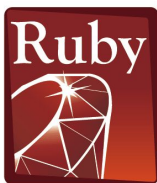


# Web開発の 全体像を知ろう！

このスライドでは、  
**DMM WEBCAMP COMMITで学ぶWeb開発技術の全体像**  
をご説明していきます！

# 学習フェーズ I で学ぶもの

## Webアプリケーション制作



## ソース管理



## 目次

1. 私たちが普段目になっているもの
2. Webサイトが表示されるまで
3. ソースコードの保管庫
4. ちなみにシリーズ

私たちが普段目に行っているもの

# ブラウザとは？

「**ブラウザ**（Webブラウザ）」という言葉

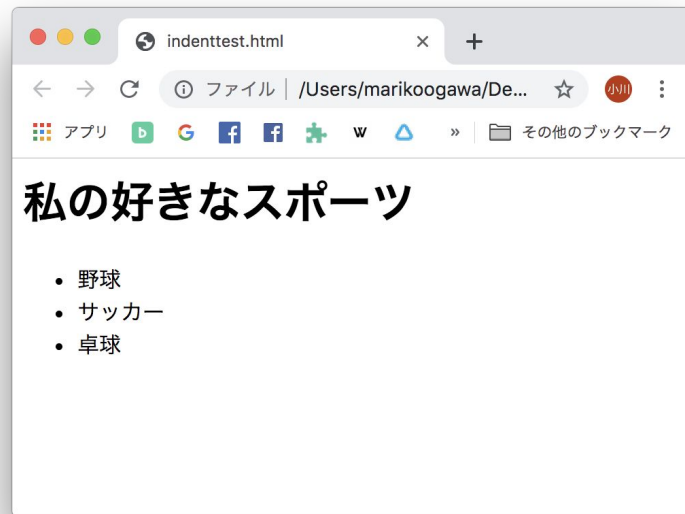
聞いたことがありますか？

聞いたことがある人は、それが何か説明できますか？

# ブラウザとは？

ブラウザは、Google ChromeやSafari、Firefox、Internet Explorer などいろいろありますが、共通して  
「HTMLファイルを読んで画面表示する」という役割を持ちます。

```
1  <html>
2    <head>
3      <title></title>
4    </head>
5    <body>
6      <div>
7        <h1>私の好きなスポーツ</h1>
8        <ul>
9          <li>野球</li>
10         <li>サッカー</li>
11         <li>卓球</li>
12       </ul>
13     </div>
14   </body>
15 </html>>
16
```

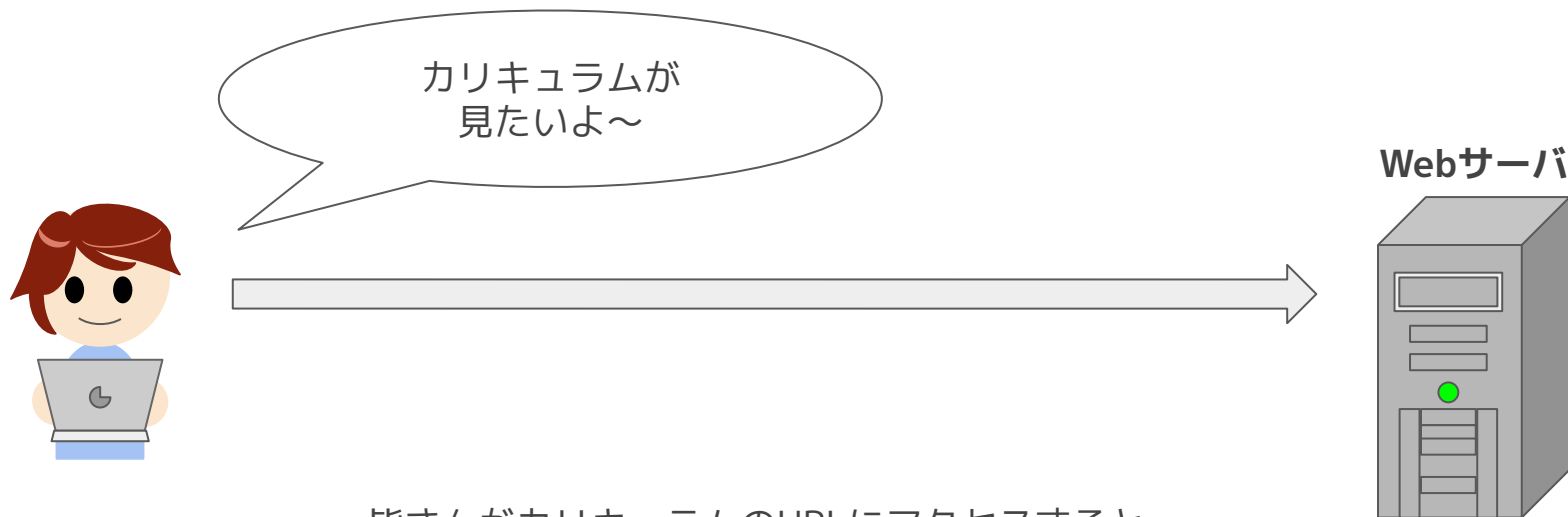


# ブラウザとは？

皆さんがDMM WEBCAMP COMMITの  
カリキュラムのURLにアクセスした場合を例に、  
ブラウザが何をやっているのかを  
説明していきます！



# ブラウザとは？

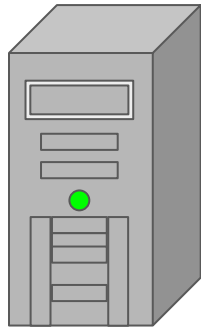


皆さんがカリキュラムのURLにアクセスすると、  
「Webサーバ」と呼ばれるサーバ（詳細は後述します）に  
その要求が送られます。

# ブラウザとは？

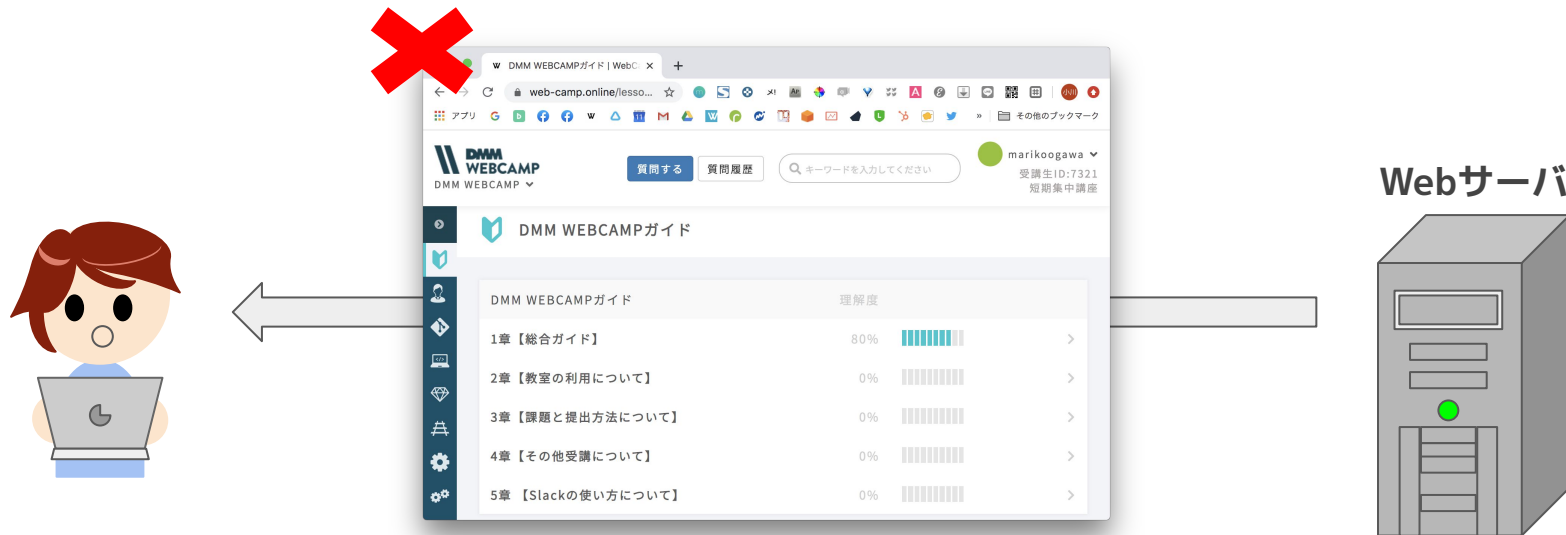


Webサーバ



Webサーバはその要求に対して  
ページを返却してくれるのですが...

# ブラウザとは？



厳密には、この画は違います。

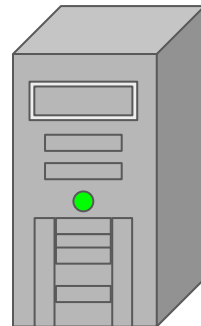
ということかという...

# ブラウザとは？



```
<!DOCTYPE html>
<!--[if IE 8]>      <html class="ie ie8"> <![endif]-->
<!--[if IE 9]>      <html class="ie ie9"> <![endif]-->
<!--[if gt IE 9]><!--> <html> <!--<![endif]-->
<head>
  <meta charset="utf-8" />
  <title>DMM WEBCAMPガイド | WebCamp オンライン</title>
  <meta name="csrf-param" content="authenticity_token" />
  <meta name="csrf-token" content="LSX7DUV8SLQ0Bku97vEmLfa0II8U+kVoBso21yyx4SsAr1zw5pRk
gz50qor0SuWlZK9uVCoDzJWapqV3LkkKtg==" />
  <link rel="stylesheet" media="all" href="/assets/
application-8fa614edbfaf0cceba257f7ca461003a699441d4b80c44b8681ef77e840e6e83.css"
data-turbolinks-track="reload" />
  <link rel="shortcut icon" type="image/x-icon" href="/assets/
favicon-82c42a74d436158dc1e3038680b33730ea53379c13372e2ee07c0fbf86c914d0.ico" />
  <meta name="keywords" content="" />
  <meta name="description" content="" />
  <!-- mobile settings -->
  <meta name="viewport" content="width=device-width, maximum-scale=1,
initial-scale=1, user-scalable=0" />
  <link rel="stylesheet" href="/public/assets/css/layouts/simplebar.css" type="text
/css">
  <link rel="stylesheet" href="/public/assets/css/layouts/fonts/font.css" type="
text/css">
  <link href="https://use.fontawesome.com/releases/v5.6.1/css/all.css" rel="
stylesheet">
  <link href="https://cdnjs.cloudflare.com/ajax/libs/toastr.js/latest/css/
toastr.min.css" rel="stylesheet"/>
```

