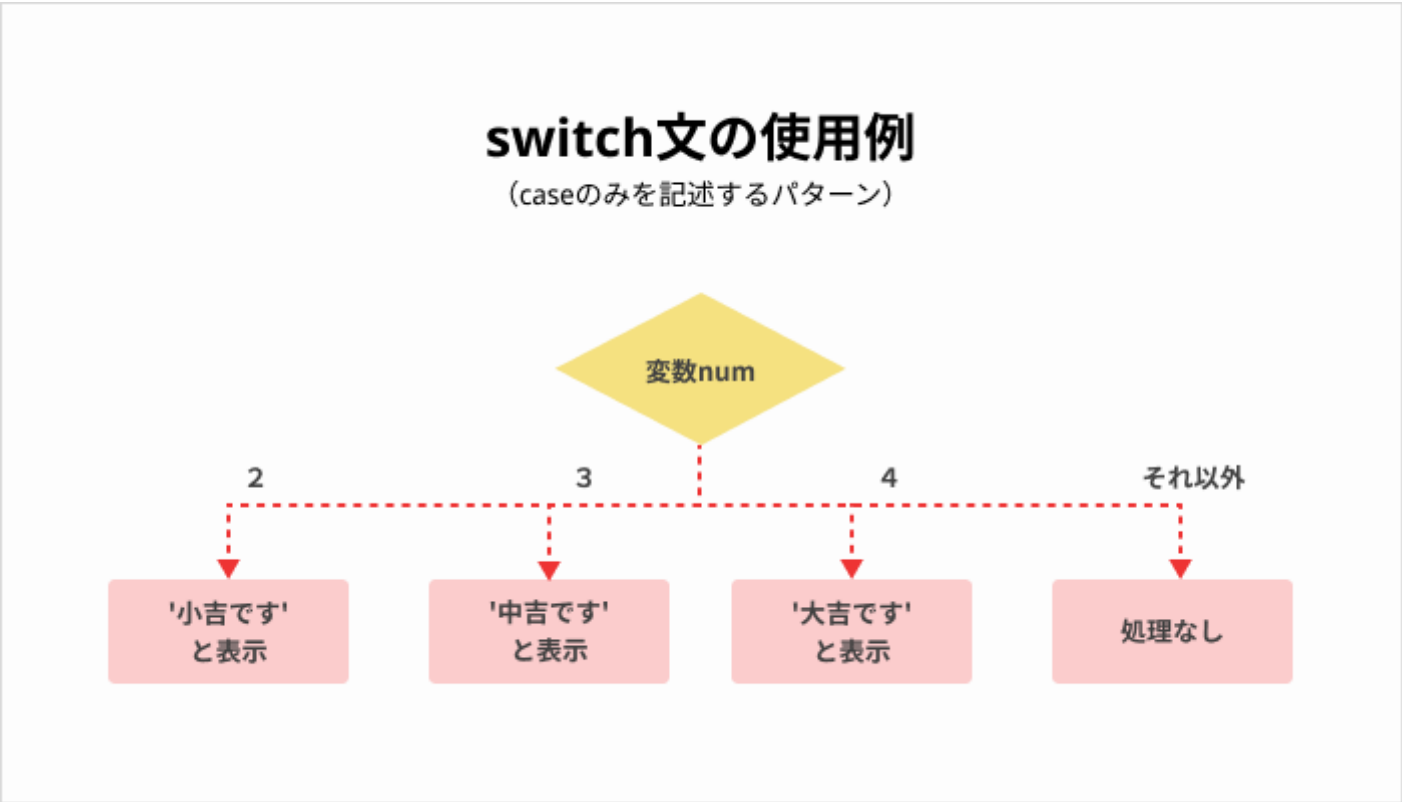
- 1. `case` のみを記述するパターン（値が〇〇のときは●●し、□□のときは■■する）
- 2. `case` と `default` を記述するパターン（値が〇〇のときは●●し、□□のときは■■し、それ以外のときは▲▲する）

