コンピュータネットワーク 第11回 Web技術概要,クライアントサイ ド:HTML/CSS/JavaScript

静岡理工科大学情報学部 コンピュータシステム学科幸谷智紀

https://na-inet.jp/compnet/

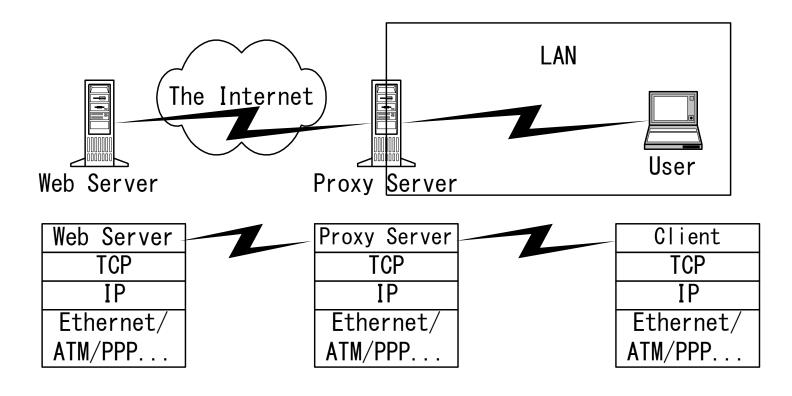
本日の内容

- Webの概要
- Webアプリケーションとは?
- クライアントサイドの技術要素
- DOMとHTML

Webとは?

- 1990年にHTMLと共に開発された, The Intenet(TCP/IP)普及のためのキラーアプリケーション(第1回参照)
- World Wide Web → Web (「蜘蛛の巣」の意味)
- Worldwide Environment for Browsingの略称ではない
- Web(システム)の構成要素
 - HTTP(HyperText Transportation Protocol) :TCPの上で動作するプロトコル
 - + over TLS:暗号化したHTTPパケットをやり取りするプロトコル(HTTPS)
 - Webサーバ(HTTPサーバ): クライアントからのリクエストに応じてデータを送出するホスト
 - Webクライアント(Webブラウザ): ユーザ側が使用する閲覧用ソフト

Webシステムの概念図



- 基本はTCPでクライアントとサーバが直接HTTP・HTTPS通信する
- プロキシ(Proxy, クライアントの代理)が間に入るケースもある →Webサーバ側からはプロキシがアクセスしたように見える。

「http://」と「https://」の違い

- 暗号化したHTTPがHTTPS (HTTP Secure or HTTP over TLS)になる
- SSL 3.0 (Secure Sockets Layer), 1995年→TLS (Transport Layer Security) 1.0, 1999年→TLS 1.3, 2018年
 - ・認証局(CA, Certification Authority)による正式な鍵発行→正式なhttpsアドレスとなる

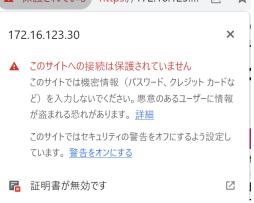
例) https://cs-tklab.na-inet.jp/

非正規な鍵発行(オレオレ認証)→外部から認められないhttpsアドレス

例) https://172.16.123.30/

現状, Googleなど主要サーチエンジンではhttpsアドレスを優先 (2015年12月告知)



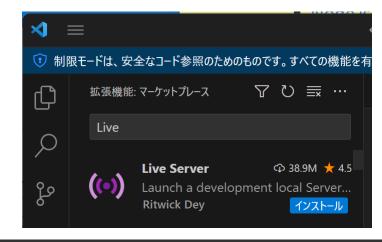


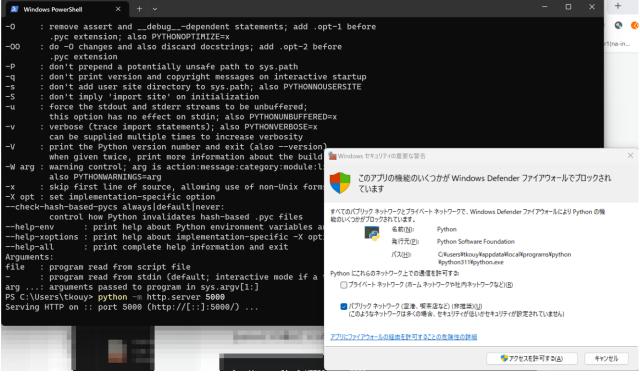
ref) https://developers.google.com/search/blog/2015/12/indexing-https-pages-by-default

