

変数と定数

今までは変数宣言を行う際`var`を使用していましたが他にも変数の宣言をする方法があるのと、それぞれ仕様があるのでそれを覚えてもらいます。

var以外の変数

var以外の変数の宣言は2つあります。

1つは`let`。もう1つは`const`です。

実務ではvarは使いません。ただなんとなく使わないのではなく、ちゃんと理由があって使いません。

それについてはこの後説明します。

それぞれの仕様

`var`:変数の再代入、再宣言が行える。

変数の再代入とは1度作った変数に対して再び値を代入することです。

再宣言は1度宣言した変数と同じ変数を作ることです。

`var`の場合のサンプルコードを用いて説明します。

```
// numと言う変数宣言を行い、1を代入している。
var num = 1
// 先ほど作成したnumへ2を再代入している。
num = 2
// 先ほど作成したnumとは別のnumを変数として作成し、3を代入している。(再宣言)
var num = 3
```

`let`:変数の再代入はできるが、再宣言はできない。

```
// numと言う変数宣言を行い、1を代入している。
let num = 1
// 先ほど作成したnumへ2を再代入している。
num = 2
// 先ほど作成したnumが既に存在しているのでエラーを起こす。
var num = 3
```

const: 変数の再代入、再宣言できない。

```
// numと言う変数宣言を行い、1を代入している。
const num = 1
// 再代入できないのでエラーが起きる。
num = 2
// 先ほど作成したnumが既に存在しているのでエラーを起こす。
var num = 3
```

注意点

説明しやすようにまとめて変数と説明しましたが、**const**は定数です。

varを使わない理由

以上のように変数と定数の使用を説明をしました。

varは再宣言、再代入ができてしまうのでグローバルな変数になりやすいです。

用意に再代入、再宣言ができてしまうのでvarで宣言した変数に対して何が入って来るか分からなくなってしまいます。

そうなって来るとバグの温床となってしまいます、なのでvarは基本使いません。

なので今後の課題でもvarの使用を**禁止**します。

補足

constは再代入できませんが配列やオブジェクトの操作はできます。

代入した配列やオブジェクトに新しい要素を追加や、代入した値を削除は可能です。

課題

1. **var**、**let**、**const**の使用についてまとめてください。
2. なぜ**var**を使わない理由をまとめてください。
3. 定数とは？