順番にやっていきましょう。

1. caseのみを記述するパターン（値が〇〇のときは●●し、□□のときは■■する）

まずは以下のようなプログラムを作成します。

- 1. 変数 `num` に `0 ~ 4` までのランダムな整数を代入する
- 2. 変数 `num` の値によって、出力する文字列を切り替える
 - 1. `2` のとき：「小吉です」
 - 2. `3` のとき：「中吉です」
 - 3. `4` のとき：「大吉です」



では、 `if-switch.js` を以下のように編集してください。

`if-switch.js`

```
1 //===== 前略 =====
2
3 // 変数numに0〜4までのランダムな整数を代入する
4 let num = Math.floor(Math.random() * 5);
5
6 // 変数numの値を出力する（確認用）
7 console.log(num);
8
9 // 変数numの値が4であれば、「大当たりです」という文字列を出力する
10 if (num === 4) {
11   console.log('大当たりです');
12 }
13 // 変数numの値が3であれば、「当たりです」という文字列を出力する
14 else if (num === 3) {
15   console.log('当たりです');
16 }
17 // それ以外のときは、「はずれです」という文字列を出力する
18 else {
19   console.log('はずれです');
20 }
21
22 + // 変数numに0〜4までのランダムな整数を代入する
23 + num = Math.floor(Math.random() * 5);
24 +
25 + // 変数numの値を出力する（確認用）
26 + console.log(num);
27 +
28 + // 変数numの値によって、出力する文字列を切り替える
29 + switch (num) {
30 +   case 2:
31 +     console.log('小吉です');
32 +     break;
33 +   case 3:
34 +     console.log('中吉です');
35 +     break;
36 +   case 4:
37 +     console.log('大吉です');
38 +     break;
39 + }
40
```



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。変数 num の値によって、以下のように表示される文字列が切り替わればOKです。

- 2 のとき：「 小吉です 」
- 3 のとき：「 中吉です 」
- 4 のとき：「 大吉です 」
- それ以外のとき：何も表示されない（処理なし）

2のとき

	Elements	Console	Sources	Network	Performance	»	⚙	⋮	✕
		top ▾	Filter	Default levels ▾	No Issues		⚙		
63									if-switch.js:2
true									if-switch.js:5
true									if-switch.js:8
false									if-switch.js:11
2									if-switch.js:17
はずれです									if-switch.js:29
2									if-switch.js:36
小吉です									if-switch.js:41
>									

3のとき

	Elements	Console	Sources	Network	Performance	»	⚙	⋮	✕
		top ▾	Filter	Default levels ▾	No Issues		⚙		
63									if-switch.js:2
true									if-switch.js:5
true									if-switch.js:8
false									if-switch.js:11
4									if-switch.js:17
当たります									if-switch.js:21
3									if-switch.js:36
中吉です									if-switch.js:44
>									

4のとき

	Elements	Console	Sources	Network	Performance	»	⚙	⋮	✕
		top ▾	Filter	Default levels ▾	No Issues		⚙		
63									if-switch.js:2
true									if-switch.js:5
true									if-switch.js:8
false									if-switch.js:11
2									if-switch.js:17
はずれです									if-switch.js:29
4									if-switch.js:36
大吉です									if-switch.js:47
>									