→以降, httpsアドレスの使用をデフォルトする

一番手軽なWebサーバ

- Node.JSのWebサーバ機能
- PythonのWebサーバ機能





主要Webサーバのシェア

Ref) https://w3techs.com/technologies/overview/web_server

注: () 内は2023年の数値

- 1. Nginx(エンジンエックス) 33.8% (34.2%)
- 2. Apache(アパッチ) 28.2% (31.1%) → 172.16.123.30で使用中
- 3. Clouldflare (クラウドフレア) 22.6% (21%) (WebサーバというよりCDN)
- 4. LiteSpeed(ライトスピード) 14.0% (12.5%)
- 5. Microsoft IIS 4.3% (5.3%)
- 6. Node.js 3.8% (2.9%)

Apache

- 本家 https://httpd.apache.org/
- Webサーバの老舗(NCSA HTTPdサーバ 1994年)→現在もシェア2位注) National Center for Supercomputing Applications at Univ. Illinois
- 主なOS(Linux, Windows, macOS)上で動作する。
- 最新版・・・これ以前のものには脆弱性の指摘あり。
 - 1.3.9 (Version 1系)→サポート終了
 - 2.4.57 (Version 2系)

Apacheの設定ファイル

- Apacheの設定は全てhttpd.conf(から派生したconfファイル)に書く。
 - サーバ動作の指定
 - /ディレクトリの指定
 - PHP, CGI動作の指定
 - MIME typeの指定

等等···

- 2.4系統で大幅にConfファイルの書き方が 変わっている
- https://httpd.apache.org/docs/2.4/



モジュール | ディレクティブ | FAQ | 用語 | サイトマップ

HTTP SERVER PROJECT

Apache HTTP サーバ バージョン 2.4

<u> Apache > HTTP サーバ > ドキュメンテーション</u>

Apache HTTP サーバ バージョン 2.4 ドキュメント

翻訳済み言語: da | de | en | es | fr | ja | ko | pt-br | ru | tr | zh-cn

この日本語訳はすでに古くなっている 可能性があります。 最近更新された内容を見るに は英語版をご覧下さい。

Google 検索

リリースノート

Apache 2.3/2.4 の新機能

Apache 2.1/2.2 の新機能

Apache 2.0 の新機能

<u>2.2 から 2.4 へのアップグレー</u>

ー Apache ライセンス

リファレンスマニュアル

<u>コンパイルとインストール</u>

動

終了と再起動

実行の設定用ディレクティブ

モジュール

<u>マルチプロセッシングモジュー</u>

<u>ル (MPM)</u>

<u>フィルタ</u> ハンドラ

Expression parser

<u>サーバとサポートプログラム</u> 用語集

ユーザの手引

Getting Started

の新機能 アドレスとポートのバインド

設定ファイル

プグレー セクションの設定

キャッシュ機能

<u>コンテントネゴシエーション</u>

動的共有オブジェクト (DSO)

環境変数

<u>ログファイル</u>

URL をファイルシステムにマ

<u>ップする</u>

性能に関する調整

セキュリティ情報

サーバ全体の設定

SSL/TLS による暗号化

CGI の Suexec 実行

mod_rewriteによる URL Rewriting

バーチャルホスト

How-To / チュートリアル

認証、承認、アクセス制御

CGI: 動的コンテンツ

.htaccess ファイル

Server Side Includes (SSI)

ユーザ専用ディレクトリ

(public_html)

プラットフォーム固有の情報

Microsoft Windows

RPMベースのシステム (Redhat

/ CentOS / Fedora)

Novell NetWare

EBCDIC 版

その他

<u>よくある質問 (FAQ)</u>

サイトマップ

開発者のためのドキュメント

<u>ドキュメンテーションプロジェ</u>

クトへの協力

<u>その他</u> ウィキ

翻訳済み言語: da | de | en | es | fr | ja | ko | pt-br | ru | tr | zh-cn

Copyright 2023 The Apache Software Foundation.

