関数

JavaScriptと関数

JavaScriptに限らず、プログラム言語には関数という機能があります。
関数を使うことでプログラムは同じ記述の重複を避け、コードの見通しをよくすることが出来ます。

例えるなら自動販売機です。

自動販売機にお金を入れるとジュースが出てくる

自動販売機にお金を入れると品物が出て来ます。

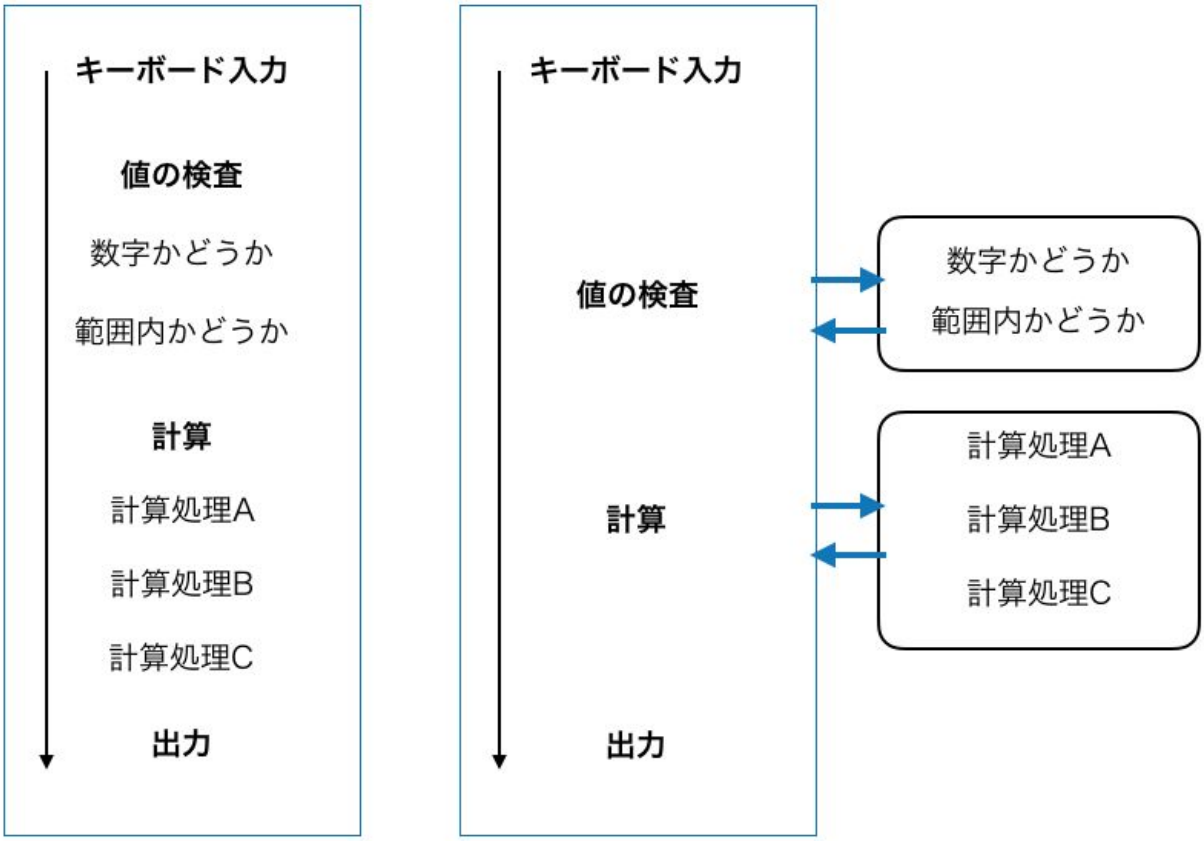
2倍する関数に「2」を入れると「4」が出てくる

関数は値を受け取ると、定められた計算や処理を実行して結果を返します。
これをプログラムにすると以下のような感じになります。

```
function nibai(x){
    return x * 2;
}
```

関数を利用する理由

関数を使わないプログラム(左)と関数を活用したプログラム(右)



関数化することでプログラムの見通しが良くなり、同じようなコードの重複を避けることが出来ます。結果、わかりやすいくメンテナンス性の高いプログラムになります。

値を返す関数と値を返さない関数

関数と戻り値

実行した結果、何か値を生み出す関数と何も値を生み出さない関数があります。
実行結果を**return**した場合、値は**戻り値**として関数を呼び出した命令に帰って来ます。

戻り値がある関数

例）ATMで現金を引き出す行為は戻り値がある関数


```
//口座番号が記載されたキャッシュカード
const card = "12345678";
//暗証番号
const password = "9999"
#関数を呼び出す(returnされた値がmy_moneyに代入される)
my_money = atm_get_money(card, password);
function atm_get_money(card, password){
    #口座からの引き出し処理
    return your_money;
}
```

