

4.3 DynamicHTML 入門

DynamicHTML, DHTML

→ [復習] Dynamic vs Static

基本: 動きのあるページ(動的なページ)を作成する = 動きのあるコンテンツを配置する

CSS と JavaScript などのスクリプト言語を用いてクライアントサイドで動的変更

- ・ HTML ページは基本的に静的な構造(印刷文書のイメージ?)
- ・ 動的なページ・コンテンツの利用が可能
 - = ページを再読み込みすることなく、ページの一部を書き換え・スタイル変更可能



ポイント) この学習課題はこの授業(Web サイト制作)だけでは完結しません

→ C 言語プログラミング ...

関連) サーバサイド JavaScript, Node.js など

JavaScript

LiveScript(Netscape, 1995) → JavaScript(Sun Microsystems) → 標準化

特徴など

- スクリプトとは? 実行可能なコンテンツ
- JavaScript は Java とは(似ているけど)違う
- 大文字・小文字は区別する
- クォートすると文字列に(文字は ' でも " でも)

関連)

Ajax(Asynchronous JavaScript and XML) - Javascript を使った XML データの非同期取得
jQuery - 便利な機能をまとめたライブラリ

HTML でのスクリプトの記述・読み込み

- 1) `<script type="text/javascript" src="URL"> ~ </script>`
`<script>~</script>` で直接書き込んでも ok
- 2) イベント属性に記述
onload, onfocus, onsubmit, onclick, onmouseover, ... など多数
例) `<body onload="~">`
例) `window.onload = function() { ~ } ... HTML 内にコードを入れたくない`
例) `<input type="button" onclick="~">`
- 3) 擬似 URL (a 要素にスクリプトを記述) → (関連)ブックマークレット
例) ` ~ `

文法(プログラミング言語の文法)

→ C 言語, Java 言語

- ・ 空白とスペースと改行
- ・ コメント文
- ・ 文の終わりのセミコロン
- ・ ブロック
- ・ 演算子(計算)
- ・ 変数(var, 型はない)
- ・ 数値と文字列の扱い
- ・ 制御文(if, switch, for, while)
- ・ 関数(function)
- ・ 配列
- ・ DOM(Document Object Model)
 - window を頂点にしたツリー構造でオブジェクトが決まっている
- ・ イベントハンドラー
 - イベントとは何らかの動作のこと

いろいろ思うところ

- ・ スクリプト実行は HTML がきちんと読み込まれているのか?
記述する場所などの依存する(ことがある)
 - コンパイラ系言語よりもエラーがはっきりしない
- ・ デバッグが大変(なこともある)
- ・ ブラウザ・実装系によって動作の違い(もありうる)
- ・ 総じて面倒くさい(と思うことも多々ある)



関連) HTA(HTML Application) という DynamicHTML を使用した Windows アプリ作成の仕組み

動きがあるページまとめ

- ・ アニメーション GIF, アニメーション PNG (APNG)
 - 通常の画像と同じ扱い, img 要素で対応
- ・ 動画などのマルチメディアデータ
 - 動画 MPEG (mpg, mp4, ...), 音声 (mpeg audio, ...)
 - 共通して使用できる形式を用いる
 - embed, object 要素でプログラムを埋め込む
- ・ Adobe Flash (旧マクロメディア Flash) → Adobe Animate CC?
 - マルチメディア統合再生環境
 - クライアント側ブラウザが対応していること (プラグイン, アドオンなど)
 - embed, object 要素でプログラムを埋め込む
 - 膨大な過去の資産が存在しているが、将来性は?
- ・ Java アプレット
 - クライアント側ブラウザが対応していること
 - embed, object 要素でプログラムを埋め込む
- ・ JavaScript
 - クライアントサイド の 入門編は学習済み
 - 関連) JavaScript の便利なフレームワーク
 - 関連) サーバサイドとの連携 → Node.js
- ・ HTML5 での canvas などの利用
- ・ その他



HTML+CSS+JavaScript → これから学ぶには一番現実的なソリューション

課題) サーバ側の技術についても調べてみよう

実習) JavaScript のフレームワークを調べて使ってみましょう

関連) CGI (Common Gateway Interface)

サーバ上のプログラムと連携動作させるための仕組み

例) データベースとの連携

例) C 言語で作成したプログラムの出力を Web ページへ