Webサーバ



HTMLファイルが返ってきます

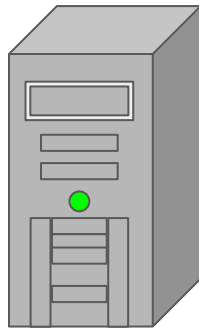
# ブラウザとは？

ブラウザがHTMLを  
私たちが見やすいように  
変換してくれている！



```
<!DOCTYPE html>
<html class="no-js">
  <head>
    <meta charset="utf-8" />
    <title>DMW WEBCAMPガイド | WebCamp オンライン/</title>
    <meta name="csrf-param" content="authenticity_token" />
    <meta name="csrf-token" content="LSX7DUV8SL008Ku97vEhLfa0IIBuKVo8so2lyx4SArizw5pK
    g55Qqo05UML2X9UC02J3aggyZLAKK" />
    <link rel="stylesheet" media="all" href="/assets/
    application-8fa614edbfafecbeba25777ca651003a699441d4b88c44b8681ef77e846e6e83.css"
    data-turbolinks-track="reload" />
    <link rel="shortcut icon" type="image/x-icon" href="/assets/
    favicon-02c42a74d4361586c1e389808083738e53379c13372e2ee7c8fbf86c914d0.ico" />
    <meta name="viewport" content="width=device-width, maximum-scale=1,
    initial-scale=1, user-scalable=0" />
    <link rel="stylesheet" href="/public/assets/css/layouts/simplebar.css" type="text
    /css"/>
    <link rel="stylesheet" href="/public/assets/css/layouts/fonts/font.css" type="
    text/css"/>
    <link href="https://use.fontawesome.com/releases/v5.6.1/css/all.css" rel="
    stylesheet"/>
    <link href="https://cdnjs.cloudflare.com/ajax/libs/toastr.js/latest/css/
    toastr.min.css" rel="stylesheet"/>
  </head>
  <body>
    <div class="container">
      <div class="row">
        <div class="col-md-12">
          <div class="page-header">
            <h1>DMW WEBCAMPガイド</h1>
          </div>
          <div class="list-group">
            <div class="list-item">
              <a href="#">1章【総合ガイド】</a>
            </div>
            <div class="list-item">
              <a href="#">2章【作業の活用について】</a>
            </div>
            <div class="list-item">
              <a href="#">3章【課題と提出方法について】</a>
            </div>
            <div class="list-item">
              <a href="#">4章【その他作業について】</a>
            </div>
            <div class="list-item">
              <a href="#">5章【Slackの使い方について】</a>
            </div>
          </div>
        </div>
      </div>
    </div>
  </body>
</html>
```

Webサーバ



# 他にも...

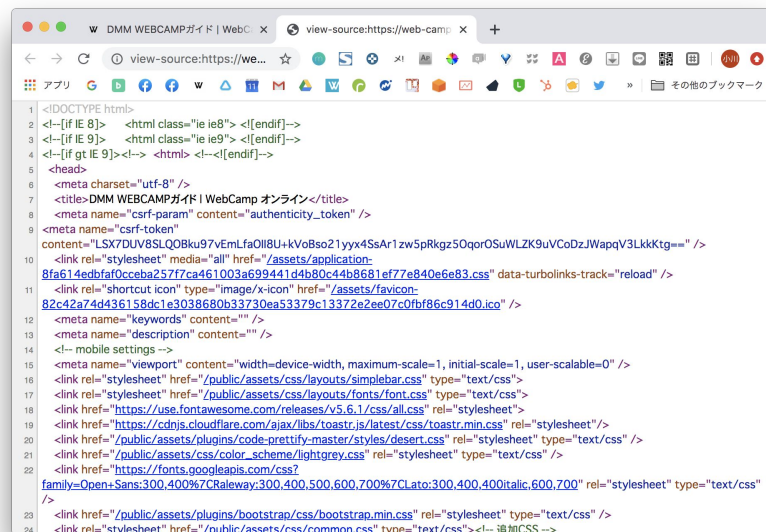
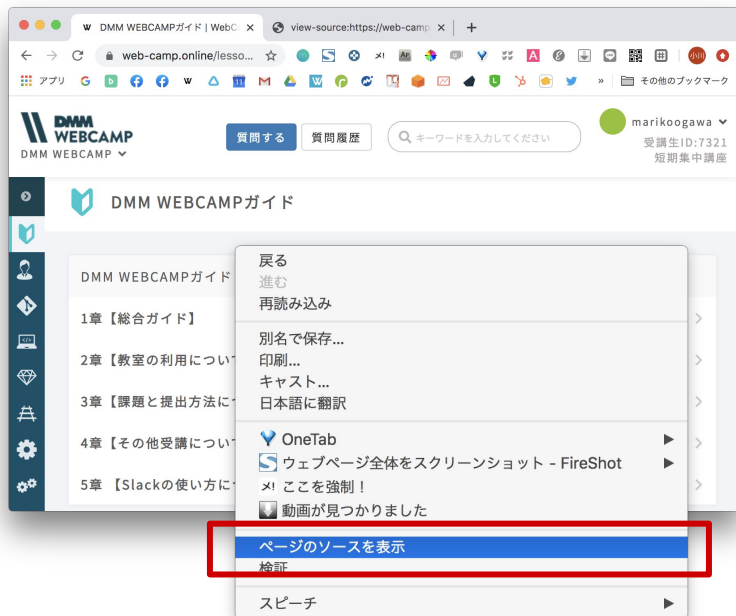
ブラウザにはサイトの閲覧機能だけではなく、

- お気に入り（ブックマーク）機能
- 検証機能
- 拡張機能

など、様々な機能があります。

# HTMLファイル

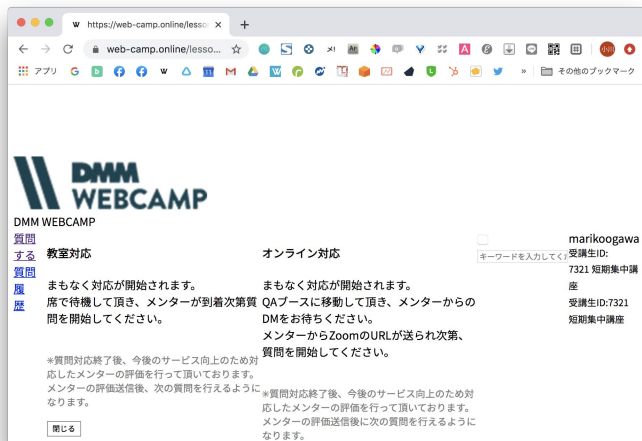
ブラウザ上で右クリック>[ページのソースを表示]で  
普段利用しているサイトのHTMLを見ることができます！



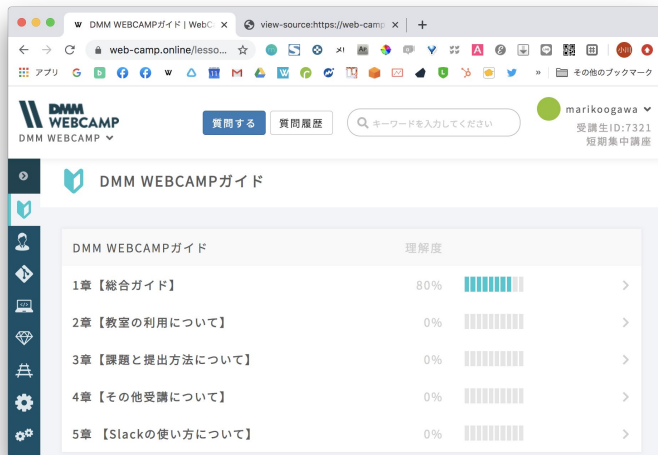
# CSSとは？

CSSは、ページの装飾をするためのファイルです。  
HTMLに記載するのは骨組みだけです。

## CSS適用なし



## CSS適用あり



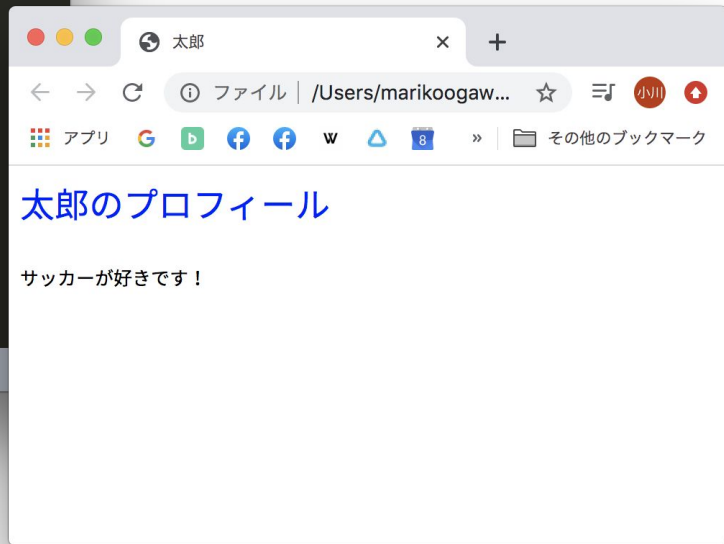


# なんでCSSが必要なの？



```
1 <html xmlns="http://www.w3.org/1999/xhtml" lang="ja">
2 <head>
3   <title>太郎</title>
4 </head>
5 <body>
6   <font size="5" color="blue">太郎のプロフィール</font>
7   <font size="2" color="black">サッカーが大好きです！</font>
8 </body>
9 </html>
```

Line 7, Column 48; Saved ~/Desktop/ウォーミングアップ/taro.html (UTF-8)



文字装飾等の記述は、HTML内にも書けます。それではなぜCSSファイルが必要なのでしょう？

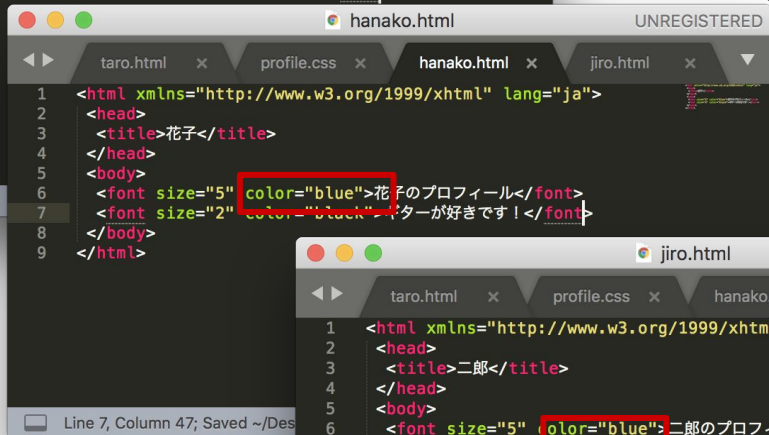
# なんでCSSが必要なの？



tarо.html

```
1 <html xmlns="http://www.w3.org/1999/xhtml" lang="ja">
2 <head>
3 <title>太郎</title>
4 </head>
5 <body>
6 <font size="5" color="blue">太郎のプロフィール</font>
7 <font size="2" color="black">サッカーが好きです！</font>
8 </body>
9 </html>
```

Line 7, Column 48



hanako.html

```
1 <html xmlns="http://www.w3.org/1999/xhtml" lang="ja">
2 <head>
3 <title>花子</title>
4 </head>
5 <body>
6 <font size="5" color="blue">花子のプロフィール</font>
7 <font size="2" color="black">サッカーが好きです！</font>
8 </body>
9 </html>
```

Line 7, Column 47; Saved ~/Des



jiro.html

```
1 <html xmlns="http://www.w3.org/1999/xhtml" lang="ja">
2 <head>
3 <title>二郎</title>
4 </head>
5 <body>
6 <font size="5" color="blue">二郎のプロフィール</font>
7 <font size="2" color="black">グルメが好きです！</font>
8 </body>
9 </html>
```

Line 7, Column 47 Tab Size: 4 HTML

文字色を緑にしたい！と思った時、  
該当のhtmlファイルを全て編集しなくてはなりません！

3ファイルならまだまし  
だけど、1,000ファイル  
とかあったら...

