

Javascript 入門

関数

鄭祥飛@2021/06/07

なぜ関数が必要

大型プロジェクト、ソースコードの再利用
ソフトウェアのモジュール化

例：
`parseInt()`
`sort()`
`substr()`

関数の定義

```
function square(number) {  
    return number * number;  
}
```

- function :関数定義のキーワード
- square: 関数名前
- number: パラメータ。関数体内有効です。必須ではない。複数設定できる
- return: 処理結果を返すキーワードです。必須ではない

関数の作成時にの注意点

- 一つ関数は一つ機能だけ実現する。
- 関数名はと実現する機能と適當する必要があります。

無名関数の定義

```
const square = function(number) {  
    return number * number  
}
```

- 関数の名前がないです
- 関数を変数に付与する
- 普通の関数も変数に付与できる

アロー関数の定義

```
const square = (number)=>{  
  return number * number  
}
```

関数の呼び出し

- 関数を定義した後、呼び出しないと関数中のソースコードが実行されません。

```
let number = 5;  
let result = square(number);  
console.log(number); // 5  
console.log(result); // 25;
```

関数のパラメータ (一)

- オブジェクトを引数として渡すと、関数がオブジェクトのプロパティを変更した場合、その変更が関数外でも有効になります。(array,map,set,object)

```
function myFunc(theObject) {  
  theObject.make = 'Toyota';  
}  
let mycar = {make: 'Honda', model: 'Accord', year: 1998};  
myFunc(mycar);  
console.log(mycar);
```


関数のパラメータ (二)

- オブジェクトを引数として渡すと、元の値に影響しないようにどうする

```
function myFunc(theObject) {  
  let newObject = {... theObject};  
  newObject.make = 'Toyota2';  
  return newObject;  
}  
let mycar = {make: 'Honda', model: 'Accord', year: 1998};  
console.log(myFunc(mycar));  
console.log(mycar);
```

関数のパラメータ (三)

- 関数自体もパラメータとして、関数にわたす事ができる

```
function log(f) {  
    console.log( f(5,6) );  
}  
const fun = (a,b)=>{  
    return a + b;  
};  
log(fun); // funは関数です。
```

関数のパラメータ (四)

- 初期値があるパラメータ

```
function test(a,b=1) {  
    return a + b;  
}  
console.log(test(1));  
console.log(test(1,5));  
console.log(test());
```

関数のパラメータ (五)

- パラメータが不定の関数

```
function multiply(multiplier, ...theArgs) {  
    return theArgs.map(x => multiplier * x);  
}  
var arr = multiply(2, 1, 2, 3);  
console.log(arr); // [2, 4, 6]
```

- multiplyを呼び出し時に、任意のパラメータを設定できる

関数の戻す値

- 関数は return 文を実行すると終了する。
- 関数中にreturn 文は必須ではない
- 何でもreturnできる、数字、文字列、配列、関数など

再帰関数

- 関数中に、自分を呼び出し。
- 注意：終了条件が必ず必要、そうしないと止まらないです

```
function loop(x) {  
    if (x >= 10) // "x >= 10" が終了条件 ("!(x < 10)" と同等)  
        return;  
    // 何らかの処理を行う  
    loop(x + 1); // 再帰呼び出し  
}  
loop(0);
```

関数中に関数を定義する

初級レベル段階、お薦めしないです。

```
function addSquares(a, b) {  
  function square(x) {  
    return x * x;  
  }  
  return square(a) + square(b);  
}  
  
a = addSquares(2, 3); // returns 13  
b = addSquares(3, 4); // returns 25  
c = addSquares(4, 5); // returns 41
```

宿題

一つパラメータの関数を作成する。この関数は指定する「パラメータ」より小さいのすべてのフィボナッチ数字を出力する

例：パラメータを 50 を設定すると下記の数字を出力する
1, 2, 3, 5, 8, 13, 21, 34,

パラメータを 20 を設定すると下記の数字を出力する
1, 2, 3, 5, 8