

Web について

谷口 陽音

2023 年 6 月 22 日

1 Web ページの仕組み

1.1 Web ページの構成

我々が見ている Web ページは、主に 3 種類程度のファイルから構成されている。

1.1.1 HTML

HTML は、Hyper Text Markup Language の略であり、主に Web ページの基本的なパーツを記述する。HTML の最新版は HTML5^{*1} であり、これはオープンソース^{*2}である。HTML は「タグ」というものを用いて記述する、特徴的な言語である。

1.1.2 CSS

CSS は、Cascading Style Sheets の略であり、主に HTML で記述された要素の見た目や Web ページのレイアウトを制御する。CSS の最新版は 2011 年から策定が進められている CSS3^{*3}であり、これもまたオープンソースである。なお、CSS は Markdown のレイアウト調整などにも用いることができる。

1.1.3 JavaScript

JavaScript は、主に Web ブラウザ上で動作するプログラミング言語^{*4} であり、Web プログラミング界のデファクト・スタンダード^{*5} である。JavaScript も例によってオープンソースである。言語仕様の面では JavaScript と Java は全くもって関連性がない^{*6} ことに注意が必要である。

^{*1} ここでは便宜上 HTML5 としたが、正確には「HTML Living Standard」である。しかし、現在において「HTML5」は広く浸透しており、かつこうなった背景にはかなり複雑な歴史があるため、ここでは割愛する。しかし、経緯自体はかなり興味深いものであるため、気になる人は Web 検索をかけてみることを推奨する。

^{*2} オープンソースとは、GitHub などソースコードを公開しながら開発された製品や、その開発のことを指す。なお、このような言語は殆どの場合オープンソースである。詳しくは JavaScript の詳細のところでふれる。

^{*3} まだ正式に勧告されていない (執筆当時)。

^{*4} HTML はマークアップ言語、CSS はスタイルシートであり、いずれもプログラミング言語ではない

^{*5} 「業界標準」の意。

^{*6} これも HTML 同様少々複雑なため説明を割愛する。なお、これについても是非とも Web サイトで調べてみてほしい。なお、現在商標権を持っているのは Oracle であるが、JavaScript のネイティブのブラウザは Firefox であるため、混同しないよう十分に注意する必要がある。

1.2 フロントエンドとバックエンド

Web ページには、ユーザーが直接見れる部分と直接は見れない部分が存在する。
前者をフロントエンド、後者をバックエンドという。
フロントエンドは §1.1 で触れた内容である。バックエンドは次の §3.3 で触れる。

1.3 バックエンド

§1.2 で述べた通り、Web ページの利用者から見えない部分をバックエンドと呼ぶ。
バックエンドでは次のようなことを行う。

1.3.1 データベース

まず、バックエンド側はフロントエンド側から受け取った情報を基にデータベースを操作することが 1 つの大きな仕事である。
ここで使われるのが SQL である。
SQL は、「SQL 文」というものを使ってデータベースにデータを追加・削除・編集したり、必要な情報を必要なぶんだけ取得したりする。
ここで、前者の「データ」を「レコード」といい、後者のときに取得したデータを「クエリ」、データを取得することを「抽出する」という。

1.3.2 WebAPI

次に、バックエンド側の処理として、WebAPI を作成し、Web 上のデータの処理を行うことが挙げられる。
WebAPI とは、Web ページ関係のプログラミングを行うためのインターフェース^{*7}である。

1.3.3 WebAPI の必要性

ところで、先程 WebAPI はプログラミング API と言ったが、Web ページのフロントエンド側にも JavaScript が実装されているではないか、と思う人もいるだろう。
両者の違いは非常に簡単である。次の §1.3.4 で詳細は述べるが、今簡単に説明しておくプログラムが実行される場所が違い、それぞれにメリットがある、ということになる。
したがって、現在の Web においては、WebAPI は必要不可欠な技術である。

1.3.4 Ruby

Ruby は、日本人のまつもとゆきひろ氏が開発したプログラミング言語である。
バックエンド開発によく使われるが、「Ruby on Rails」というフレームワーク^{*8}を用いるとフロントエンドにも使える。
どちらかというと「楽しく書ける」言語であり、日本語ドキュメントも充実している。

^{*7} 入出力用の端子のようなもの。Web ページをプロジェクターとし、PC をプログラムとすると、PC をプロジェクターにケーブルでつなげることで、どこに何を表示するのか、プロジェクターに命令を出すことができる。

^{*8} 「枠組み」の意。プログラミング用のテンプレートのようなもの。

1.3.5 Node.JS

Node.JS は JavaScript をサーバー上^{*9}で動かせるようにしたもの。
JavaScript で簡潔にサーバーサイドのプログラムを書くことができる。
npm というパッケージマネージャーがあり、様々なライブラリがそこからインストールできる。

1.3.6 Java

定番中の定番。
上の 2 つに比べて記述が難しく、プログラムが長くなりやすいので最近あまり使われない。
「Java VM」上で動作するので OS に関係なく同じように動作する。
数年前までは Android 開発の公式言語として Google が推奨していた^{*10}

1.3.7 Python

なんだかんだで Python も動く。
Python は科学計算用途に強く、そのようなことをやるときは Python を用いることがある。
最近 Ruby に押され気味である。

^{*9} サーバー上で直接動かす環境のこと。普通の実行環境はブラウザ上の環境なのでサーバーでは直接動かせない。

^{*10} Java はライセンス関係が複雑で、最近商用ライセンスが有料化されたため、「Java 離れ」はより加速することが予測される。なお、現在 Google は Java の上位互換を目指して開発されている「Kotlin」を推奨している。なお、「Kotlin」のライセンス元はチェコの JetBrains 社である。