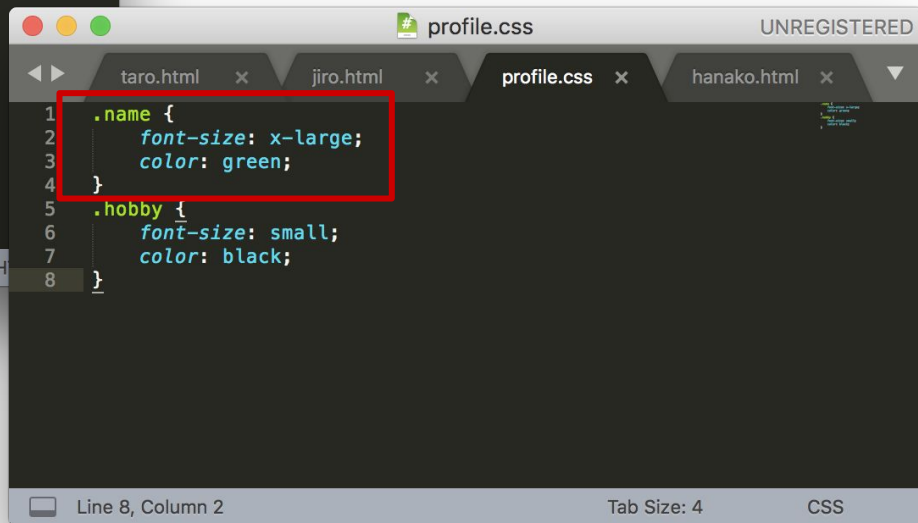
# なんでCSSが必要なの？

CSSを読み込めば  
設定しているファイル全てに  
反映される！



```
1 <html xmlns="http://www.w3.org/1999/xhtml" lang="ja">
2 <head>
3 <title>二郎</title>
4 <link rel="stylesheet" type="text/css" href="profile.css">
5 </head>
6 <body>
7 <div class="name">二郎のプロフィール</div>
8 <div class="hobby">グルメが好きです！</div>
9 </body>
10 </html>
```

Line 10, Column 8      Tab Size: 4



```
1 .name {
2   font-size: x-large;
3   color: green;
4 }
5 .hobby {
6   font-size: small;
7   color: black;
8 }
```

Line 8, Column 2      Tab Size: 4      CSS

# CSSは洋服のようなもの！

HTML



CSS適用



ブラウザ上で起こっていることを  
「**フロントエンド**」と呼びます。

直接ユーザの目に触れる部分のことです。

**Webサイトが表示されるまで**

# これから自分が携わっていくものの仕組み

みなさんはDMM WEBCAMP COMMITで  
WEBアプリケーション開発について学びます。

学習を進める上で、

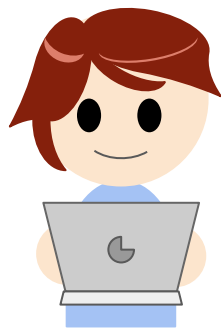
**「どのような仕組みでWEBサイトが表示されるのか」**

**「今どの部分を学んでいるのか」**

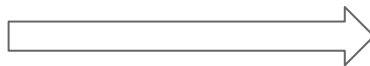
を理解しておくことは、非常に重要です。

# サイトが表示されるまで

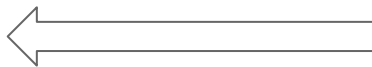
DWCのfacebook  
チェックしよっと



URLにアクセスする



表示される



DMM WEBCAMPの  
facebookページにアクセスした時を例に...



<https://www.facebook.com/DMMWEBCAMP/>



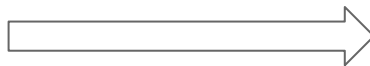
# サイトが表示されるまで

DWCのfacebook  
チェックしよっと

ここでなにが起きているかを  
解説します！！



URLにアクセスする

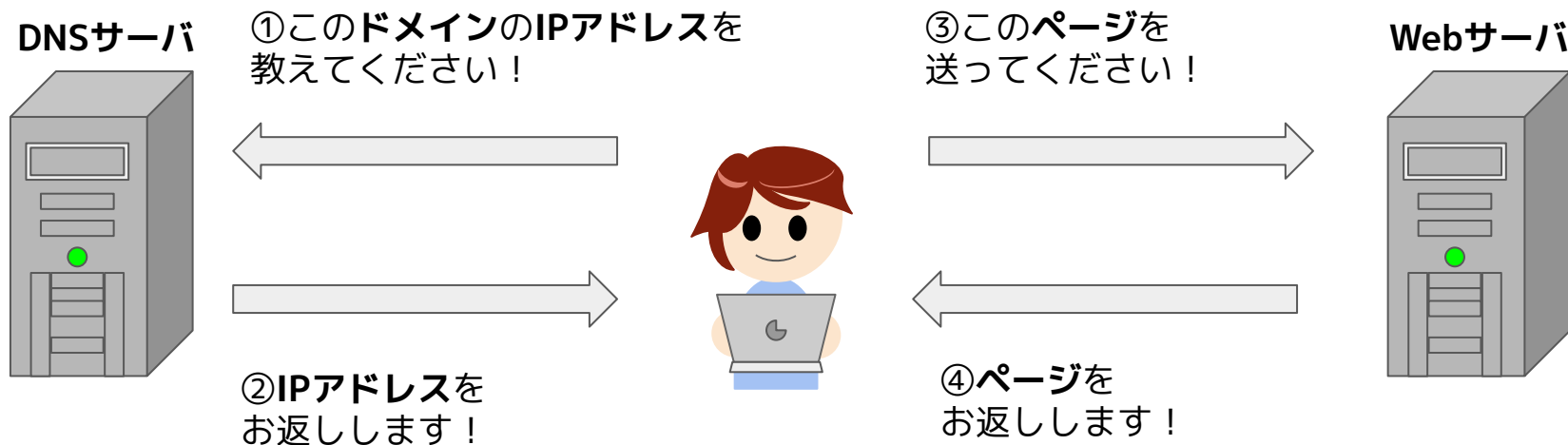


表示される



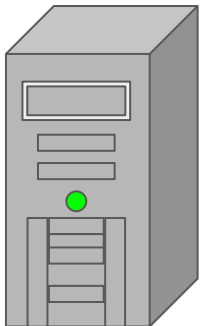
# Web表示の全体像

全体像はこんな感じです。ひとつずつ見ていきましょう！

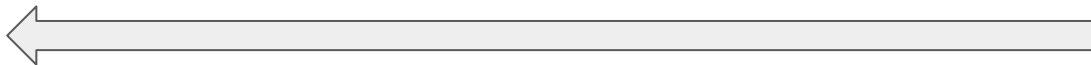


# ①DNSサーバにアクセス

DNSサーバ



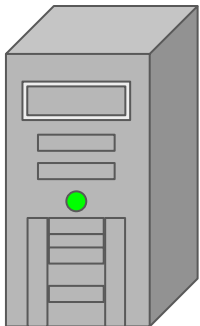
①このドメインのIPアドレスを教えてください！



先ほど、皆さんがURLにアクセスすると  
「Webサーバ」と呼ばれるサーバにその要求が送られる、と述べましたが、  
正確には最初にDNSサーバに要求が飛んでいます。

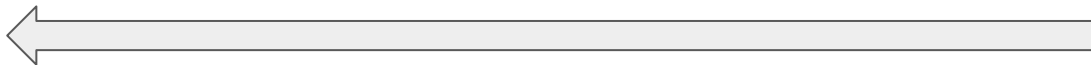
# ①DNSサーバにアクセス

DNSサーバ



なにそれ？

①この**ドメイン**の**IPアドレス**を教えてください！



# ドメインとIPアドレス

`https://www.facebook.com/DMMWEBCAMP/`

||

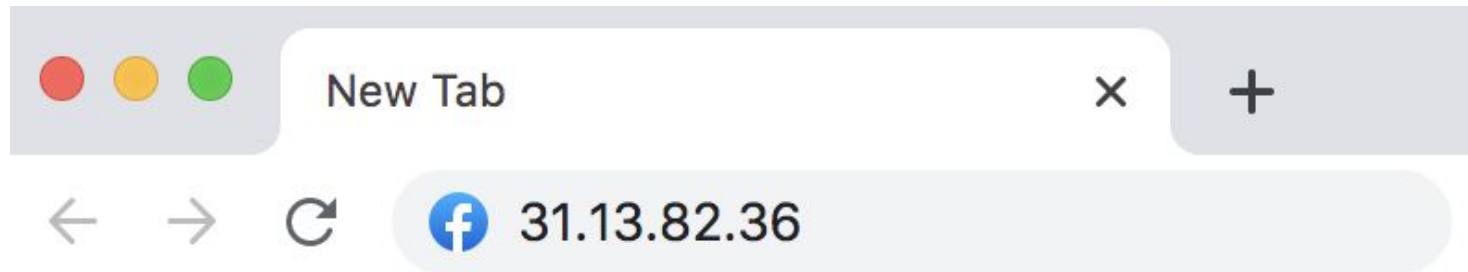
**31.13.82.36**

IPアドレスは、簡単にいうと「コンピュータの名前」です。数字で表されます。

facebookなら、31.13.82.36がIPアドレスです。

IPアドレスはサイトにアクセスするために必要な値ですが（後述）、いちいちIPアドレスを指定してアクセスするのは大変なので、「www.facebook.com」のような「一目でfacebookだ！」と分かるあだ名（ドメイン）をつけている、という感じです。

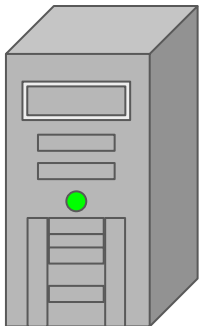
実験してみよう！！



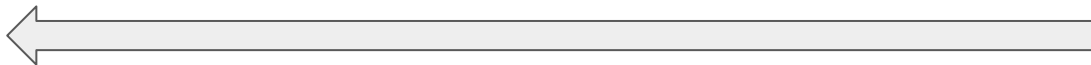
試しに、ブラウザのアドレスバーに「31.13.82.36」と打って  
アクセスしてみてください！