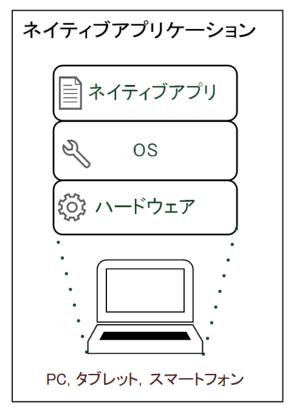
この文書は Apache License, Version 2.0 のライセンスで提供されています。.

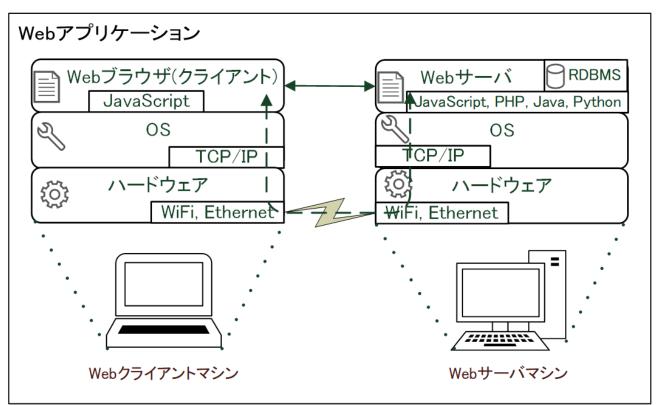
https://・・・/の挙動

- 1. DocumentRoot ・・・で指定されたディレクトリが参照される。
- 2. DirectoryIndex・・・(又は<IfModule mod_dir.c>・・・
 </IfModule>)で指定された名前のファイルが呼び出される。
- UserDir ・・・(又は<lfModule mod_userdir>・・・</lfModule>)でユーザ毎のWebページを置くディレクトリを指定する。

 $\underline{\text{https://www.hoge.jp/}^{tkouya/}} \leftarrow \rightarrow /\text{home/tkouya/public_html/index.html}$

Webアプリケーションとは?





- Webシステム上で動作するアプリケーションの総称
 - Webサーバ・クライアント(ブラウザ)
 - クライアントサイドの技術要素: HTML/CSS/JavaScript
 - サーバサイドの技術要素: PHP/Python/JavaScript/C#
- ネイティブアプリ化したWebアプリケーションも存在する

Webアプリケーションの長所・短所

Webアプリケーションの長所

- ブラウザとサーバでコンピュータ資源を分担しあえる
- ダウンロードやインストールをする必要がない
- ひとつのWebアプリでいろいろな端末(OS)に対応させることができる

Webアプリケーションの短所

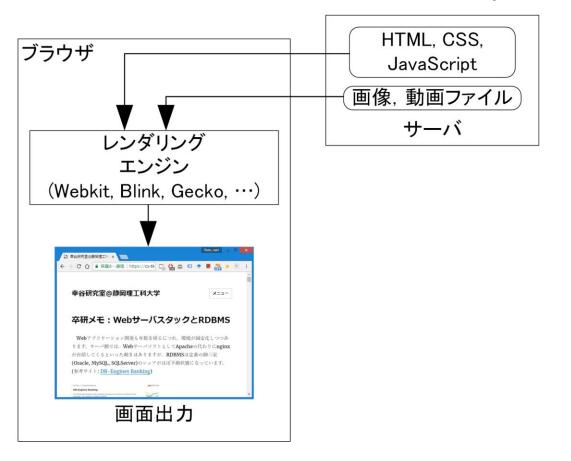
- ネイティブアプリケーションに比べて処理が遅い
- 端末の機能(例:スマートフォンのカメラなど)を活かしづらい