それ以外（0、1）のとき

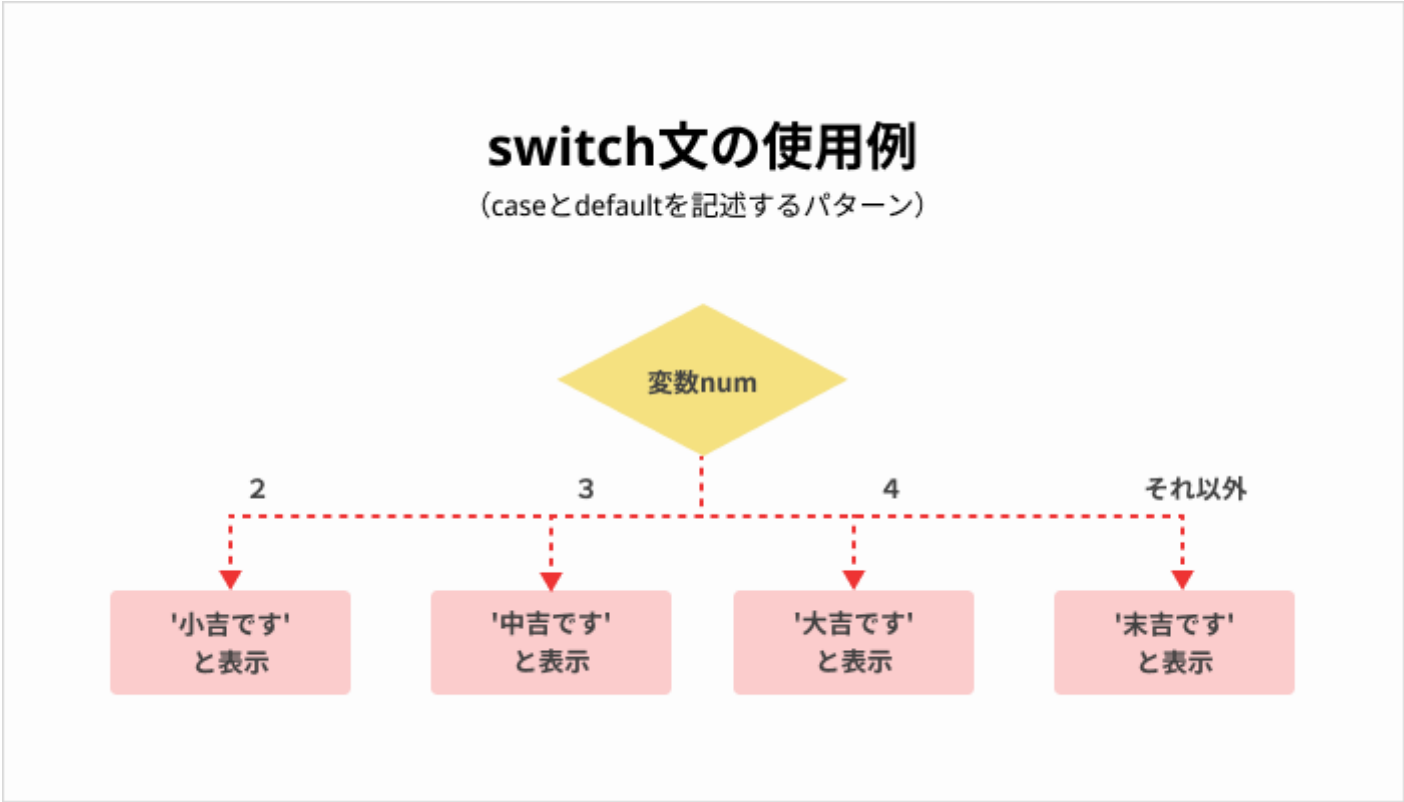


	Elements	Console	Sources	Network	Performance	>>	⚙️	⋮	✕
▶	🔍	top ▾	👁	Filter	Default levels ▾	No Issues	⚙️		
63									if-switch.js:2
true									if-switch.js:5
true									if-switch.js:8
false									if-switch.js:11
3									if-switch.js:17
当たりです									if-switch.js:25
0									if-switch.js:36
>									

2. caseとdefaultを記述するパターン（値が〇〇のときは●●し、□□のときは■し、それ以外のときは▲▲する）

続いて、以下のようなプログラムを作成します。

- 変数 `num` に 0 ～ 4 までのランダムな整数を代入する
- 変数 `num` の値によって、出力する文字列を切り替える
 - 2 のとき：「小吉です」
 - 3 のとき：「中吉です」
 - 4 のとき：「大吉です」
 - それ以外のとき：「末吉です」