# ①DNSサーバにアクセス

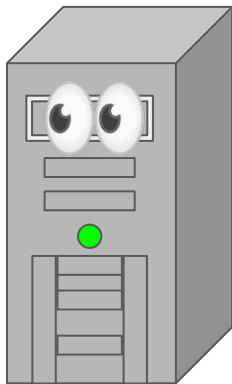
DNSサーバ



①「**www.facebook.com**」のIPアドレスを教えてください！



# DNSサーバ

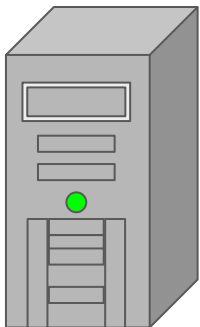


ドメインとIPアドレスを紐付けるサーバです。  
コンピュータのあだ名（ドメイン名）から、  
コンピュータの本名（IPアドレス）を教えてくれる役割を果たしています。

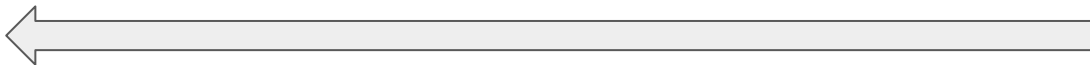


## ②IPアドレスの取得

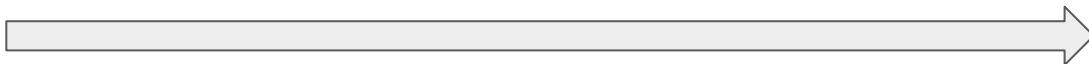
DNSサーバ



① 「**www.facebook.com**」の**IPアドレス**を教えてください！



② 「**www.facebook.com**」のIPアドレスは**31.13.82.36**です！



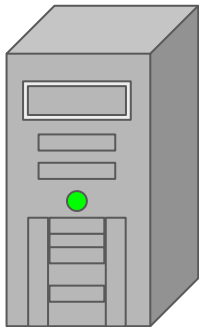
### ③Webサーバにアクセス



③ このページを送ってください！

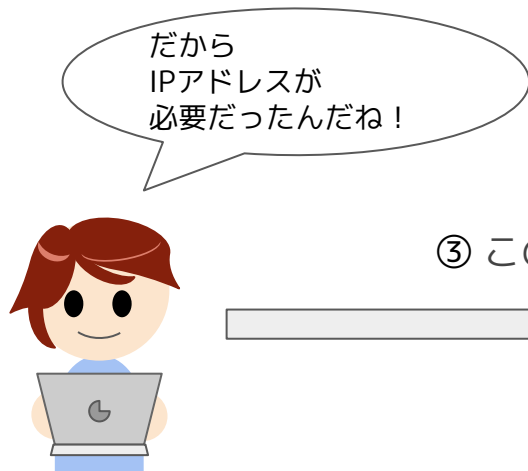


Webサーバ



DNSサーバに問い合わせたIPアドレスは  
どのように使われるかというと...

## IPアドレスを使ってWebサーバを特定する

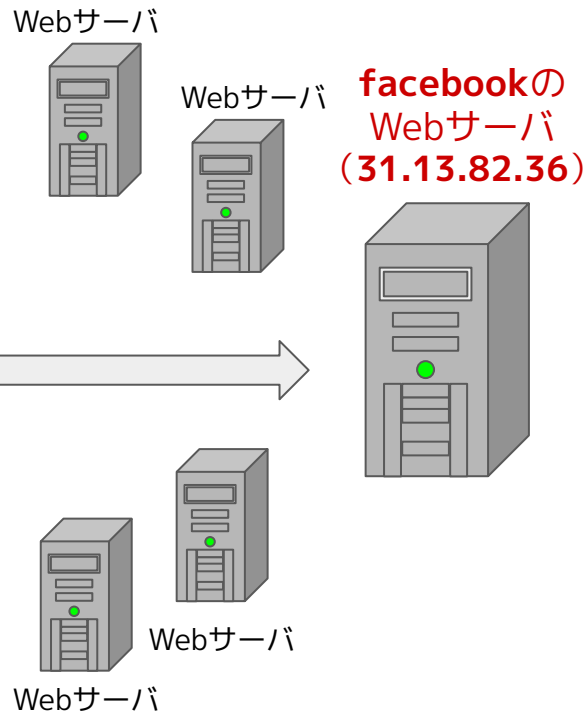


③ このページを送ってください！

無数にあるWebサーバを

特定するためにIPアドレスを取得したというわけです。

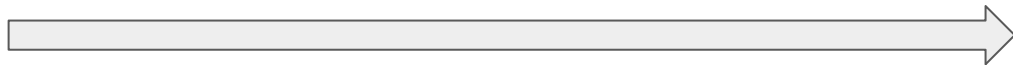
(IPアドレスはコンピュータの住所のようなものですね！)



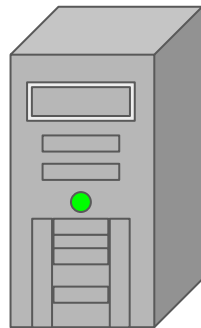
### ③Webサーバにアクセス



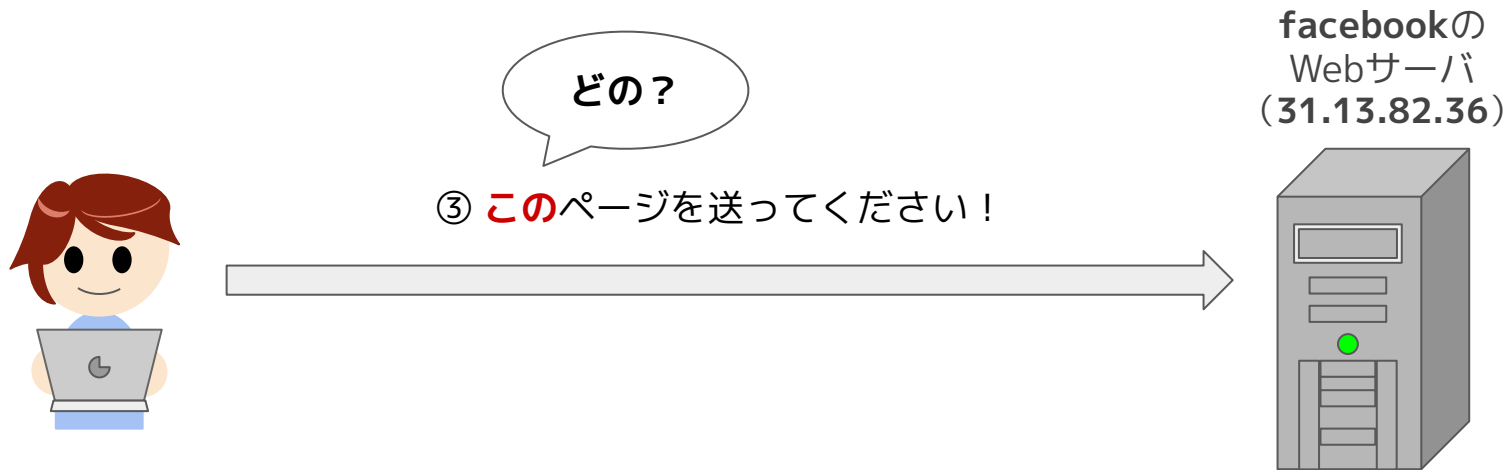
③ このページを送ってください！



facebookの  
Webサーバ  
(31.13.82.36)



### ③Webサーバにアクセス



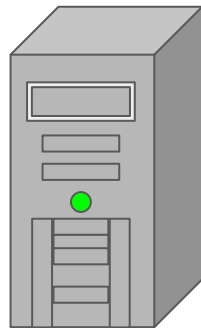
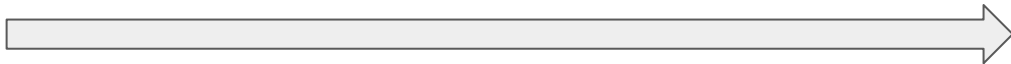
### ③Webサーバにアクセス

https://www.facebook.com/**DMMWEBCAMP**/

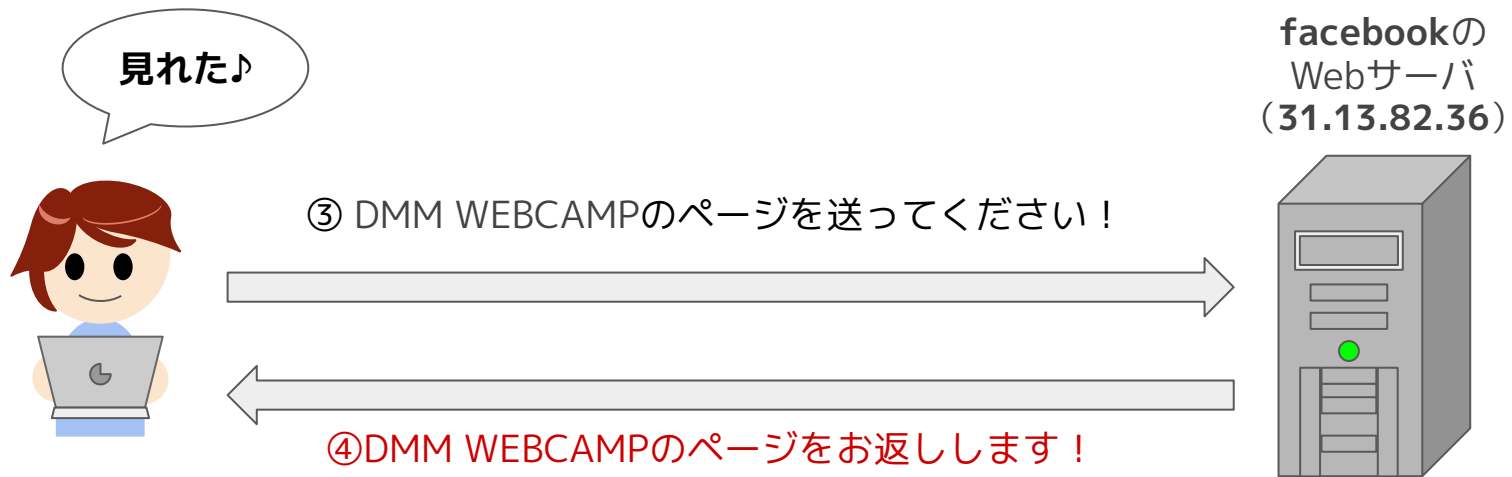
facebookの  
Webサーバ  
(31.13.82.36)



