

条件分岐

前回はifとelseを使った処理を作りましたが、もっと細かい条件分岐を作りたい場合もあるのでそれを覚えていきましょう。

ifとelseの組み合わせ

elseの特性はifの処理に入らなかった場合、処理されます。

```
if("条件A") {  
    // 条件Aに一致した時の処理  
} else {  
    // 条件Aに一致しなかった時の処理  
}
```

一見便利に思えますが、言い換えると条件Aに一致しなかったらどんな場合でもこの処理に入っていきます。

意図的に条件Aの処理にも入らない、elseの処理にも入らない処理を作ることもあるのでその時は条件Bを作りましょう。

じゃあ早速作ってくださいと言った時にどんなコードを想像しますか？

プログラミング言語にはじめて触れている方はこんなコードを想像するかもしれません。

```
if("条件A") {  
    // 条件Aに一致した時の処理  
}  
if("条件B") {  
    // 条件Bに一致した時の処理  
} else {  
    // 条件Aと条件Bに一致しなかった時の処理  
}
```

処理は正しく動きますが冗長なのと、他の人が見ると、**if("条件A") {}**と**if("条件B") {}**の比較対象は別々だと捉える可能性も生まれてきます。

そう言ったことが起きないようにまとめて1つの処理にしたい場合は、**else if() {}**で条件Bの処理をすることで解消されます。

使用例

```
var num = 20;
if (num == 10) { // 条件A
  console.log("条件Aの処理");
  console.log(num);
} else if (num == 20) { // 条件B
  console.log("条件Bの処理");
  console.log(num);
} else { // 条件Aと条件Bに当てはまらなかった時の処理
  console.log("条件Aと条件以外の処理");
};
```

このようにすれば1つの処理としてまとまりができます。

注意点

elseと**if**は別々なので**elseif**と書かないようにしましょう。

こう書いてしまうと、違う処理とみなされエラーを起こします。

なので必ず**else**と**if**の間に半角スペースを入れましょう。

課題

文字列を使った条件分岐の処理を作ってください。

条件A、条件B、それ以外の構成をお願いします。