Webサーバ側のソフトウェア階層

Webアプリケーション PHP PHP C#, ASP.NET C# スクリプト |(スクリプト, プログラム)| スクリプト スクリプト・プログラム .NET Core PHP core PHP core .NET Core .Net Framework 実行コア MySQL IIS SQLServer WebサーバドRDBMS Apache MySQL Apache nginx | SQLServer Linux Linux Windows Server OS Windows Ubuntu, CentOS, ... Ubuntu, CentOS Windowsベースの Linuxベースの XAMPP for LAMP .NET環境 .NET環境 Windows

• Weサーバ側で動作するアプリケーション用のインタプリタやデータベース(ファイル入出力代わり)を組み込んでいる

Webクライアント(ブラウザ)の仕組み







Google Chrome, Safari, Microsoft Edge (Webkit, Blink)

Firefox (Gecko)

- レンダリング(画面出力)はすべてブラウザ側の仕事→レンダリングエンジン
- レンダリングエンジンの集約化が進む
- レンダリング作業をサーバ側で行うケースもある

クライアントサイドの技術要素

- HTML (Hypertext Markup Language)・・・ テキストの構造を定義するマークアップ 言語(タグ(tag))
- CSS (Cascading Style Sheet)・・・HTMLにより構造化されたテキストのデザインを決定する指定子
- →DOM(Document Object Model)
- JavaScript (「Java」に非ず)・・・動的に DOMを制御することができる言語。国際 規格としてはECMA Scriptと呼ばれる

```
「<metaucharset="UTF-8"」/>
<title>CSSサンブル</title>
<style_type="text/css">
#aaau{ucoloru:uredu}
</style>
〈body〉
〈p〉ここの文字は黒です。〈/p〉
<puid="aaa">ここは文字が赤です。
ここも文字は黒です。
</body>
```



ここの文字は黒です。

ここは文字が赤です。

ここも文字は黒です。

DOMとHTMLファイル

```
<!DOCTYPE html>
                                                                                  html
                                                                                                head
                                                                                                               meta
<html lang="ja">
<head>
                                                                                                                title
 <meta charset="UTF-8" />
 <title>DOMテスト: 静的HTMLファイル</title>
                                                                                                                h1
                                                                                                body
</head>
<body>
                                                                                                             address
テスト
<h1 id="first h1" onclick="window.alert('h1タグをクリックしました。')">DOMテス
ト</h1>
<address style="color: red">HPC Lab.@ Shizuoka Institute of Science and
Technology</address>
</body>
</html>
```

- HTML, CSSを文書(Document)中のオブジェクト(Object)として扱う
- 動的なDOMの制御はJavaScriptで行う

[復習] 本日の内容

- Webの概要
- Webアプリケーションとは?
- クライアントサイドの技術要素
- DOMとHTML

本日の課題

- 1. 172.16.123.30の自分のフォルダ(ホームディレクトリ)に「public_html」フォルダを作る
- 2. public_htmlに右のHTMLファイルを 作る
- 3. <a href="https://172.16.123.30/~自分のユーザ" 14 ID/にブラウザ(何でもよい)にアクセスしてスナップショットを取る
- 本日の課題フォームに回答し、上 記のURLとスナップショットもアッ プロードせよ。

https://forms.office.com/r/uwnnNDWfmN

```
<!DOCTYPE html>
<html lang="ja">
   <head>
      <meta charset="UTF-8" />
      <title>自分の名前のWebページ</title>
   </head>
   <body>
      <h1>自分の名前のWebページ</h1>
      <h3>20YY年MM月DD日(曜日)</h3>
      <hr />
      ここにメインコンテンツを置く
      <hr />
      <address>自分の名前(ローマ字で)(c)20YY</address>
   </body>
```