戻り値のある関数は実行結果を受け取る必要があるため、実行結果を変数に格納するか、別の関数の引数に設定するなど、処理を引き継ぐ必要があります。

戻り値がない関数

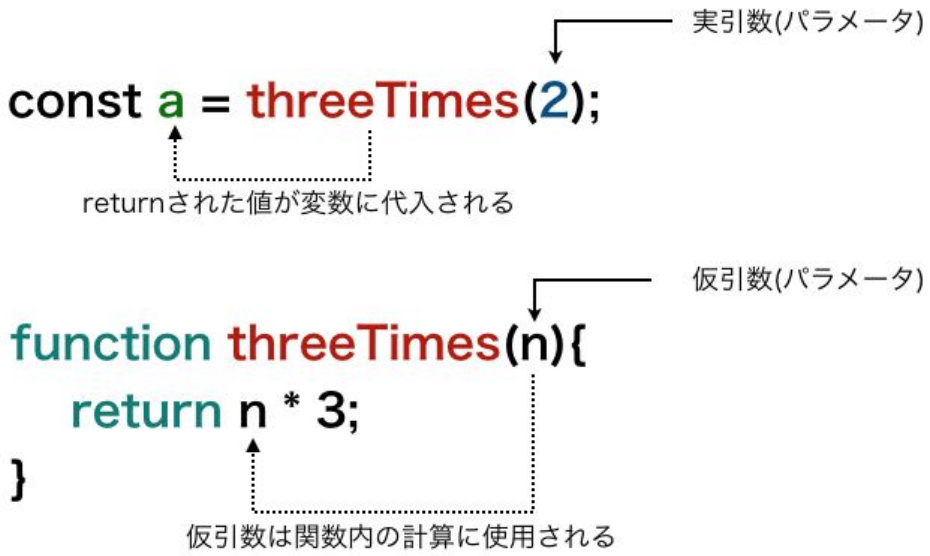
関数によっては画面に出力する、変数や配列、その他オブジェクトを変更するなど、呼び出し元に**実行結果の値を戻さない**場合もあります。

例）ATMで口座に入金する行為は戻り値の無い関数


```
//口座番号が記載されたキャッシュカード
const card = "12345678";
//現金
const cash = 10000;
atm_update_account(card, cash);
function atm_update_account(card, cash){
    //口座に現金を入金する(残高を増やす処理)
}
```

戻り値の無い関数はreturn処理がありません。

関数のルール



IDにMyNameとついたタグを「山田花子」書き換えます。
ただし、タグで修飾したいとします。

```
$('#MyName').html('<strong>山田花子</strong>');
```

と記述することも可能ですが、同じような処理が繰り返される場合は関数を使って効率的に記述することが出来ます。

```
$('#MyName').html(strongString('山田花子'));

function strongString(name){
    return '<strong>' + name + '</strong>';
}
```

値を返さない関数

関数の中には値を返さない関数もあります。
以下のassert関数は呼び出されるとコンソールにラベルと値を書き出し終了します。呼び出し元に何か値を戻す訳ではありません。

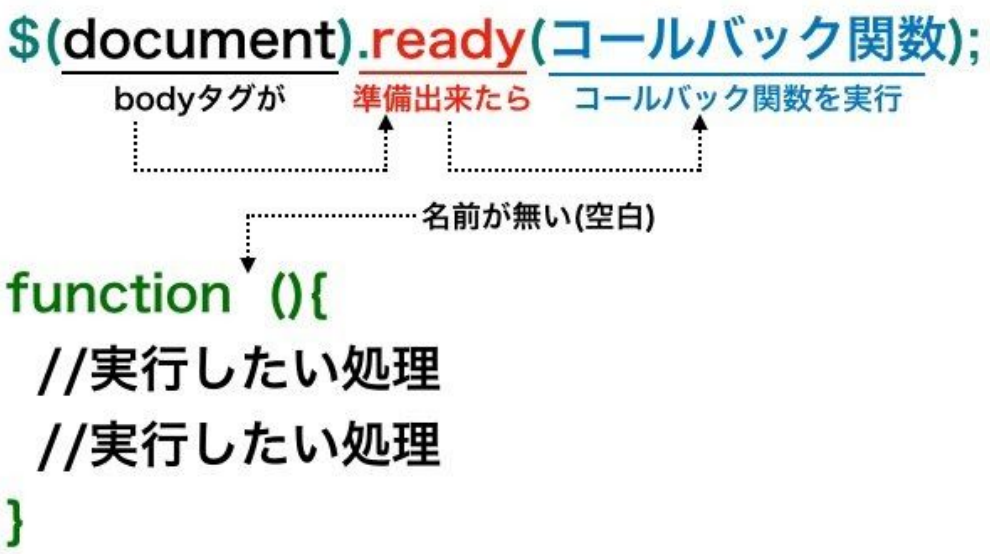
```
function assert(label, val){
    console.log( label + ':' + val);
}
```

関数定義の方法

関数定義には何種類か方法があります。以下は変数に関数を代入することで関数を定義しています。定義方法は異なりますが上の関数と全く同様に扱えます。

```
const assert = function(label, val){
    console.log(label + ':' + val);
}
```

無名関数とコールバック



イベントが起きたら実行される関数をコールバック関数と言います。
jQueryでよく見るこのスタイルはdocument(ブラウザーに読み込まれたウェブページ)がready(準備完了)したら、コールバック関数を実行します。

下記は同じ意味になります。

```
$(document).ready(function(){
            console.log("読み込み完了");
});
```

```
$(document).ready(assert());
function assert(){
            console.log("読み込み完了");
}
```

*無名関数はそのタイミングで一度切りしか使用しないため名前をつける必要がない。

組み込み関数

独自に定義する以外にJavaScriptには事前に用意された組込関数があります。代表的な関数をいくつか紹介します。

乱数を生成する

実行するたびに異なる数を生成します。
サイコロを振るイメージです。

乱数を生成する関数はMath.random()です。
Math.random()は0から1までの小数を含む値を生成します。

整数の乱数が欲しい場合は小数点以下を切り捨てるMath.floor()と合わせて使用します。

```
//0〜9までの乱数を生成する
Math.floor(Math.random()*10);
```