

pixivエンジニアが 教えるプログラミング 入門

金子達哉

**プログラム、
まずは一行
書いてみよう!**

**pixivエンジニアが教える
プログラミング入門**

金子達哉

星海社

63



はじめに

はじめまして。ピクシブ株式会社でエンジニアをやっています、金子と申します。本書を手にとってください、ありがとうございます。

この本では、「画像投稿掲示板を作ってみる」という、シンプルな作業を通して、プログラミングの基礎を学んで頂きたいと思っています。

本書は、エンジニアでない人のためのプログラミング入門書です。例えば次のような方にオススメをします。

- ・システムエンジニアやプログラマーとプロジェクトを共にする、Web ディレクター／デザイナーの方
- ・プログラミングに興味はありつつも、なんとなく難しそうでは始めるきっかけが掴めない方（10年前の筆者）
- ・就職／転職の結果、全くの未経験からシステムエンジニアになることになった方
- ・プログラミングによって何か動くものが作ってみたい方

本書に書かれていることを実践して頂き、見事画像投稿掲示板を作りあげることができれば、上記の方々が直面している課題のいくつかを解決することができるでしょう。

本書は、私が社内で行った新卒社員向けの研修（非エンジニアの新卒にプログラミング研修を行いました [pixiv engineering blog http://inside.pixiv.net/entry/2014/04/28/140859](http://inside.pixiv.net/entry/2014/04/28/140859)）が元になっています。研修は、コード（コンピュータに指示を出すための言語）を1行も書いたことのない、全くの素人に向けた、6日間の集

中講義でした。

英語で言えば、アルファベットも知らないような状態から、英作文ができるように仕上げなさいという話で、大変骨の折れる仕事でした。おかげで、普段何気なく使っている言葉やテクニック、概念について考え直すことができました。書籍化にあたり、研修の過程で受講者から出た疑問に答えるのはもちろん、詰まりがちな部分についても、丁寧な補足を加えています。相当「実践的」な内容になっているはずです。

非エンジニアの方々に読んで頂くにあたり、「分かりやすさ」にも徹底してこだわりました。パラパラとめくって頂くとコードが多数目につき、取っつきにくい印象を持たれるかもしれませんが、ご安心を。順番に読んでもらえれば、100%理解して頂けるように書いてあります。

講義に入る前に、大事な言葉を紹介したいと思います。

「プログラムの三大美德」と呼ばれるもので、Perl というプログラミング言語を作ったラリー・ウォールの名言です。

- ・怠 惰 (Laziness)
- ・短 気 (Impatience)
- ・傲 慢 (Hubris)

これだけを見るとプログラミングをする人の人間性が疑われるかもしれません。

しかしこれはプログラミングに必要な心得を非常に端的に表しています。

怠惰とは面倒だと思ったら自動化をしたい。

短気とは気が利かないソフトウェアを見るとイライラして直そうとする。

傲慢とは文句をいう余地のない完璧なプログラムを書く。

いずれも、プログラミングをする上で重要な心得ですのでぜひ覚えてください。

なお本書の内容をサポートする Web サイトとして

https://github.com/catatsuy/support_programming を用意しました。合わせて、ぜひご活用ください。

それではさっそく、講義をはじめましょう。

PC (できれば Mac) に電源を入れ、ページをめくってください。

目次

はじめに 2

第1章

Webサイト作りの基本のキ 11

サイトはみるものじゃなくて作るもの 12

HTMLとCSSってなに? 13

CSSの書き方 17

ひとまずエディタをインストールしよう 20

第2章

Macを使った開発環境構築 25

なぜWindowsではダメなのか 26

ターミナルの操作を覚えよう 29

Macで開発が出来るようにする 32

〈Git/GitHubの使い方〉 33

Gitがなぜ必要なのか	33
GitHubを使ってみる	34
Gitの便利な設定	35
Gitの流れ	37
〈Ruby環境をインストールする〉	45
Rubyってなに？	45
〈Rubyの書き方を覚えよう〉	46
実行の仕方	47
書き方紹介	48
p	48
コメントアウト	48
変数 (Variable)	49
配列・ハッシュ (Array/Hash)	51
if	53
each	54
〈SQLiteとデータベースの基礎〉	55