③ **DMM WEBCAMP**のページを送ってください！



## ④HTMLファイルの取得

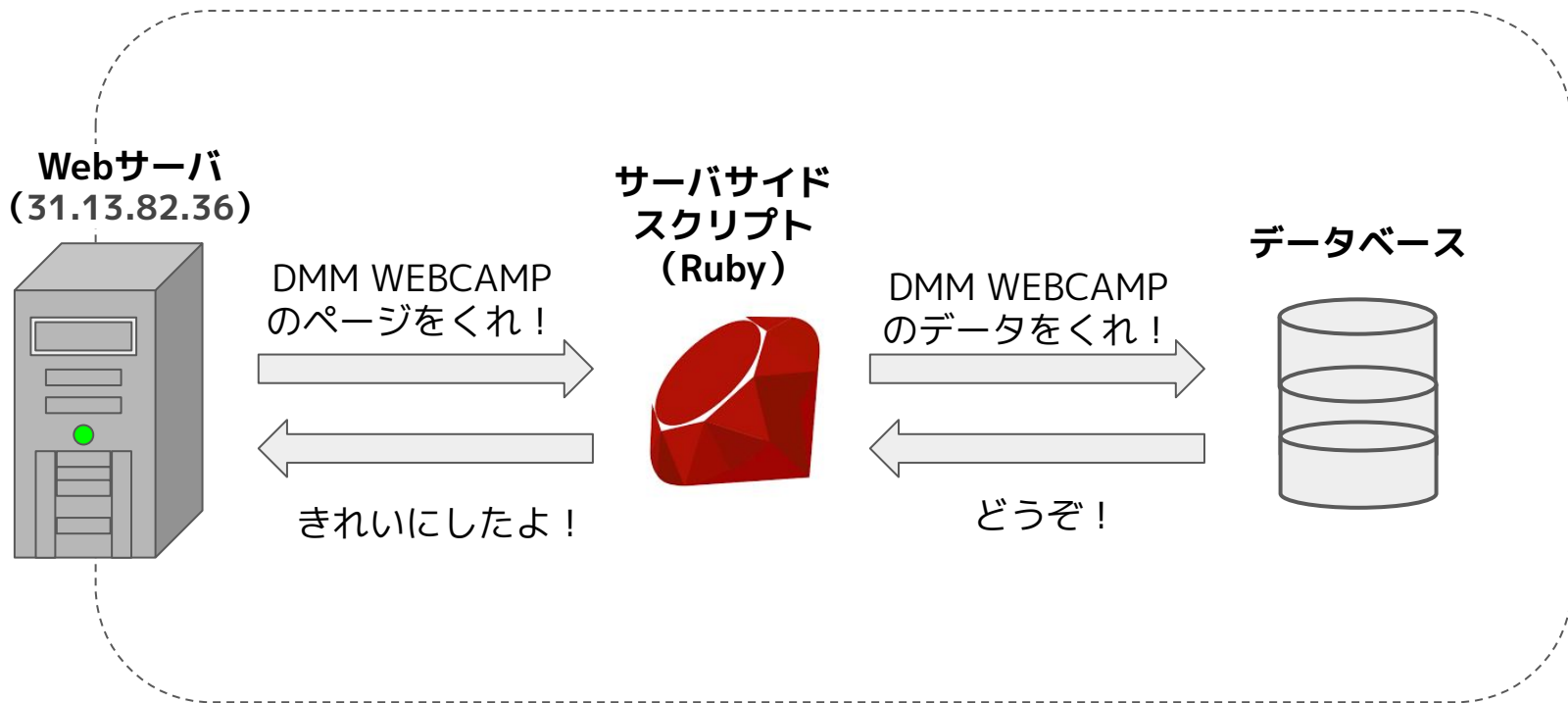


重要なのはWebサーバの中！

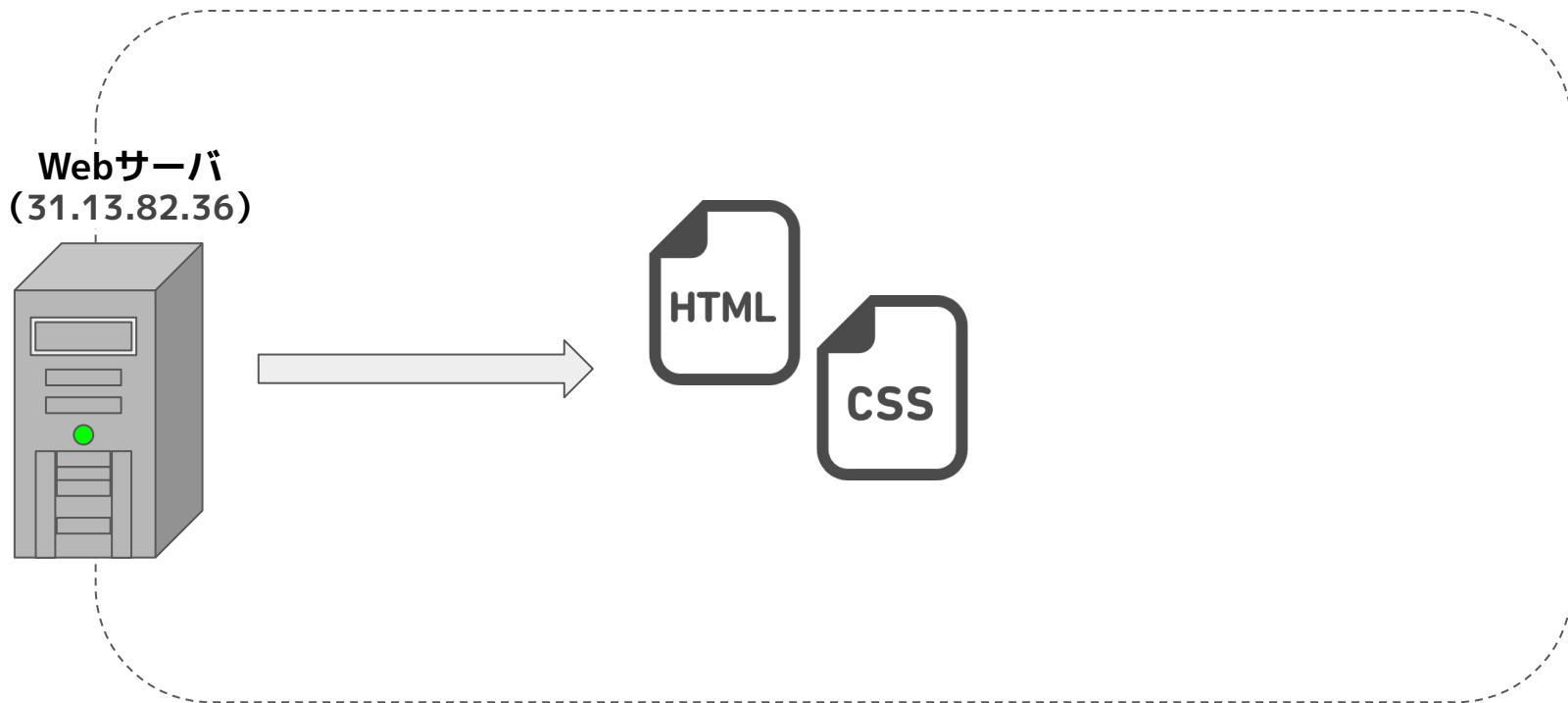
...なのですが、  
**Webサーバ**の中で行われていることが  
一番重要なので、  
ひとつひとつ見ていきます！



# Webサーバの全体像はこんな感じです



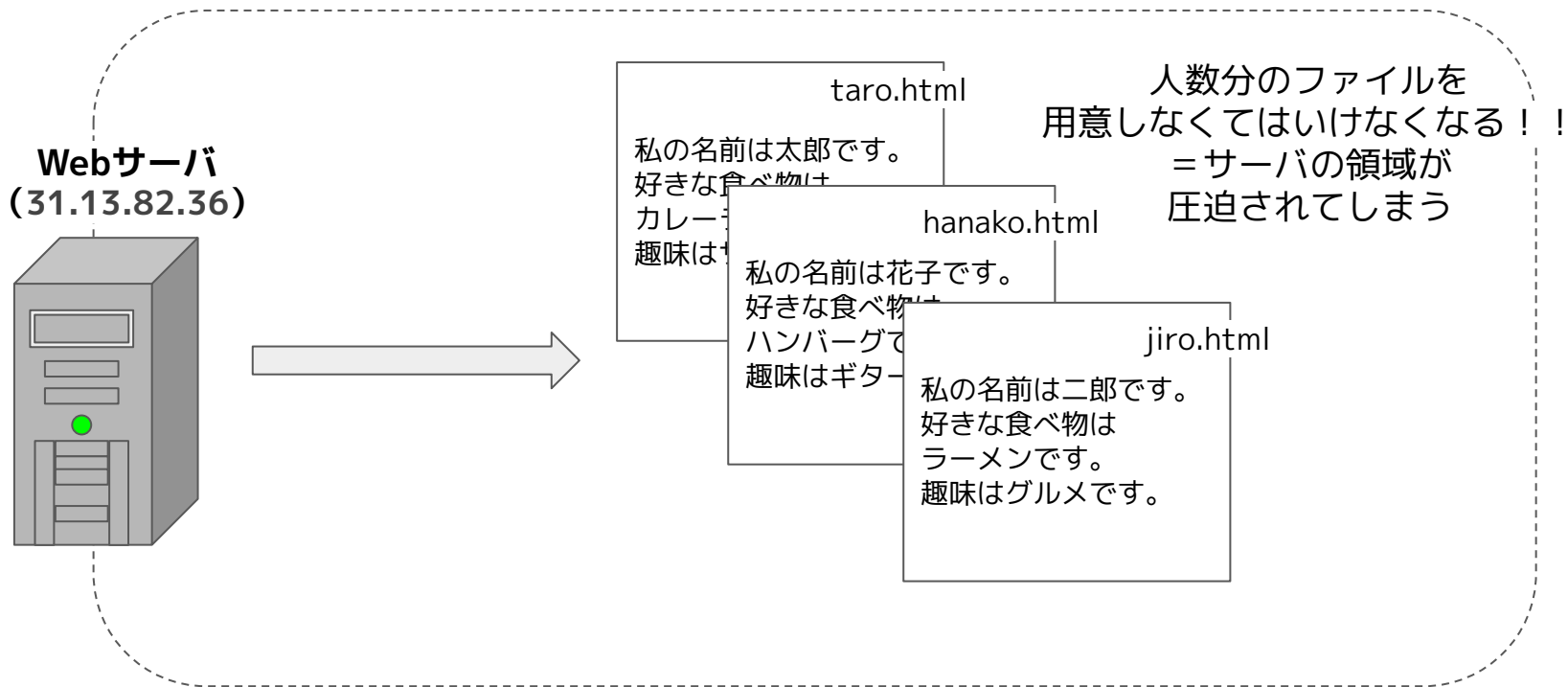
# HTML・CSSの配置だけでもいいが...



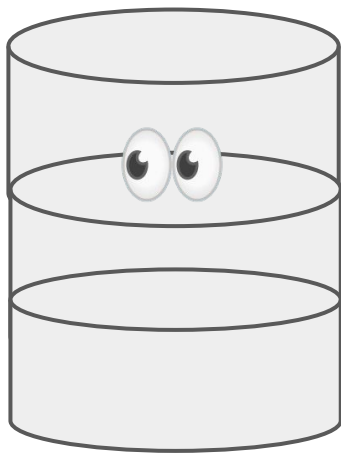
# HTML・CSSの配置だけでもいいが...



