

# else

---

前回のif文で条件に当てはまるものはif文の中のコードが処理されるのは理解できましたね。

今回はelseを使った処理を覚えていきましょう。

## elseとは

ifで条件に当てはまらなかった時の処理をelseを使用して処理できます。

elseは条件に関係なく、ifの条件に当てはまらなかった時だけ処理されます。

elseの使用はifがある時だけ使用できるので独立して使用はできません。

またif文とelseの間に処理書いてもエラーとなってしまうので注意してください。

## 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 = false;
if(boolean){
  console.log("trueの処理");
  console.log(boolean);
} else {
  console.log("falseの処理");
  console.log(boolean);
};
```

今回は変数にfalseを代入したのでifの処理には入っていきません。

その代わりにelse{}の中のコードが処理されているのが確認できたと思います。

上記で説明した、

if文とelseの間に処理書いてもエラーとなってしまうので注意してください。

についても実際のコードで説明します。

```
var boolean = false;
if (boolean) {
  console.log("trueの処理");
  console.log(boolean);
}
`console.log("テスト");`
else {
  console.log("falseの処理");
  console.log(boolean);
} ;
```

ifとelseの間に`console.log("テスト");`と入っているのですがifとelseの関係性が切れてしまいました。

この状態で読み込むとUncaught SyntaxError: Unexpected token 'else'と表示されます。

簡単に説明すると、意図しない、ブラウザ側と認識の違うところでelseが使われていると言う意味になります。

今回のようにエラーが発生するとコンソールにエラーが表示されているので実装した時は動くかどうかの確認も勿論ですが、コンソールにエラーを表示していないかも確認しましょう。

SyntaxErrorについて詳しく説明している文章があるのでご確認お願いします。

[参考記事](#)

## 課題

プログラミングの課題とテキストにする課題が混在していますのでJavaScriptを書いたコード上にコメントとして記述し、提出してください。

1. elseを使用する時の注意点
2. JavaScriptで実装した時の確認ポイント
3. 数値型を比較してelseの処理に入る処理を作ってください。
4. 文字列型を比較してelseの処理に入る処理を作ってください。