SQLiteの使い方 57

SQLiteの起動方法 58

テーブルの作成・確認・削除 58

〈まとめ〉 61

レコードの追加・確認・更新・削除 61

〈まとめ〉 65

RubyからSQLiteを操作する 65

第3章

実際に作る画像投稿掲示板の仕様を決める 71

画像投稿掲示板をつくるのは結構大変 72

初心者向けに仕様を決めていく 73

第4章

安全なWebアプリケーションを作る 77

XSS 78

SQLインジェクション 80

第5章

実際にコードを書いて画像投稿掲示板を作ろう 85

GitHubでリポジトリをForkする 86

Sinatraの使い方 87

投稿ができるようにする 93

投稿をデータベースに保存する 96

投稿一覧を表示する 100

画像を投稿できるようにする 104

投稿にスターを付けられるようにする 121

JavaScriptを覚えよう 127

jQueryを使ってみる 136

Ajaxを使って非同期に投稿にスターを 138
付けられるようにする

おわりに 153

第1章

Web サイト作りの基本のキ

サイトはみるものじゃなくて作るもの

皆さんは普段 Google Chrome (以下 Chrome) や Internet Explorer などのウェブブラウザやブラウザと呼ばれるアプリケーションを使い様々な Web サービスを使っていると思います。

少なくとも現在の日本においてはインターネットなしでの生活は考えられず、日々様々な Web サイトを閲覧していることと思います。

これらの Web サイトはどうやって作られているのでしょうか。

実は簡単に見ることができます。

まず Chrome をインストールしていなければ Chrome をインストールしてください。

Chrome で www.pixiv.net を開いてみてください。

そしてウィンドウ上部にあるメニューから『表示』を選択して『開発 / 管理』から『デベロッパー ツール』を選びます。

すると pixiv のソースコードを見ることができます。

これは Chrome の Developer Tools という機能です。

Chrome の Developer Tools は非常に機能が充実しており、開発に大変役立ちます。

ここで表示されているのは HTML と呼ばれるものです。

今は何が書かれているか分からないかもしれませんが、本書を読み終わったころには、何が書かれているかほとんど分かるようになるでしょう。

HTMLとCSSってなに？

プログラミングについて学ぶにあたり、まず HTML と CSS について覚える必要があります。

文章を書く際に Microsoft Word や iWork の Pages などのワープロソフトを使ったことがある人は多いと思います。ワープロソフトで文章を書く際、文字に様々な装飾をすることがありませんか？

例えば最初にタイトルを書いたらそれを中央揃えにし、大きくするなどの装飾です。

ワープロソフト上では明確な区別なく行えるこの装飾、ブラウザでは異なります。

ブラウザでコンテンツを表示するには文章と文章の構造を HTML で、文章自体に関係のない見た目に関わる部分を CSS で記述します。

まずは HTML から説明します。

何はともあれ実際の HTML を見てもらいましょう。

例えば `index.html` のようなファイルです。

HTML の拡張子 (ファイル名の最後につく、ファイルの種類を規定する部分) は、`html` となります。

```
<!DOCTYPE html>
<html lang="ja">
  <head>
    <meta charset="UTF-8">
```

```
<title> タイトルをここに書く </title>
</head>
<body>
  <h1> 見出し 1</h1>
  <p> ここは段落 </p>
  <h2> 見出し 2</h2>
  <p> ここの段落 </p>
  <ul>
    <li> 箇条書き </li>
    <li> 箇条書き </li>
  </ul>
  <h3> 見出し 3</h3>
  <p> 段落 </p>
  <h3> 見出し 3</h3>
  <p>
    段落 <br>
    段落段落
  </p>
</body>
</html>
```