# ファイルが無数に必要となってしまう



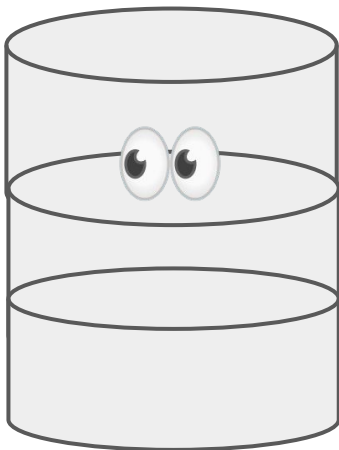
# ここで登場するのがデータベース



エクセルの表のような形式で、データを蓄えておける仕組みです。

ID	名前	食べ物	趣味
1	太郎	カレーライス	サッカー
2	花子	ハンバーグ	ギター
3	二郎	ラーメン	グルメ

# データベース



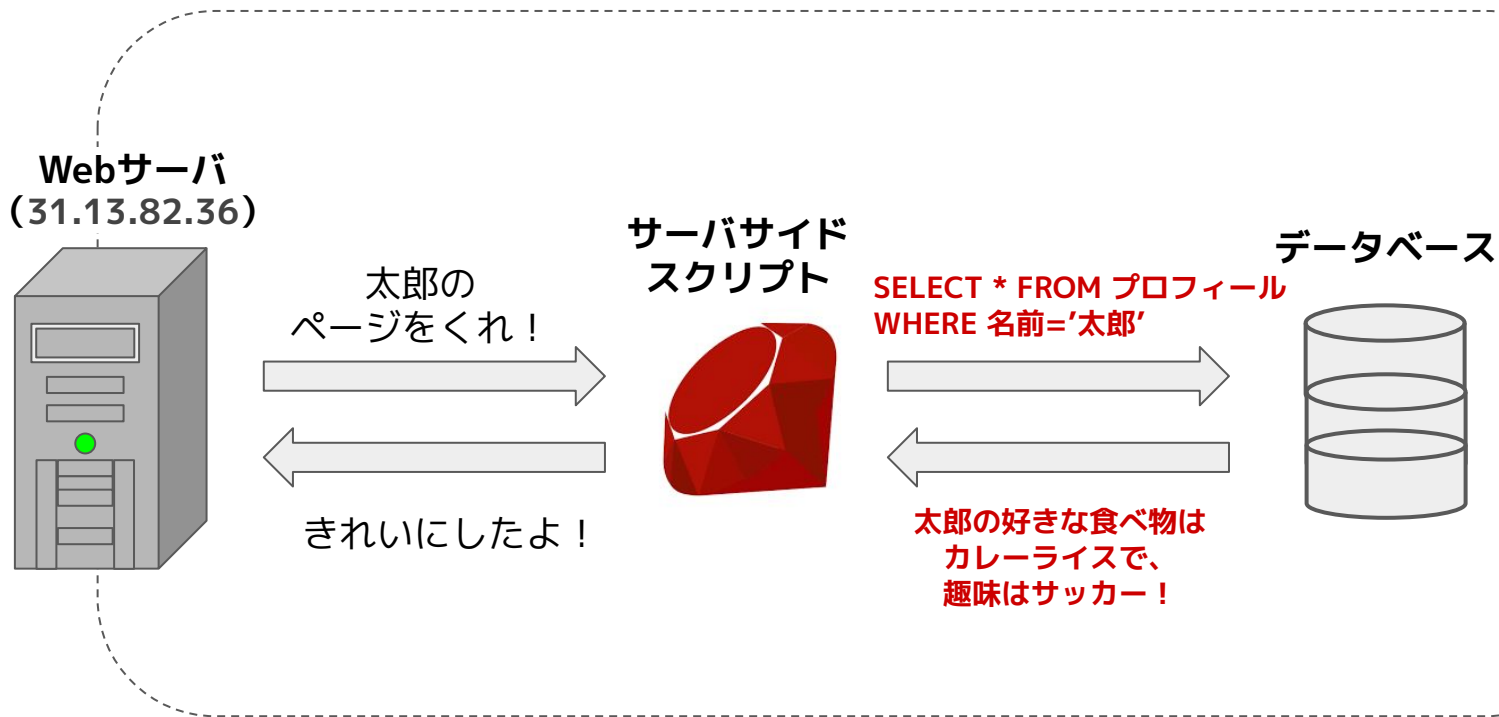
データベースにアクセスし、検索や登録、削除などを行うためには、SQLという言葉を使います。

**SELECT \* FROM profiles WHERE 名前='太郎'**

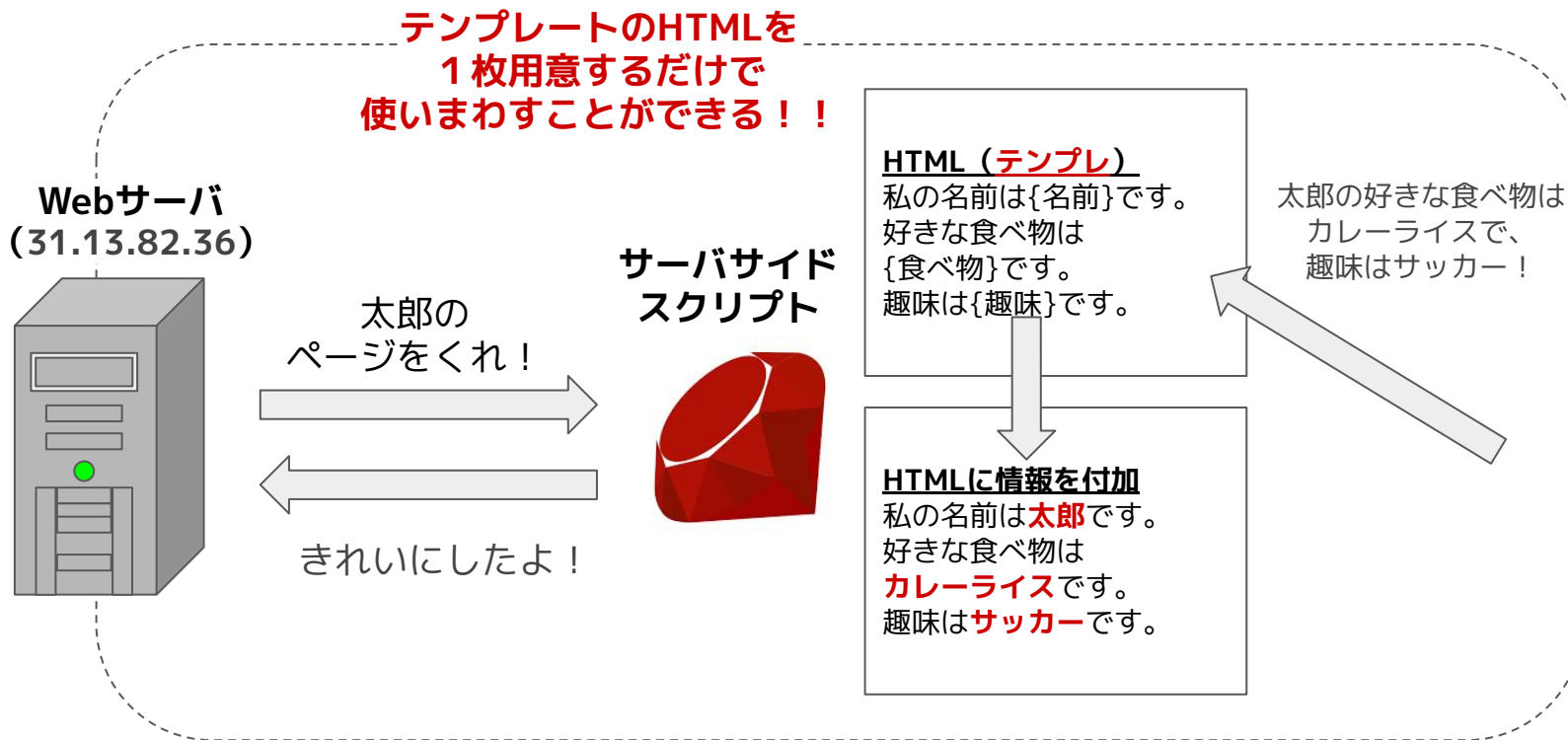
プロフィール表(profilesという名前にした場合)

