

SVG 資料第 7 回目 (その 1) 変数のスコープと仮引数

メディア専門ユニット I(SVG)

2016/6/6

変数のスコープとは

第 7 回目 (その 1)

メディア専門ユニット I(SVG)

JavaScript における変数のスコープ

- ▶ 変数のスコープとは使用されている変数が有効な範囲のこと
- ▶ JavaScript では変数は宣言しなくて使用できる。この場合、変数はグローバル変数 (非推奨)
- ▶ 関数内で var で宣言された変数はその関数内で有効で、関数の外からは見えない。
- ▶ 関数の途中で新たに var で変数を宣言しても、関数内の先頭で宣言されたものとみなされる
- ▶ let で宣言された変数は宣言されたブロック内で有効
- ▶ 関数の仮引数もそこで宣言されているものとして扱われる

var と let の違い-実行例

第 7 回目 (その 1)

メディア専門ユニット I(SVG)

JavaScript における変数のスコープ

```
1<!DOCTYPE html>
2<html>
3  <head>
4    <meta charset="UTF-8"/>
5    <title>var VS let</title>
6    <script type="text/ecmascript">
7      //
8        console.log('x=${x}');
9        x = "defined as global";
10       console.log('x=${x}');
11       {
12         var x = "defined as local?";
13         console.log('x=${x}');
14       }
15       console.log('x=${x}');
16       console.log('v=${v}');
17       let v = "defined as global";
18       console.log('v=${v}');
19       {
20         let v = "defined as local";
21         console.log('v=${v}');
22       }
23       console.log('v=${v}');
24     //]]&gt;
25   &lt;/script&gt;
26 &lt;/head&gt;
27 &lt;body&gt;
28 &lt;/body&gt;
29&lt;/html&gt;</pre></div>
```

確かめ (2)

```
x=undefined
x=defined as global
x=defined as local?
x=defined as local?
v=defined as global
v=defined as local
v=defined as global
```

- ▶ 変数 x は 12 行目で宣言されている。
- ▶ 9 行目の段階では変数に値が代入されていないので undefined がその値となっている。
- ▶ 11 行目から 14 行目のブロック内の var による変数の宣言と代入の結果がブロックの外に持ち出されている (15 行目)。
- ▶ 19 行目から 22 行目のブロック内で let で宣言されている変数 v の代入結果はブロックの終了後には存在しないことが 23 行目の出力でわかる。
- ▶ 16 行目のコメントを外すとの時点で変数 v が定義されていないというエラーが発生する。

やってみよう

第 7 回目 (その 1)

メディア専門ユニット I(SVG)

JavaScript における変数のスコープ

1. 17 行目と 20 行目にある変数 `v` の宣言を `let` または `var` に置き換え (`let` 以外の組み合わせ 3 通り)、16 行目のコメントを外した時の動作とその理由について報告する。
2. `var` と `let` による変数の宣言の違いをまとめる。

関数内の変数のスコープ

第 7 回目 (その 1)

メディア専門ユニット I(SVG)

JavaScript における変数のスコープ

```
1<!DOCTYPE html>
2<html>
3  <head>
4    <meta charset="UTF-8"/>
5    <title>変数のスコープ</title>
6    <script type="text/ecmascript">
7      //<![CDATA[
8        function F1(x) {
9          a = 2;
10         x = x+1;
11       }
12       function F2(x) {
13         var a = 20;
14         x[0] = x[0]+1;
15       }
16       function F3(x) {
17         b = 200;
18         x[0] = x[0]+1;
19         let b;
20       }
21     //]]>
22   </script>
23 </head>
24 <body>
25 </body>
26</html>
```

関数内の変数のスコープ-解説

関数 F1(x) について

- ▶ 変数 a に 2 を代入
- ▶ 変数 a は宣言されていないのでグローバル変数となる
- ▶ 仮引数 x の値を 1 増加

関数 F2(x) について

- ▶ 変数 a を関数内で宣言 (ローカル変数) し、20 を代入
- ▶ 仮引数は配列を想定
- ▶ 配列の先頭の要素の値を 1 増加

関数 F3(x)

- ▶ 変数 b を関数内に 200 を代入
- ▶ 仮引数は配列を想定
- ▶ 配列の先頭の要素の値を 1 増加
- ▶ その後で変数 b を宣言 (19 行目)
- ▶ したがって、17 行目の代入はこの関数のローカル変数に代入していることになる

実行結果

第 7 回目 (その 1)

メディア専門ユニット I(SVG)

JavaScript における変数のスコープ

```
>x=5;      //変数 x に値を設定（グローバル変数 x が定義）
5
>F1(x);
undefined //関数の戻り値。return がないので undefined
>x;
5          //F1(x) では仮引数の値を増加させているので
           //グローバルには影響がない
>a;        //F1(x) 内で変数 a がグローバル変数として定義された
2
>y=[1,2,3]; //関数 F2 に渡すための配列を定義
[1, 2, 3]
>F2(y);     //F2(y) を呼ぶ
undefined
y;
>[2, 2, 3]  //y[0] が変化している
>b;         //グローバル変数 b は存在していない
Uncaught ReferenceError: b is not defined(...)
>F3(y);
undefined
>b;         //F3(y) を実行した後もグローバル変数 b は存在していない
Uncaught ReferenceError: b is not defined(...)
```


関数の引数について (まとめ)

第 7 回目 (その 1)

メディア専門ユニット I(SVG)

JavaScript における変数のスコープ

- ▶ 関数の引数で数値のような単純なものは値渡し
- ▶ 関数の引数で配列は参照渡し
- ▶ 配列で渡された引数に手を加えると、元の配列が変化する。これにより、見つけにくいバグが発生する。
- ▶ JavaScript では配列が簡単に作れるので、関数の戻り値として配列を使うとよい。

やってみよう

第 7 回目 (その 1)

メディア専門ユニット I(SVG)

JavaScript における変数のスコープ

- ▶ 関数の引数がオブジェクトのとき、そのオブジェクトの値を変えたり、キーを追加すると、呼び出し元の値に変化があるか調べる。
- ▶ JavaScript の変数のスコープルールや関数の引数の仕様を他の言語と比較してその特徴を述べる。