HTML は最初に `<!DOCTYPE html>` と書きます。

これはこのファイルが HTML であることを示すものなので必ず書きましょう。

HTML をはじめ、コンピュータやインターネット関係の

もののほとんどはアメリカで作られたため、英語がベースになっています。

HTMLは基本的に<html></html>のように<OO>と</OO>でテキストを囲います。

これをHTMLタグといいます。HTMLタグは開始タグ<OO>と終了タグ</OO>を使って書きます。

開始タグには属性というものを指定できます。日本語を扱う文章ならhtmlタグの属性に<html lang="ja">と書きます。

このように属性は開始タグの中に属性="値"のように指定します。

今回のlang="ja"はlanguageとJapaneseの略で『言語は日本語にする』という意味になります。

このように属性や値には英単語を略しているものが多いです。

HTMLの文章は<html>タグの中に書きます。

<html>タグの中に<head>タグと<body>タグを書きます。<head>タグの中にはこの文章のタイトルなどの情報を書きます。

詳しくは後で説明します。

<body>タグの中には実際に表示したい文章を書きます。この中に書かれているものはブラウザ上で表示されます。

<h1>はheadingの略で見出しのことで番号は順番を表します。

番号は <h1> から <h6> まであります。

文章自体は <p> タグで囲います。これは paragraph の略で段落を表します。

箇条書きをするには タグで囲い、箇条する内容をそれぞれ タグで囲います。

それぞれ unordered list と list item の略です。

1,2,3……のように順番を表示したい場合は ordered list ですので タグで囲います。

改行したい場合は
 タグを使います。普通に改行しただけでは改行にならないので気をつけてください。

 タグは改行を意味する Line Break の略です。

また HTML で非常によく使われるリンクを作るタグの <a> タグも紹介します。

このタグは Anchor の略で『アンカータグ』と呼ばれます。<a> タグは以下のように使います。

```
<a href="http://www.pixiv.net/"> イラストコミュニケーションサービス [pixiv (ピクシブ)]</a>
```

<a> タグには href 属性を使ってリンク先の URL を必ず書く必要があります。

このような必ず与えないといけない属性のことを必須属性と言います。

必須属性を持っているタグは他にもいくつかありますのでその都度紹介していきます。

HTML タグはこれだけではありません。

今後必要に応じて紹介していきます。

CSS の書き方

CSS は HTML とは異なり文章の見た目を担当します。

CSS は HTML の中に直接書くこともできますが、ファイルを分けた方が管理や編集が楽です。

実際の開発では HTML の中に直接 CSS を書くことはまずありません。

まず HTML の <head> タグの中に以下のように書きます。

```
<link rel="stylesheet" type="text/css"
href="style.css">
```

続いて HTML ファイルを保存しているフォルダに style.css というファイル名でファイルを作ります。

CSS の設定はこの style.css に書いて保存すれば適用されます。

CSS で設定する見た目などのことをスタイルと呼びます。そのため CSS を適用して見た目を変えることを『スタイルを適用する』『スタイルを当てる』などと表現します。

では実際の CSS の書き方を紹介します。

先程の HTML で例えば <p> タグの文字の色を赤色にしてみます。

```
p { color: red; }
```

これで <p> タグの文字の色は赤色になります。

この時に **color** のことをプロパティと呼び、**red** のことを値と呼びます。

さらにフォントのサイズを 30px にしたい場合は以下のようにします。

```
p {
```

```
color: red;
font-size: 30px;
}
```