ID	名前	食べ物	趣味
1	太郎	カレーライス	サッカー
2	花子	ハンバーグ	ギター
3	二郎	ラーメン	グルメ

# Webサーバの全体像

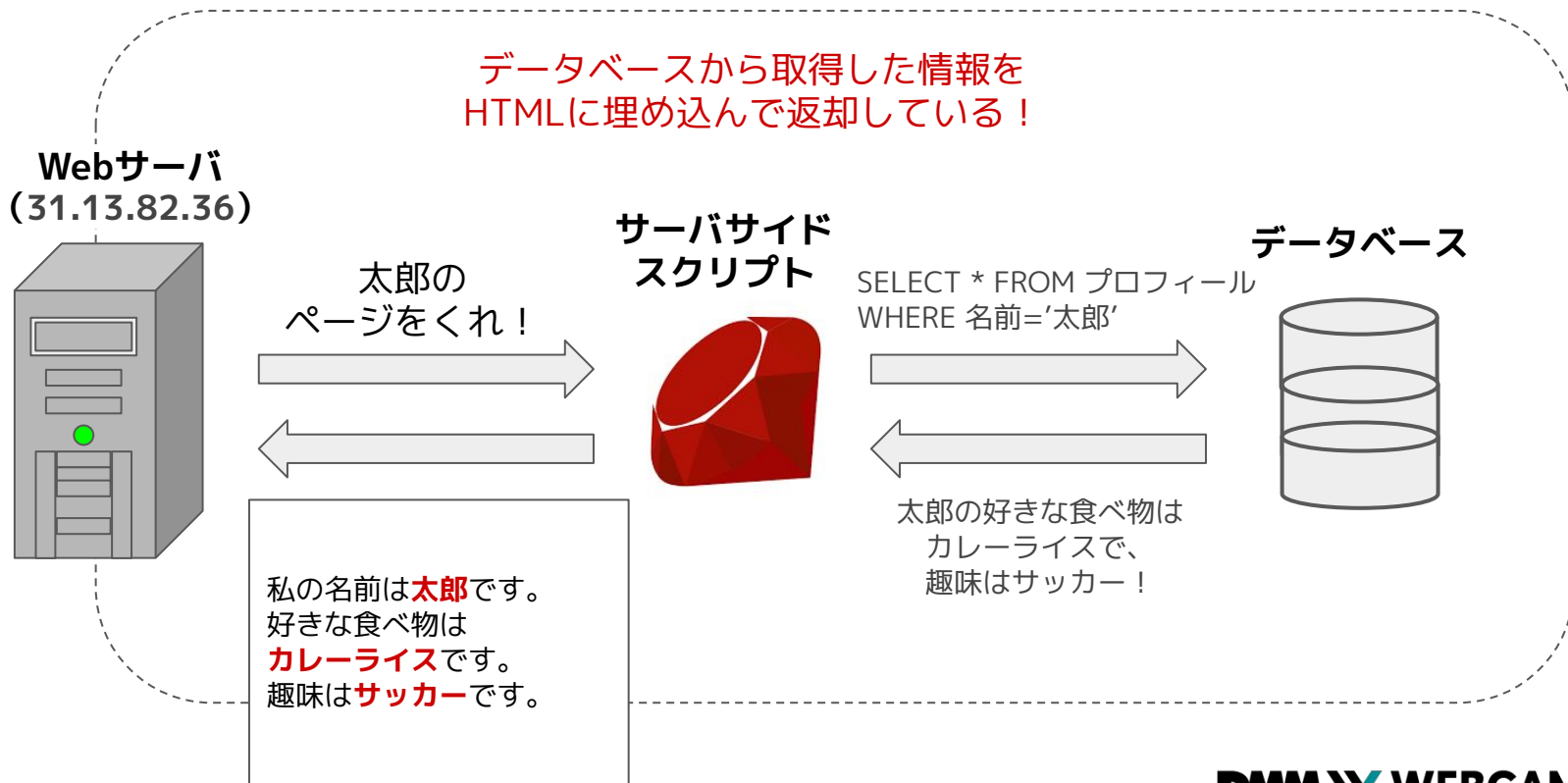


# データベースの情報をHTMLへ！

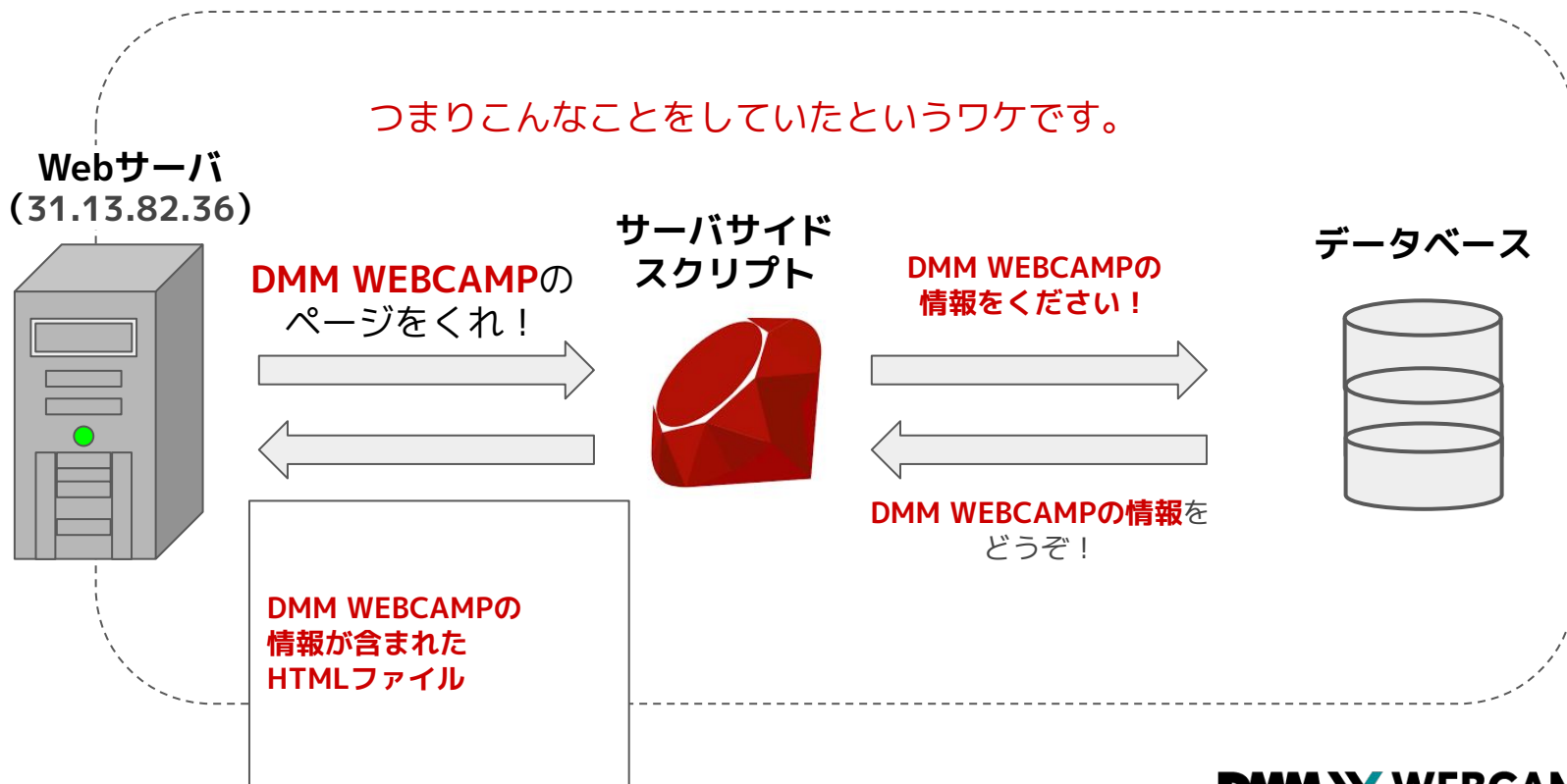




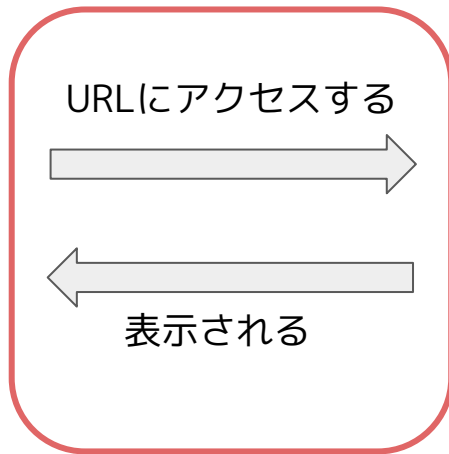
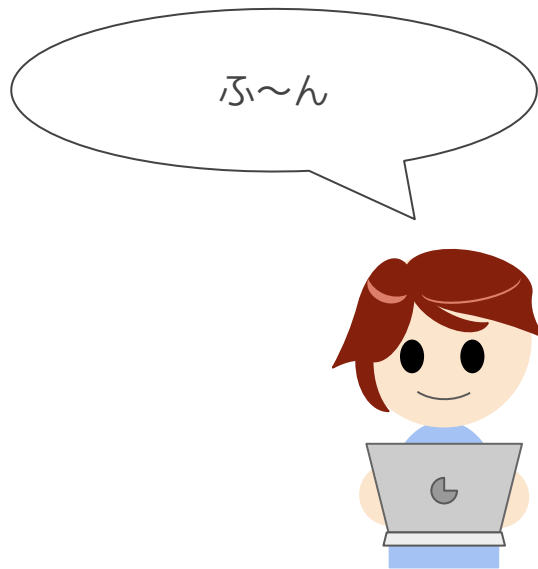
# データベースの情報をHTMLへ！



# データベースの情報をHTMLへ！



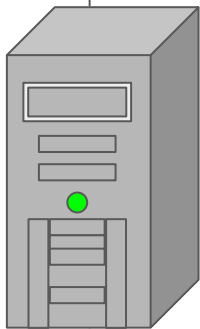
# たった一瞬でこれだけのことが行われています！



普段何気なくアクセスしているサイトは、  
こんな流れでみなさんのブラウザに表示されているのです！

# いろいろなことができます！

Webサーバ  
(31.13.82.36)



## 分岐

趣味が音楽の人は...  
和食が好きな人は...

## 繰り返し

人数分表示する

## 演算

カレーが好きな人を  
数える

などなど...

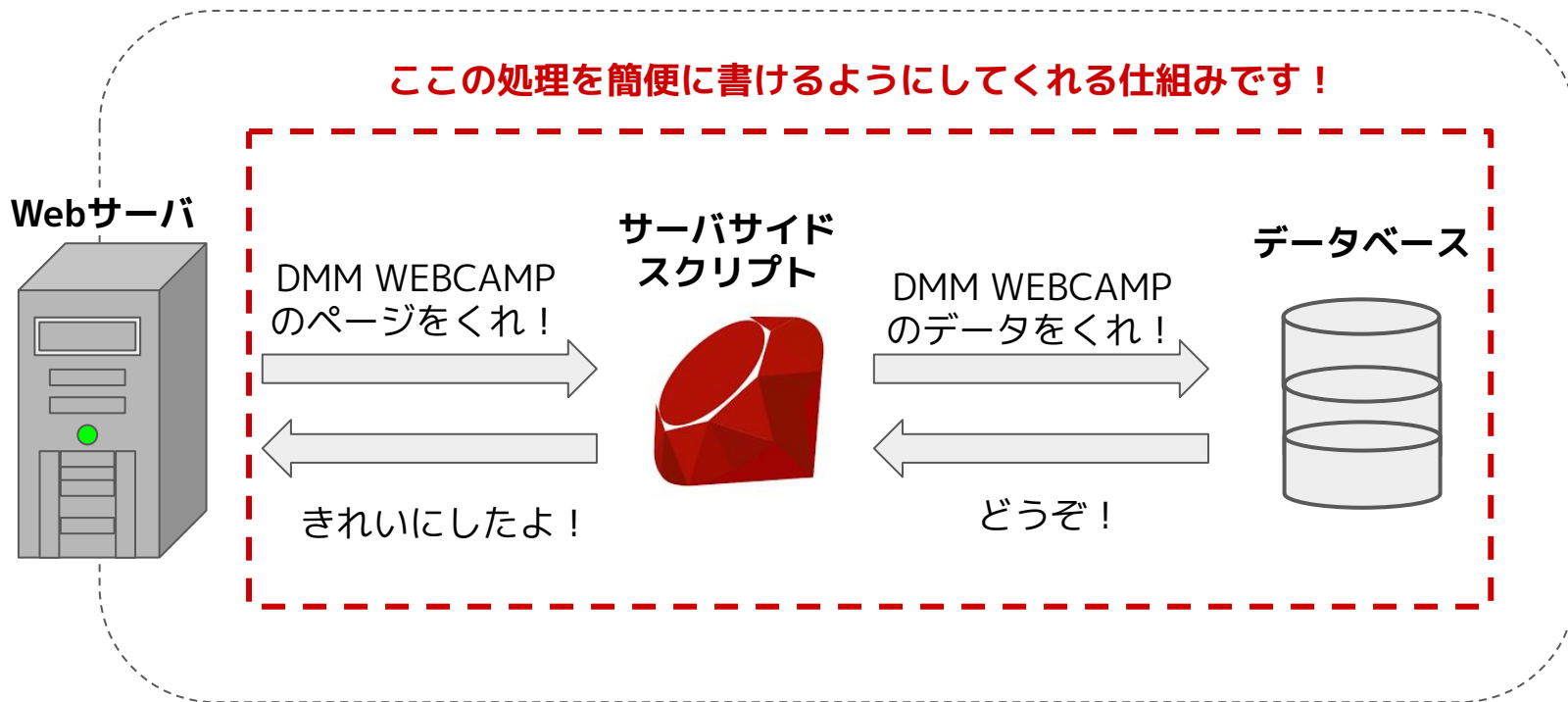
プログラミングでは、  
様々な処理を実現することができます！！

# Ruby on Railsはどこで登場するの...？

Webアプリケーション制作



# Ruby on Railsの役割



# 例えば...

## Gem (ジェム)

Webサイトでよくある処理を、誰かが代わりに作ってくれています！

- Devise : ログイン機能
- kaminari : ページング機能
- runsack : 検索機能

などなど、Gemは無数にあります！

先人たちのおかげで、Rubyでイチから書く、という必要がありません。

自分でGemを作ることもできますよ！

# 例えば...

## ORマッパー

データベースにはSQLという言語を使ってアクセスする、と書きましたが（46ページ参照）、よく使う文法はORマッパーという仕組みで簡略化することができます。

例えばプロフィール表(profiles)のID1番を取得したい場合、SQLで書くと

```
SELECT * FROM profiles WHERE ID=1
```

