

# JavaScript 講座 第1回

## 「JavaScriptを始めよう」

2026/01/15

Tam

## ◆第1回／令和8年1月15日(木)

「JavaScriptを始めよう」

- プログラミングの基礎知識
- JavaScriptとはどういったものか。何ができるのか
- 本講座でのJavaScript開発環境について
- JavaScriptを始めよう

言語仕様に基づいたプログラミングJavaScriptとHTMLの関係

# プログラミング言語とは？

（大まかに言えば）機械に依頼する作業の指令書。

プログラミング言語は、相手が機械なので、融通がまったく効きません。

1文字間違えても、スペルミスしてもエラーになります。

# JavaScript とは？

長所：

1. ほとんどのブラウザの上で動く。
2. インタプリタ（一行ずつ解釈しながら実行）であるため、作りやすい。

短所：

1. インタプリタのため、コンパイラ（全て解釈してから実行）より、実行速度が遅い。
2. 後付けで度々拡張が行われたため、プログラミング言語として難解な部分がある。

※「インタプリタ」、「コンパイラ」という単語の意味は、今は知る必要はありません。後で学んでいきましょう。

## 開始前の準備

1. デスクトップにフォルダを作成しましょう。  
フォルダ名は "JS1" などの区別のつく名前にして下さい。
2. Visual Studio Code を立ち上げましょう。
3. 上で作成したフォルダを開いておきましょう。

# Hello, World

最初の例文として「Hello, World」を表示するだけのプログラムがよく使われます。

※プログラミング言語界隈の慣習として定着しています。

早速やってみましょう。

# 入力方法

今回のプログラムの入力には、先に立ち上げた Visual Studio Code（VSCode と略することが多い）を使っていきます。

約束事：

1. 基本的に入力は **半角モード**で行ってください。  
漢字やひらがな入力モードで入力した文字は受け付けてもらえません。
2. VSCode で **赤線が出た部分**には、たいてい **入力間違い**が含まれています。  
マウスカーソルを合わせると間違いの内容が表示されますので、  
メッセージに従い修正していきましょう。

# HTML

まずは以下の HTML の入力をお願いします。

ファイルは、先程作成したフォルダ内で「hello01.html」などとしてください。

```
<html>
  <head>
  </head>
  <body>
    <p id="greeting">Hello, World.</p>
    <!-- ここはコメントアウトと呼ばれ、プログラムの実行に影響を与えません。
        コメントアウトにはプログラムの解説などを書いています、
        入力する時間がもったいないので、入力しないでください。 -->
  </body>
</html>
```



# JavaScript を埋め込んでみよう！

```
<html>
  <head>
  </head>
  <body>
    <p id="greeting">Hello, World.</p>
  </body>
  <!-- ここから追加 -->
  <script type="text/javascript">
    // ここは JavaScript のコメントアウトです。
    // コメントアウトは入力しないでください。
    // id="greeting" となっている HTML タグを取得
    let element = document.getElementById("greeting");
    // "Good Evening" というテキストを HTML タグに差し込む
    element.innerText = "Good Evening";
  </script>
  <!-- ここまで -->
</html>
```

# おめでとうございます！

正しく入力できていれば、

**"Hello, World"** の部分に **"Good Evening"** が表示されています。

あなたにとっての初めての JavaScript プログラムが出来上がりました。

## 正しく動かなかったら？

まずは Visual Studio Code 上で赤線が出てないか確認しましょう。

次に Debug Console 上でエラーが出てないか確認しましょう。

※Debug Console の表示の仕方は画面で説明します。

# デバッグコンソール

ブラウザでのエラー情報などが表示されるウィンドウがあります。

表示のさせ方：

1. ブラウザの右上の「 」マークをクリックします。
2. 「その他のツール」 → 「デベロッパーツール」とクリックしていきます。
3. 表示されたウィンドウから「Console」タブをクリックします。

# 物足りないですね。

文字に色を付けてみましょう。

```
<!-- ここから追加 -->
<script type="text/javascript">
  // id="greeting" となっている HTML タグを取得
  let element = document.getElementById("greeting");
  // HTML を差し込む
  element.innerHTML = '<span style="color: red;">Good Evening</span>'
</script>
<!-- ここまで -->
```

現在時刻を表示するように改造してみましょう。

```
<!-- ここから追加 -->
<script type="text/javascript">
  // id="greeting" となっている HTML タグを取得
  let element = document.getElementById("greeting");
  // 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;
</script>
<!-- ここまで -->
```

※この現在時刻は、一回表示して終わりなので、1秒毎に時刻を刻んだりは出来ません。

# デバッグコンソールに出力

以下はデバッグコンソールに "Hello, World" を表示する例です。

早速試してみましょう。

```
<!-- ここから追加 -->  
<script type="text/javascript">  
  // Hwllo, World を表示する  
  console.log("Hello, World.");  
</script>  
<!-- ここまで -->
```

## 現在時刻をデバッグコンソールに表示する

```
<!-- ここから追加 -->
<script type="text/javascript">
  // 現在時刻を取得
  let t = new Date();
  // 現在時刻を表示する
  console.log(t);
</script>
<!-- ここまで -->
```



## この講習会の目標について

第6回の最終回に課題として、以下のものを各自に作成してもらいます。

1. カレンダー
2. 前後の月や視点した年月のカレンダーを表示できる
3. 日付の欄にメモを残せる機能付
4. その他各自の欲しい独自機能

この目標に向かって、必要となる機能の紹介を行っていきます。