

ソフトウェア開発 第1回目授業

平野 照比古

2018/9/28

授業内容 – 概略

- JavaScript を通常の計算機言語として利用するための解説を10回の講義で行う。
- 進度が今までのプログラミングの授業より早いので復習をよくすること。
- パソコンを授業に持参す必要がある場合はその旨、前回の授業で指示する（多分ない）。






授業内容 – 復習用課題

- 配布された復習用の課題は用紙に解答を直接記入するか、印刷したものを添付して次回の授業の前日の木曜日 10 時までに 6 階の平野のレポートボックスに提出
- 復習用課題と同時に配布されるルーブリック評価表も自己評価を付けて提出
- レポートの内容はルーブリック評価表の「標準的」の項目を参照すること。
- レポート課題に関して (必須) が付いているものは必ず解答すること。
- 復習課題の提出方法はメール提出に変更する場合もある。

授業内容 – その他

- 復習課題とそのルーブリック評価表は次回の授業の開始時に添削、評価をしたものを返却する。
- 復習用課題は 10 回程度を予定している。成績評価の 60% を占める。
- 最終回の授業に試験を行う。成績評価の 40% を占める。
- 配布資料等は <http://www.hilano.org/hilano-lab> で公開する予定である。

参考図書

-  E. Brown, 初めての JavaScript 第 3 版, -ES2015 以降の最新ウェブ開発, オライリージャパン
-  David Flanagan, JavaScript 第 6 版, オライリージャパン
-  David Flanagan, JavaScript クイックリファレンス第 6 版, オライリージャパン
-  Nicholas C. Zakas, ハイパフォーマンス JavaScript, オライリージャパン
-  Nicholas C. Zakas, メンテナブル JavaScript 読みやすく保守しやすい JavaScript コードの作法, オライリージャパン

授業回数	内容
第 1 回	授業のガイダンスとブラウザの開発者モードについて JavaScript の実行環境の確認
第 2 回	JavaScript が取り扱うデータ データの型と演算子に関する注意など
第 3 回	関数の定義方法と変数のスコープ
第 4 回	オブジェクトの定義とクラス
第 5 回	オブジェクトの詳細 関数に要るオブジェクトの定義、エラー処理
第 6 回	正規表現と文字列の処理
第 7 回	DOM の利用 HTML 文書、CSS と DOM の基礎

授業回数	内容
第 8 回	イベント処理 イベントモデルとイベント処理の例
第 9 回	PHP 超入門 PHP に関する簡単なプログラム
第 10 回	サーバーとのデータの交換 PHP 入門の続きとサーバーとのデータ交換の基礎
第 11 回	jQuery DOM の処理を簡単にするライブラリーの紹介
第 12 回	XML ファイルの処理 JavaScript と PHP による XML ファイルの処理の例
第 13 回	クラスの例
第 14 回	システム開発のヒント
第 15 回	最終試験と解説

JavaScript の実行方法

最近のブラウザは JavaScript の統合環境を提供している。

- Chrome のデベロッパーツール、FireFox の Web 開発、Internet Explorer の開発者ツールなどは JavaScript におけるプログラミングにおいてデバッグなどの統合環境を提供
- これらのツールは「F12」または「Control+Shift+I」というショートカットキーで表示、非表示ができる。
- このときに表示されるタブの内容は名前が異なっても機能はほとんど同じ

Strict モードについて

- ECMAScript の最新版では strict モードと呼ばれる厳密な解釈をするモードが導入
- このモードでは従来見つけにくい単純なバグがエラーとなる
- プログラムを strict モードにするためにはプログラムの先頭に `"use strict";` を記述
- 関数の定義の直後に `"use strict";` を記述するとその関数内は strict モードになる。

非 strict モードと strict モードの主な違い

	非 strict モード	strict モード
変数の宣言	必要ではない	必要
書き込み不可なプロパティへの代入	エラーが発生しない	エラーが発生
関数の arguments オブジェクトの値の変更	可能	不可能
関数の arguments.caller	参照可能	エラーが発生
関数の arguments.callee	参照可能	エラーが発生
8 進リテラル (0 で始まる数)	使用可能	エラーが発生

JavaScript の実行例

```
1 <!DOCTYPE HTML>
2 <html>
3 <head>
4 <meta charset="UTF-8"/>
5 <title>初めての JavaScript</title>
6 <script type="text/ecmascript">
7 //<![CDATA[
8 function foo(){
9     let i;
10    for(i=1;i<10;i++) {
11        console.log(`${i} ${i*i}`);
12    }
13 }
14 alert("デベロッパーツールからコンソールを開いてコンソールから foo(); と入力し
15 //]]>
16 </script>
17 </head>
18 </html>
```

JavaScript の実行例-解説 (1)

- 1 行目 HTML5 における HTML 文書の宣言
- 4 行目 このファイルの文字コード (エンコーディング) を UTF-8 に指定。
- 6 行目 スクリプトの開始の要素。プログラミング言語が ECMAScript であることを宣言している。
- 7 行目 `//` は行末までの部分をコメントにする JavaScript の記法。残りの部分はこれ以降 12 行目までは通常の文字として解釈することを指定 (CDATA セクションの開始)。
7 行目と 13 行目を消去したらどうなるのか確認することまたその理由も考えること¹。

¹最近のブラウザではエラーが起きないかもしれない。

JavaScript の実行例-解説 (2)

- 8 行目は関数 `foo()` の宣言。13 行目までがこの関数の定義範囲
- 9 行目は変数 `i` の宣言
- 10 行目 C 言語などでおなじみの繰り返しの指定
- 11 行目 引数内の式をコンソールに出力
- 14 行目はメッセージボックスにコンソールを開くことを指示。

- ◀ ◻ ▶ ◀ ◻ ▶ ◀ ≡ ▶ ◀ ≡ ▶ ≡ ▶ ↺ 🔍 ↻