

最新 Web

- Part1 これだけは押さえておきたい
Webプログラミングの基礎知識 p.24
- コラム 手軽に導入できるサーバーサイドの開発環境
XAMPPをインストールしよう p.32
- Part2 書いて動かして覚える
Webプログラミング超入門 p.34
- Part3 Web標準の進化を押さえて
HTML 5の新機能を使ってみよう p.40
- Part4 デザインを統一するCSS
新版ではUIに特化した機能が多数 p.48
- Part5 フレームワークのsymfonyを使って
天気予報サイトを作成しよう p.56
- Part6 最新Ruby on Railsでハムスターと
チャットする人工無能を作成しよう p.66

まるごと大特集

プログラミング入門

IDEA①

インターネットには、検索サイトやブログ、掲示板、SNSなど、実に様々なWebアプリケーションが存在します。

こうしたWebアプリケーションを作成するには、コンテンツの仕様や通信の仕組み、各種プログラムの実装方法など、学ぶべきことが数多くあります。さらに、Web関連の技術は進歩し続けています。今日学んだ技術が明日使えるとは限りません。

そこで本特集では、「Webプログラミングをこれから始めたい」「最新のWeb関連技術を先取りしたい」プログラマやクリエイターを対象に、Webアプリケーションの基礎から実践までをまるごと紹介します。

HTMLやJavaScript、HTTPの基礎から、HTML 5、CSS3、MVCフレームワークまでと盛りだくさんですが、実際にコードを動かして楽しみながら理解していきましょう。

特集 Part1

これだけは押さえておきたい Webプログラミングの基礎知識

Part1では、Webプログラミングに不可欠な基礎知識を説明します。まずは、何を学べばよいのかを整理して、Webアプリケーションが動作する仕組みを理解しましょう。それからWebページを構成するHTMLを習得し、手軽に始められるJavaScriptでWebプログラミングの第一歩を踏み出しましょう。

安藤 正芳

Webサイト上でアプリケーションを提供する**Webアプリケーション**の重要性は年々高まっています。これから取り組むには、何から学べばよいのでしょうか？

入門者が混乱してしまいがちな理由の一つに、Webプログラミングが**様々な技術の集合体**で成り立っていて、どこから手を付けてよいのかわかりづらいことが挙げられます。

そこでPart1では、Webプログラミングの第一歩として、Webアプリケーションがどのようにして動作するのか、その仕組みを押さえてどこかポイントなのかを知るところから始めましょう。実際に体験しながら学んでいくと、普段使っているWebサイトの仕組みが見えてきて、とてもおもしろいですよ！

Webブラウザを使ってアクセス 一体何が起きている？

私たちが、普段から閲覧しているインターネット上のWebサイトの“裏側”では、様々なプログラムがWebアプリケーションとして動いています。Webサイトにアクセスするときの処理を解き明かして、学ぶべき事柄を整理してみま

しょう(図1)。

目的のWebサイトにアクセスするとき、まずはクライアントのWebブラウザを立ち上げてから、URL*¹を打ち込む、またはお気に入りからWebページを選択することでしょう。するとWebブラウザは、指定されたURLのWebサーバーに向けて、**HTTP (Hypertext Transfer Protocol)**という通信プロトコルに従った形式のリクエストを送信します。

リクエストを受けたWebサーバーは、そのリクエストに合わせて**Webサーバー上のプログラム (Webプログラム)**を動作させ、その実行結果をレスポンスとして、**HTML (HyperText Markup Language)**ファイルや画像ファイルをWebブラウザに返します。

レスポンスを受け取ったWebブラウザは、その内容を解析して、HTMLファイルで指定されたレイアウト通りに並べて表示するわけです。これが、Webプログラミングの基本的な動作になります。

ここまでくれば、もうおわかりでしょう。Webプログラミングに踏み出すために必要最低限の基礎知識は、HTML、Webブラウザ、Webプログラム、HTTPなどですね。では