注目していただきたいのは、変数 `num` の値が「それ以外のとき」、つまりいずれの `case` にも当てはまらないときの処理です。

`if`文においては `else` を記述しましたが、`switch`文においては `default` を記述することで、いずれの `case` にも当てはまらないときに処理を行うことができます。



```
1 switch (式) {
2   case 値1:
3     式が値1のときの処理
4     break;
5   case 値2:
6     式が値2のときの処理
7     break;
8   case 値3:
9     式が値3のときの処理
10    break;
11  default:
12    式がいずれのcaseにも当てはまらないときの処理
13    break;
14 }
15
```

では実際にやってみましょう。 if-switch.js を以下のように編集してください。

if-switch.js

```
1 //===== 前略 =====
2
3 // 変数numに0～4までのランダムな整数を代入する
4 num = Math.floor(Math.random() * 5);
5
6 // 変数numの値を出力する（確認用）
7 console.log(num);
8
9 // 変数numの値によって、出力する文字列を切り替える
10 switch (num) {
11   case 2:
12     console.log('小吉です');
13     break;
14   case 3:
15     console.log('中吉です');
16     break;
17   case 4:
18     console.log('大吉です');
19     break;
20 +   default:
21 +     console.log('末吉です');
22 +     break;
23 }
24
```

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。変数 num の値によって、以下のように表示される文字列が切り替わればOKです。

- 2 のとき：「 小吉です 」
- 3 のとき：「 中吉です 」
- 4 のとき：「 大吉です 」
- それ以外のとき：「 末吉です 」

2のとき



63	if-switch.js:2
true	if-switch.js:5
true	if-switch.js:8
false	if-switch.js:11
2	if-switch.js:17
はずれです	if-switch.js:29
2	if-switch.js:36
小吉です	if-switch.js:41
>	

3のとき

63	if-switch.js:2
true	if-switch.js:5
true	if-switch.js:8
false	if-switch.js:11
4	if-switch.js:17
大当たりです	if-switch.js:21
3	if-switch.js:36
中吉です	if-switch.js:44
>	

4のとき

63	if-switch.js:2
true	if-switch.js:5
true	if-switch.js:8
false	if-switch.js:11
2	if-switch.js:17
はずれです	if-switch.js:29
4	if-switch.js:36
大吉です	if-switch.js:47
>	

それ以外（0、1）のとき

63	if-switch.js:2
true	if-switch.js:5
true	if-switch.js:8
false	if-switch.js:11
4	if-switch.js:17
大当たりです	if-switch.js:21
1	if-switch.js:36
末吉です	if-switch.js:50
>	

break;を記述しなければどうなるか確かめてみよう

3節でswitch文の書き方を学んだときに、以下のように解説しました。





なお、`break;` はswitch文を終了するためのコードです。 `break;` を記述しないと、最初に値が一致した `case` 以降の処理がすべて実行されてしまいます。 `case` ごとに `break;` を記述する習慣をつけておきましょう。