ですが、RailsはORマッパーのおかげで

```
Profile.find(1) だけで済むんです！短くなった～！
```



サーバ上で起こっていることを  
「**バックエンド**」と呼びます。

ユーザの目に触れない部分のことです。  
(**サーバサイド**とも呼びます)

# ソースコードの保管庫

# ソースコードとは？

ソースコードとは、プログラミング言語を使って記述された文章のことです。これから皆さんは、たくさんのソースコードを書いていくことになると思います！

興味があったら読んでみよう！

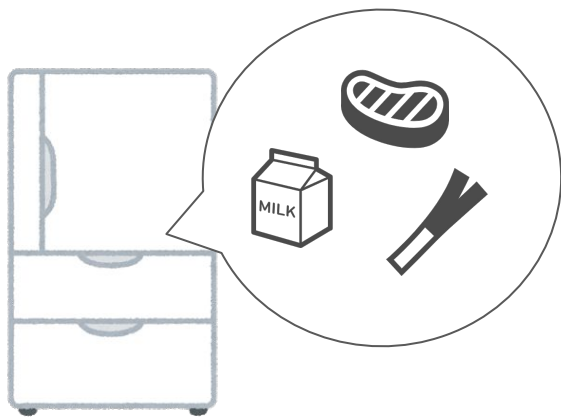
## ソースコード (source code) とは

「分かりそう」で「分からない」でも「分かった」気になれるIT用語辞典

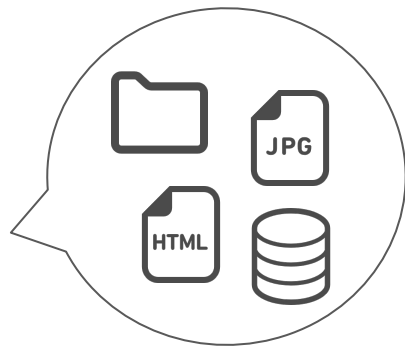


# Gitとは？

Gitでは、リポジトリと呼ばれる保管庫に、ソースコードや変更履歴、コメントなどを格納していきます。食べ物のリポジトリは冷蔵庫、プログラミングの成果物を入れるリポジトリはGit、のような感覚でOKです！



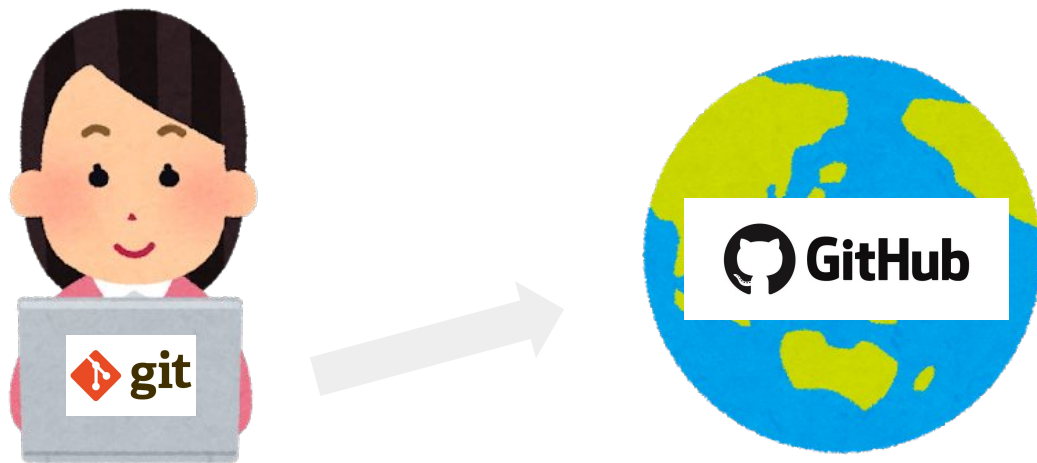
**git**



# GitHubとは？

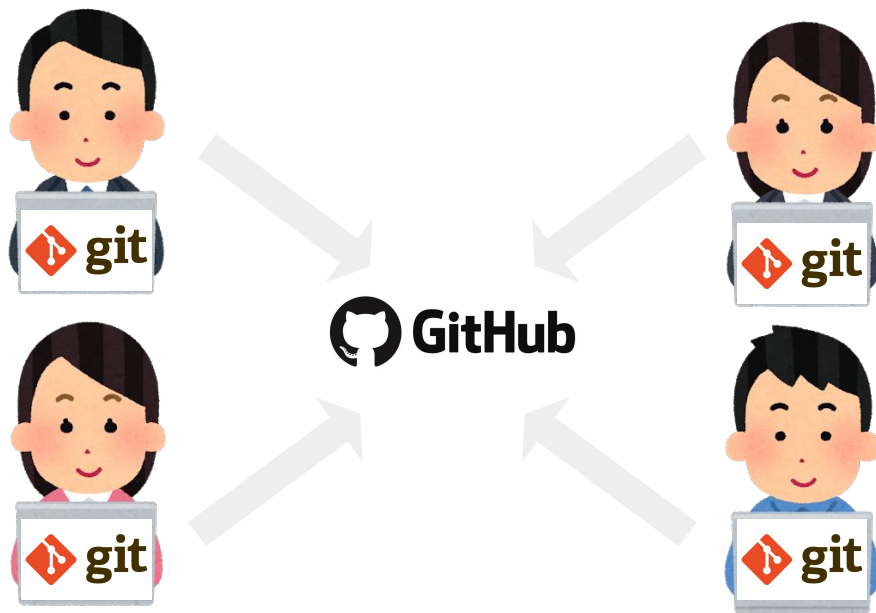
Gitの仕組みを利用したWebサービスです。

自分のソースコード等を保存したり、世界に公開したりすることができます。



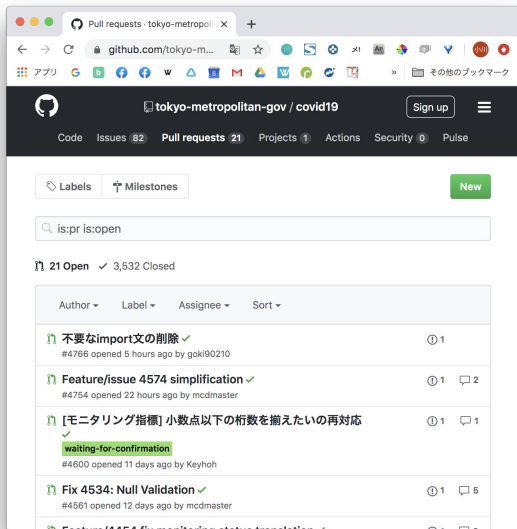
# チーム開発でも力を発揮します！

個人開発でも便利ですが、チーム開発など複数人での開発プロジェクトでも使われています。複数人での利用も楽しみにしててください！



# 普段利用しているサイトの開発にも携われる！

企業や行政なども、サイトのソースコードをGitHubに公開しています。  
あなたも様々なサイトの開発に関わることができたりします！



<https://github.com/tokyo-metropolitan-gov/covid19>

東京都の新型コロナウイルス感染症対策サイトは、オープンソースソフトウェア（OSS）で公開されています。  
GitHubを覗くと、一般のプログラマがPull Request（コードの変更・追加等のリクエスト）を出しています。

# Git & GitHubは第一関門...

GitとGitHubは学習を進める上での第一関門かもしれません。

なかなかイメージが湧かず理解しづらいと思います。

「Git 例え」「Git イメージ」

などで検索してみると、様々なサイトにHITします。

これからずっと使っていくので、直に慣れます！

**大丈夫！徐々に仲良くなっていきましょう！**



# ITを楽しみましょう！

以上が、最初の学習フェーズで学ぶ内容についての概要でした。  
普段自分が何気なくアクセスしているサイトが、  
こんな仕組みで動いていると思うと、なんだかワクワクしませんか？  
身の回りのテクノロジーに興味を持つこと。  
これはIT業界を目指す上で非常に大切であり、  
**好奇心は日々の学習を加速してくれます！**

ちなみに

## ちなみに①

サーバサイドスクリプトには、様々な言語があります。

Rubyをはじめ、PHPやPython、Java等様々な存在しますが、  
基本的な仕組みは同じです。方言のようなものだと思ってください。

よく聞くJavaScriptは、クライアントサイドスクリプトと言います。  
サーバサイドスクリプトがWebサーバ上で動作するのに対し、  
JavaScriptはブラウザ上で動作します。

**JavaとJavaScriptは全くの別物なので気をつけましょう！**

## ちなみに②

「**サーバ**」は、**Server = 供給**する人です。様々な要求（リクエスト）に応えるので、そう呼ばれています。レストランで料理をサーブする、と言ったりしますよね。そのサーブです。

皆さんが使っているPCをサーバとして使うことも可能です。しかし、長時間の稼働や数多くのリクエストに耐えなければならず、サーバ専用の機器として用意されているものを使うことが多いです。



## ちなみに③

「サーバ」の対義語は、「**クライアント**」です。

Clientは「依頼する人」です。「このページ見せてくれよ～」とリクエストを出す皆さんのPCやスマホは「クライアント」です。

クライアントのリクエストに対して、供給（レスポンス）しているのがサーバです。

## ちなみに④

WebサーバにHTMLをそのまま置いても問題ありません。

データベースを利用しなくてもいいサイト（お店の公式サイトや行政のサイトのように、頻繁に更新のないサイト）などはHTMLを直接修正することが多いです。そのようなサイトを「**静的サイト**」と言ったりします。

対して、SNSやブログなどデータベースに頻繁に更新が入るサイトは「**動的サイト**」と言います。静的サイトは「**情報提供**」、動的サイトは「**対話**」を目的としています。

## ちなみに⑤

**https://www.facebook.com/DMMWEBCAMP/**

この部分ってなんなの？と思った方、いませんか？

この部分は、「スキーム」と言い、通信方式の指定をする部分です。httpやhttpsはインターネット通信であることを意味します。（httpとhttpsの違いは検索してみてください！）

デスクトップ上のHTMLファイルをブラウザで開くと、「**file:///**」と表示されると思います。これは、「ファイルやフォルダを開く」という指定です。

**お疲れ様でした！**