ブラウザ上でサイズを指定する時はほとんどの場合でピクセルという単位が使われます。

ピクセルは px と表され、1px はディスプレイの1つのドットの大きさの事です。

このように設定したいタグに対して CSS の設定を列挙していきます。

しかしこのようにタグで設定するとそのタグ全てに設定が適用されてしまいます。

実際には特定の HTML 要素に対して CSS を設定したいことの方が多くでしょう。

そういった時に使われるのが id と class です。

以下のように指定します。

```
<p id="sample-id"></p>
<p class="sample-class"></p>
```

id は最初に # を、class は最初に . を付けて CSS を指定します。

```
#sample-id {  
    color: blue;  
}  
  
.sample-class {  
    color: green;  
}
```

id と class の違いは id が同じ HTML ファイル内に1つしかあってはならないのに対して class はいくつあっても良い点です。

CSS の指定には class の方が使い勝手が良いので常に class を使うことをおすすめします。

ひとまずエディタをインストールしよう

Web アプリケーションを作る上でこれからプログラミングを行っていきます。

プログラミングはエディタと呼ばれる種類のアプリケーションを使って行いますのでエディタをインストールする必要があります。

エディタといっても特別なアプリケーションではありません。

代表的なエディタとして Windows のメモ帳や Mac のテキストエディットがあります。

これらのエディタならば最初からインストールされているのでわざわざインストールをする必要はありませんし、これらのエディタでもプログラミングを行うことができます。しかしこれらの最初からインストールされているエディタはプログラミング専用ではないので様々な機能が不足しています。

不足している機能をいくつか紹介します。

これから紹介していきますがプログラミングは、分かりやすいようにスペースを適宜挿入します。

こういったスペースのことをインデントと呼びますが、インデントの挿入を自分で行うことはとても面倒です。

また実際にプログラミングを行う際、プログラムに色が付いていないと種類の判別が難しく、プログラミングが困難なものになります。

インデントを調整する機能があり、プログラムの文字に色を付けてくれるエディタを使用するのはプログラミングにおいて必須です。

インデントを調整する機能があり、プログラムに色を付け

てくれるエディタは色々ありますが、今回は使い方がテキストエディットなどと似ていてなじみやすく、また機能が充実しており、最近プログラマの間で人気が急上昇している Sublime Text をおすすめします。

Sublime Text には先程紹介したインデントを調整する機能や色を付けてくれる機能の他にもコードの一部を書くと残りの部分を補完してくれる機能などがあります。

基本的にプログラミングは慣れたエンジニアでもスペルミスなど必ずミスをします。

また長い名前を一字一句全て覚えるというのは現実的ではありません。

しかしプログラムというのは厳密なルールがあり、基本的に一字一句合っていなければ正しく動作しません。

そこで Sublime Text の補完機能を使うことでミスを減らすことができます。

そのため補完機能を活用することをおすすめします。

補完機能を使うのに追加の設定などはいりません。

Sublime Text は以下のサイトからダウンロード出来ます。

<http://www.sublimetext.com/3>

通常のアプリと同じようにインストールしてください。

もし『開発元が未確認のため開けません。』という警告が出てインストールができない場合はリンゴマークから『システム環境設定』を開いて『セキュリティとプライバシー』を開きます。

そして『一般』のタブから『ダウンロードしたアプリケーションの実行許可』で『全てのアプリケーションを許可』を選択してください。

変更にはパスワードが必要です。

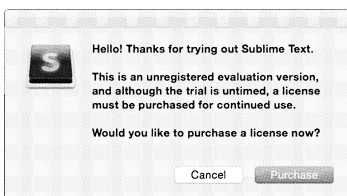
Sublime Text はファイルの拡張子からインデントや色を付けてくれますが、保存する前のファイルなど拡張子を指定する前でもそういった機能を使うことができます。

