

if文

ifとは

if文とは`if() {}`の`()`の中に比較する条件を書き、その条件が`true`の場合は`{}`の中の処理に入っていく、`false`の場合は`{}`の中の処理に入らないような条件分岐を作って処理を行う時に使用します。

条件は自由に設定ができますが、その条件に対して`true`か`false`で判断します。

プログラミングをする上でif文は必ず使います。

使用例

今から書くコードを実際にコンソールで出力させながら確認して見てください。

```
var boolean = true;
if (boolean) {
  console.log(boolean);
};
```

`var boolean = true;`で`true`を代入しました。

`if(boolean)`で`boolean`が`true`か`false`なのかを判断し、今回は`true`なのでその中の`console.log(boolean);`が実行されます。

試しに`false`を入れて見てください。

コンソールに何も表示されないことが確認できたと思います。

このようにしてif文は条件を下に処理を分岐させることができます。

数値、文字を使用した条件比較

先ほどは変数の中の値が`true`か`false`のどちらかで判断していましたが、今度は条件を変えて比較を行なっていきます。

```
var num = 100
if (num == 100) {
  console.log(num);
}
```

これは`num`の中の値が`100`だった場合、ifの処理に入ると言うプログラムです。

今回は`num`に代入している値が`100`、比較条件の値も`100`なので条件を満たしたのでifの中の処理を実行しています。

試しにifの処理の中に`console.log(num == 100);`を追加してコンソールで確認してください。

`true`が表示されましたか？条件を比較した時に、変数の中の値と、条件となる値が一致したので`true`を返しています。

もし条件となる値が一致しなかった場合は`false`が返ってきます。

```
var text = "JavaScript"
if (text == "JavaScript") {
  console.log(text);
}
```

これは文字列の比較を行なっています。

代入した文字と条件となるもじが一致するかで処理の分岐を作っています。

より厳密な比較

先ほど紹介した`==`は数字や文字の比較はしますが、データ型が一致しているかまでは見ていません。

```
// 文字列として100を代入
var num = "100"
// 比較する値は数値型の100
if (num == 100) {
  // ==で比較しているので文字列でも数値型でもtrueになる。
  console.log(num);
}
```

それだと予期せぬバグが生まれてしまう可能性があるので比較を行う時は`===`を使用します。

これを使用するとデータ型も見るようになるので上記のパターンで比較を行うとエラーを返します。

基本的に比較をする場合は`===`を使ってください。

課題

1. 変数に`true`を代入し、ifの条件が`true`の時に`console.log();`で変数の中身を表示させる処理を作ってください。
2. 変数に`100`を代入し、ifの条件が`true`の時に`console.log();`で変数の中身を表示させる処理を作ってください。
3. 変数に`"JavaScript"`を代入し、ifの条件が`true`の時に`console.log();`で変数の中身を表示させる処理を作ってください。