

# JavaScript講座 第3回

2026/01/29

Tam

## 第3回／1月29日(木)

「関数を使ってみよう」

- 関数とは
- 関数を作ろう
- ブラウザからボタン操作でJavaScriptを実行させよう

# 関数とは

プログラミングにおける「関数（かんすう）」を一言でいうと、  
「特定の処理をひとまとめにした部品」です。

# 万年1月カレンダー

先週作成した1月のカレンダーを、任意の年で出力できるようにしてみましょう。

```
function calendar(year) {  
  for (let d = 1; d <= 31; d++) {  
    console.log(year + "年1月" + d + "日");  
  }  
}
```

これで関数が作成できます。

※関数は作成しただけでは、実行されません。

# 関数を使う

関数を使うことを、「関数を呼び出す」、「関数を実行する」などと表現します。

以下のように、関数を引数付きで記述することによって、関数を呼び出すことができます。

```
calendar(2026);
```

2001年～2026年の1月カレンダーを作りたいければ、以下のようにします。

```
for (let year = 2001; year <= 2026; year++) {  
    calendar(year);  
}
```

# ライブラリ

JavaScript では、いくつもの便利に使える関数が予め用意されています。

```
let date = new Date(2026, 0, 29); // 2026年1月29日を取得  
let wday = date.getDay(); // date の曜日を数字で取得  
console.log(wday);
```

# 定期呼び出し

関数を使えるようになると、1秒毎の定期的な呼び出しなどが出来るようになります。

第1回で作った時計を、自動更新してみましょう。

```
<html>
  <head>
  </head>
  <body>
    <p id="time">00:00:00</p>
    <script type="text/javascript">
      // ここに次のページの JavaScript を差し込んでください。
    </script>
  </body>
</html>
```

## setInterval関数

```
// 時刻を表示する display関数を作成
function display() {
    // id="time" となっている HTML タグを取得
    let element = document.getElementById("time");
    // 現在時刻を取得
    let t = new Date();
    // 現在時刻を HTML に埋め込む
    element.innerText = t;
}

// display関数を 1000ミリ秒毎に呼び出すように設定
setInterval(display, 1000);
```



# 返り値

関数から値を返却することも出来ます。カレンダーを HTML で表示してみましょう。

```
function calendar() {  
    let html = "<table><tr>";  
    let wday = 4; // 1月1日は木曜日  
    for (let i = 0; i < wday; i++) {  
        html += "<td></td>"; // 1日より前を空白で埋める  
    }  
    for (let d = 1; d <= 31; d++) {  
        if (wday === 0) {  
            html += "</tr><tr>"; // テーブルを改行  
        }  
        html += "<td>" + d + "</td>";  
        wday = (wday + 1) % 7;  
    }  
    html += "</tr></table>";  
    return html;  
}
```

# カレンダーHTML

```
<html>
  <head>
  </head>
  <body>
    <p id="calendar"></p>
    <script type="text/javascript">
      // ここに前のページの calendar 関数を挿し込んでください。
      // id="calendar" となっている HTML タグを取得
      let element = document.getElementById("calendar");
      // カレンダーを作成
      let table = calendar();
      // HTML を差し込む
      element.innerHTML = table;
    </script>
  </body>
</html>
```

## 演習

1. カレンダー table の先頭に、ヘッダとして日～土を表示してみましょう。
2. 任意の年の1月カレンダーを作ってみましょう。（曜日の取得方法を思い出してください。）
3. 任意の年月のカレンダーを作ってみましょう。月末はすべて31日まで出力して構いません。
4. 月末が30日や28日までしかない月に対応してみましょう。（配列が便利に使えます。）
5. うるう年に対応してみましょう。

# ブラウザからの呼び出し

ブラウザからの操作で JavaScript を呼び出すことができます。

```
<html>
  <head>
  </head>
  <body>
    <p id="calendar" onclick="display();">Push</p>
    <script type="text/javascript">
      // ここに calendar 関数を挿し込んでください。
      // ここに次のページの display 関数を挿し込んでください。
    </script>
  </body>
</html>
```

Push をクリックすると、p タグの onclick に書かれた JavaScript が呼び出されます。

呼び出す display 関数は以下のようにしましょう。

```
function display() {  
  // カレンダーを作成  
  let table = calendar();  
  // id="calendar" となっている HTML タグを取得  
  let element = document.getElementById("calendar");  
  // HTML を差し込む  
  element.innerHTML = table;  
}
```

# 万年カレンダーtable

HTML から値を取得してみましょう。

```
<html>
  <head>
  </head>
  <body>
    <input type="text" id="year">
    <p id="calendar" onclick="display();">Push</p>
    <script type="text/javascript">
      // ここに JavaScript を挿し込んでください。
    </script>
  </body>
</html>
```

# calendar 関数

```
function calendar(year) {
  let html = "<table><tr>";
  let wday = new Date(year, 0, 1).getDay(); // 1月1日の曜日を取得
  for (let i = 0; i < wday; i++) {
    html += "<td></td>";
  }
  for (let d = 1; d <= 31; d++) {
    if (wday === 0) {
      html += "</tr><tr>";
    }
    html += "<td>" + d + "</td>";
    wday = (wday + 1) % 7;
  }
  html += "</tr>";
  return html;
}
```

## display 関数

```
function display() {  
  // id="year" となっている HTML タグを取得  
  let element1 = document.getElementById('year');  
  // カレンダーを作成  
  let table = calendar(element1.value);  
  // id="calendar" となっている HTML タグを取得  
  let element2 = document.getElementById("calendar");  
  // HTML を差し込む  
  element2.innerHTML = table;  
}
```