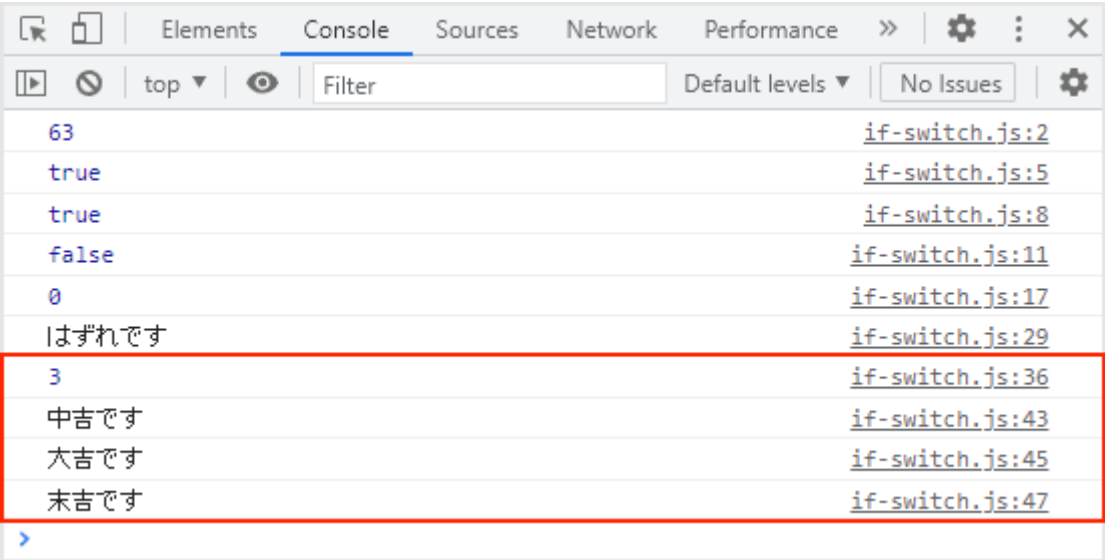
では、`break;` を記述しなければどうなるか実際に確かめてみましょう。 `if-switch.js` を以下のように編集し、 `break;` をコメントアウトしてください。

if-switch.js

```
1 //===== 前略 =====
2
3 // 変数numに0~4までのランダムな整数を代入する
4 num = Math.floor(Math.random() * 5);
5
6 // 変数numの値を出力する（確認用）
7 console.log(num);
8
9 // 変数numの値によって、出力する文字列を切り替える
10 switch (num) {
11   case 2:
12     console.log('小吉です');
13 -   break;
14 +   // break;
15   case 3:
16     console.log('中吉です');
17 -   break;
18 +   // break;
19   case 4:
20     console.log('大吉です');
21 -   break;
22 +   // break;
23   default:
24     console.log('末吉です');
25 -   break;
26 +   // break;
27 }
28
```

`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、最初に値が一致した `case` 以降の処理がすべて実行されてしまいます。

以下の例であれば変数 `num` の値が `3` なので、 `case 3` 以降のすべて（ `case 3` 、 `case 4` 、 `default` ）が実行されてしまっています。



意図せぬ動作を防ぐためにも、switch文を使うときは `break;` を記述する習慣をつけておきましょう。コメントアウトした `break;` はもとに戻しておいてください。

if-switch.js





```
1 //===== 前略 =====
2
3 // 変数numに0〜4までのランダムな整数を代入する
4 num = Math.floor(Math.random() * 5);
5
6 // 変数numの値を出力する（確認用）
7 console.log(num);
8
9 // 変数numの値によって、出力する文字列を切り替える
10 switch (num) {
11     case 2:
12         console.log('小吉です');
13         // break;
14         break;
15     case 3:
16         console.log('中吉です');
17         // break;
18         break;
19     case 4:
20         console.log('大吉です');
21         // break;
22         break;
23     default:
24         console.log('末吉です');
25         // break;
26         break;
27 }
28
```

本章の学習は以上です。お疲れさまでした。

まとめ

本章では以下の内容を学習しました。

- switch文は「式（変数など）の値によって処理を切り替える」という条件分岐である
- 変数が3つ以上の値をとり、値によって処理を切り替えたいときにswitch文がよく使われる
- break; はswitch文を終了するためのコードで、 case ごとに記述する必要がある
- break; を記述しない場合、最初に値が一致した case 以降の処理がすべて実行されてしまう

```
1 switch (式) {
2     case 値1:
3         式が値1のときの処理
4         break;
5     case 値2:
6         式が値2のときの処理
7         break;
8     case 値3:
9         式が値3のときの処理
10        break;
11    default:
12        式がいずれのcaseにも当てはまらないときの処理
13        break;
14 }
15
```

次章では、繰り返し処理について学びます。





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

8 / 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。