右下から様々なプログラミング言語を選択することができます。たくさんありすぎて迷うと思いますが、例えば HTML を選択すれば HTML を書くことができます。

他にもいくつか覚えて欲しい機能がありますのでその都度解説します。

また何度かファイルを保存すると有料版購入を勧めるポップアップが出てきます。

Cancel を入力すれば使い続けることができますが、長く使うのであれば購入しましょう。



第2章

Macを使った開発環境構築

なぜ Windows ではダメなのか

Windows は OS の一種です。

OS は Operating System の略で日本語では基本ソフトと訳されます。

OS はコンピュータの中でハードウェアとアプリケーションの仲介を行います。

もう少し分かりやすく説明します。

例えばコンピュータを起動して Chrome を起動して pixiv を見ようとしています。

コンピュータは様々な企業が発売していますが、どのコンピュータでも Windows が入っていればインターネットに繋げることができて Windows 版の Chrome は動かすことができます。

しかしどのコンピュータでも中に使われている部品等が異なります。

またコンピュータは精密機器ですのでコンピュータの上で Chrome のようなアプリケーションを動かすには様々な部品の制御が必要になります。

一時的に記憶をするメモリや計算をする CPU やファイルなどを保存する HDD や SSD などが代表的な制御しなければならないハードウェアの一部です。

そのような様々なハードウェアの制御を引き受けて微妙な差異も吸収するのが OS の役割です。

OS がハードウェアを制御するのでアプリケーション側は OS の機能を使ってメモリや CPU や HDD や SSD などを操作します。

こうすることで OS が全てのアプリケーションがやりたい操作をまとめて効率よくハードウェアの制御を行います。そうすることでアプリケーションはハードウェアの制御を考慮することなくハードウェアを実質的に扱うことができます。

代表的な OS が Windows です。

OS は Windows だけではありません。最も代表的な OS である UNIX の紹介をします。

UNIX はアメリカ最大手の通信業者である AT&T のベル研究所で開発された OS です。

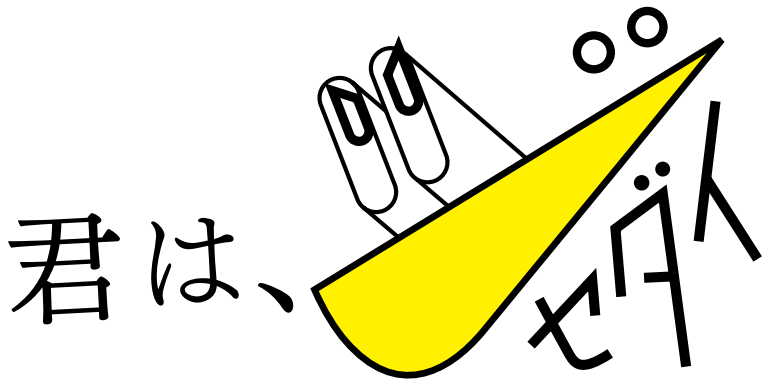
1970年代に広く普及した歴史のあるシステムで、主に大学や研究機関で使われてきました。

現在でも UNIX は事実上の標準になっています。

Mac OS X は UNIX の一種です。

Linux は商標の関係で UNIX とは名乗れませんが、事実上 UNIX の一種と考えて問題ありません。

今回学ぶ Web プログラミングは基本的に UNIX で動かすことが前提になっているので UNIX 系の OS を使用する必要があります。



君は、

ジセダイ

何と闘うか？

<http://ji-sedai.jp/>

「ジセダイ」は、20代以下の若者に向けた、**行動機会提案サイト**です。読む→考える→行動する。このサイクルを、困難な時代にあっても前向きに自分の人生を切り開いていこうとする次世代の人間に向けて提供し続けます。

**メインコンテンツ
イベント**

著者に会える、同世代と話せるイベントを毎月開催中！ 行動機会提案サイトの真骨頂です！

ニッポンのスタートアップ

3年後に再会することを約束して行う、未来アポ付きスタートアップインタビュー！

ジセダイジェネレーションズU-25

彼らはどうやって「闘う相手」を見つけたのか。各界の超新星に、その軌跡と未来を聴く。

マーカー部分をクリックして、「ジセダイ」をチェック!!!

行動せよ!!!