早速、Webページの作成に欠かせないHTMLの仕様から見

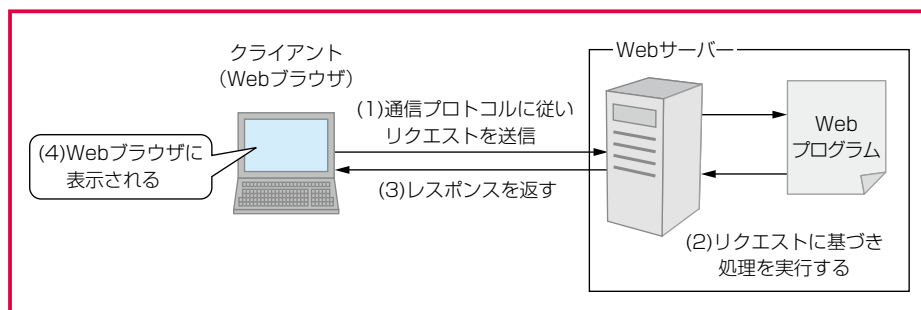


図1 ● 様々な技術が使われてWebブラウザに目的のサイトが表示される

*1 URLは、Uniform Resource Locatorの略。アクセス先の情報が、インターネット上のどの位置にあるかを指し示す表記方法のことです。

Ruby on Railsでハムスターとチャットする人工無能を作成しよう

このPart6では、Webアプリケーション・フレームワークのRuby on Railsを使って、「人工無能」のプログラミングに挑戦します。Rubyを知らなくても大丈夫。なぜなら、Rubyは直感的でわかりやすく、比較的少ないコードで記述できるからです。symfonyとはまた違った開発の楽しさを体験してみましょう。

伊尾木 将之

Part6では、注目を集めているプログラミング言語のRubyを使って、Webアプリケーションを実装してみます。

Rubyの標準機能だけでもWebプログラミングは可能ですが、様々な機能を自分で作り込まなければならず、結構な手間がかかります。そこで、雑多な作業を簡略化できるWebアプリケーション・フレームワーク、Ruby on Rails（以下、Rails）を活用しましょう。

Railsは、Webアプリケーションを手軽に作成するための機能を数多く備えており、少ないコードでサクサクとアプリケーションを開発できます。半面、機能が多いぶん、何が行われているか見えにくくなっている部分もあります。ここでは、Railsの込み入った話は避けて、開発を進めていきましょう。一通りの開発手順がわかれば、「Railsとは何か」が実感できるでしょう。

人工無能ってなんですか？

それでは、Webアプリケーションを作成していきましょう。今回は、「人工『超』無能」を作ります！

おっと、ここでいきなり、人工（超）無能という言葉が登場しました。人工知能ではなく、人工無能です。皆さん、人工無能はご存知ですか。

人工無能とは、「人工知能のようにきちんと動くわけではないが、人間のように（一応）応答してくれるアプリケーション」のことです。最近では、あたかも人間のようにチャットする人工無能が数多くあります。また、日記に人間らしいコメントを返す人工無能もあります。

今回はRailsを使って、人工無能を備えたチャット機能を作ります。人工無能は、人間のように思考しているわけではなく、あくまでも「人間らしく」みせるだけのアプリケーションです。

人工無能とチャットをしていると、思い切り意味不明な発言を返すことがあります。そこが、なんとも言い難い人工無能のおもしろさなのです。

今回の記事でも、人工無能を知的にするつもりは全くありません。逆に、人工無能のさらに無能版というわけで、「人工『超』無能なチャット・アプリケーション」を作しましょう*1。

ハムスターと楽しく会話しよう

まずは、人工『超』無能なチャット・アプリケーションのイメージを考えましょう。ただの人工無能と会話をして、少し味気ないので、ハムスターと会話してみます（図1）。ハムスターといっても、実際は画像があるだけです…。ここ

*1 通常の人工無能では、日本語を解析するために形態素解析という処理を施します。形態素解析とは、日本語の文の構造を調べて、名詞や動詞の候補を解析する処理のことです。形態素解析は非常に面白い処理なのです

が、やや難解のため、今回は対応しません。