



JAVASCRIPT 入門

鄭祥飛

2021/04/21


2021©ROCOCO





ブロック文

```
{  
    statement_1;  
    statement_2;  
}
```

- 波括弧中に複数の指令を集める。「IF」,「FOR」,「WHILE」など制御文によく使用する。
 - ブロック文内に「let」、「const」で定義する変数がブロック内部で有効です。
 - ブロック文内に「Var」で定義する変数がブロック外部で利用できる。（お薦めしないです）
 - ブロック外部に定義された変数がブロック内でも利用できる。（前提：ブロック前に定義された変数）
- 

条件文

```
If(condition) {  
    statement_1;  
    statement_2;  
}
```

```
If(condition) {  
    statement_1;  
    statement_2;  
} else {  
    statement_3;  
    statement_4;  
}
```

```
If(condition) {  
    statement_1;  
    statement_2;  
} else if(condition_2) {  
    statement_3;  
    statement_4;  
} else if(condition_3) {  
    statement_5;  
    statement_6;  
}
```

- ブロック文中に、「IF」, 「FOR」, 「WHILE」 など制御文にも使える。

Swith文

ある「変数」、「関数の返す値」の値により、色々な分岐処理の必要があれば、「Swith」文が使える。


```
switch (expression) {  
  case label_1:  
    statements_1  
    [break;]  
  case label_2:  
    statements_2  
    [break;]  
  ...  
  default:  
    statements_def  
    [break;]  
}
```

例：

```
let country = "japan";  
switch(country) {  
  case "japan":  
    console.log("speaking Japanese");  
    break;  
  case "china":  
    console.log("speaking Chinese");  
    break;  
  case "usa":  
    console.log("speaking English");  
    break;  
  default:  
    console.log("speaking English");  
    break;  
}
```



説明

- 1 . Switchの括弧中の式は変数、関数を設定できる。
 - 2 . Case :最初に式の値と一致するラベルを持つCASEから実行する。
 - 3 . Break: switch から抜き出し。
 - 4 . Default : 一致するラベルがない場合、プログラムはオプションの default 節を実行する。
 - 5 . 慣例により、default 節は最後の節に置きますが、そうしなければいけないわけではありません
- 

例外処理

- 1 . サーバー上のAPIを読み出しのプログラム、サーバー起動されていない。
- 2 . データベースと接続するプログラム、ネットワークが異常中。
- 3 . ファイルを書き出しプログラム、名前の同じファイルが既存し、別のソフトウェアで開いているなどなど

```
openMyFile();
try {
    writeMyFile(theData); // ここでエラーが投げられる可能性がある
}
catch(e) {
    handleError(e); // エラーを受け取り、それを処理する
}
finally {
    closeMyFile(); // 常にリソースが閉じられる
}
```



例外処理

- 1 . Try: 例外発生可能性部分のプログラム。
- 2 . Catch: 例外処理は発生する場合、処理方法。普通は例外の詳細情報を処理する。
- 3 . Finally: 例外発生時に適切にスクリプトを停止させることができます。例えば、スクリプトで使用していたリソースを解放しなければならない場合などです。

例外が処理しないと、プログラム全体は崩れる、続きの実行はできない。

例外を処理した場合、プログラムは続け処理できる。





例外を作成

`throw 'Error2';` // 文字列型

`throw 42;` // 数値型

`throw true;` // 論理型





循環處理

for 文
do...while 文
while 文
for...in 文
for...of 文

While文


```
let j = 0;  
while(j<5) {  
  j += 1;  
  console.log(j);  
}
```



For ... in文

```
const arr = [3, 5, 7];  
arr.foo = 'hello' ;  
for (let i in arr) {  
    console.log(i); // "0", "1", "2", "foo" が出力される  
}
```

```
for (let i of arr) {  
    console.log(i); // 3, 5, 7 が出力される  
}
```





For ... of文


循環処理できる対象に対して、ループして処理する。
循環処理できる対象は下記にある：
Array,map,set,arguments

```
const arr = [3, 5, 7];  
for (let i of arr) {  
    console.log(i);  
}
```

// 以下のプログラムは間違い

```
let test = {  
    name: "mars",  
    age: "10",  
    tel: "1111111"  
};
```

```
for(let i of test) {  
    console.log(i);  
}
```



For ... of と For ... inの区別

循環処理できる対象に対して、ループして処理する。

循環処理できる対象は下記にある：

Array,map,set,arguments

```
const arr = [3, 5, 7];  
for (let i of arr) {  
    console.log(i);  
}
```

// 以下のプログラムは間違い

```
let test = {  
    name: "mars",  
    age: "10",  
    tel: "1111111"  
};
```

```
for(let i of test) {  
    console.log(i);  
}
```




Break文

ループ、Switchブロック内の処理を終了する。

```
Let a=[1,3,5,2,4,4,5];  
for (let i = 0; i < a.length; i++) {  
    if (a[i] === 3) {  
        console.log(a[i] );  
        break;  
    }  
}
```

```
for (let i = 0; i < a.length; i++) {  
    if (i === 3) {  
        console.log(a[i] );  
        break;  
    }  
}
```





continue文

今回の処理は、continueまで終了する。次のループを続き処理する。

```
Let a=[1,3,5,2,4,4,5];  
for (let i = 0; i < a.length; i++) {  
    if (a[i] === 3) {  
        console.log(a[i] );  
        continue;  
        console.log( "test" );  
    }  
}
```






宿題

1. 下記のソースコードはどのような内容を実行しますか、どのように理解していますか

```
let i = 0;
let n = 0;
while (i < 5) {
  i++;
  if (i === 3) {
    continue;
  }
  n += i;
  console.log(n);
}
```



宿題

1. 下記データの最小値を求める
[100,21,14,22,55,22]
2. 1から10000まで、すべての整数値の合計値を求める
3. 下記年齢の所属する年齢代を出力する。例：
25才の場合、“20代”を出力する。
39才の場合、“30代”を出力する。

[11,26, 28,88,91,65,33,22,73,34,62]