

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  
ド: **HTML/CSS/JavaScript**

静岡理科大学

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/>

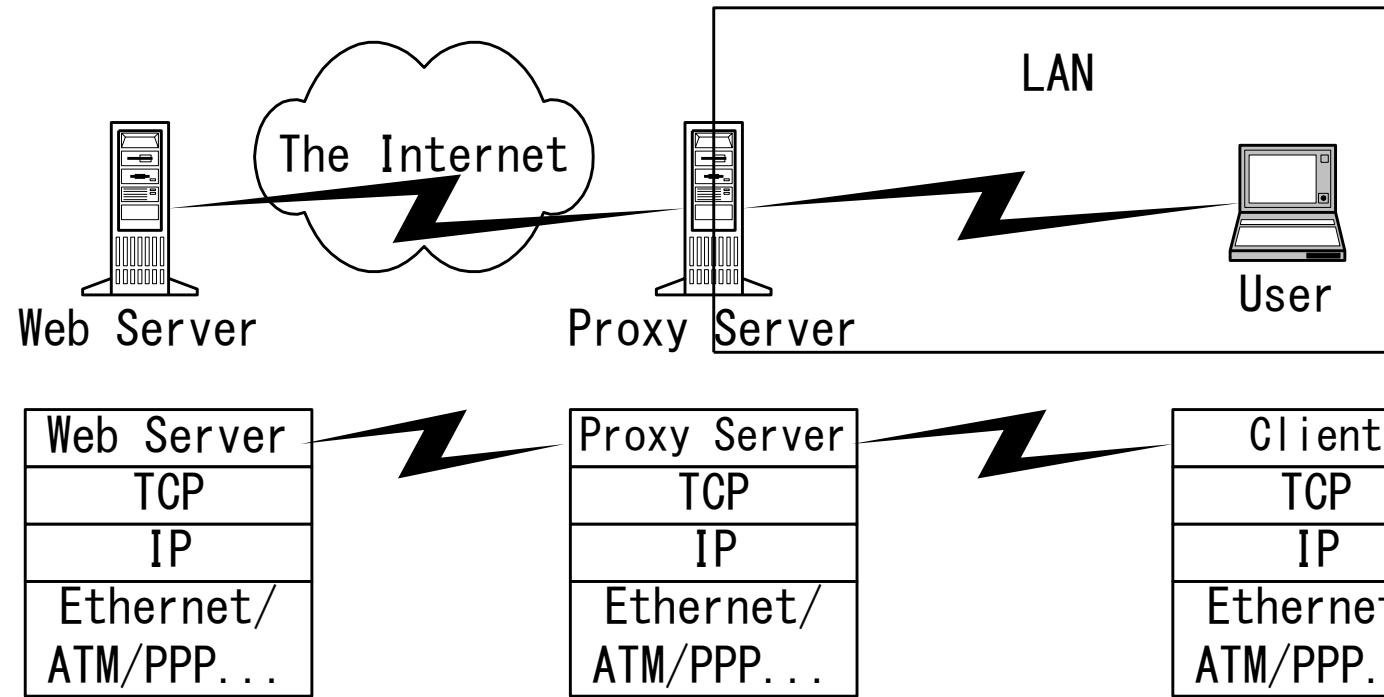
# 本日の内容

- 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 Internet(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(HTTP~~S~~)
  - 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



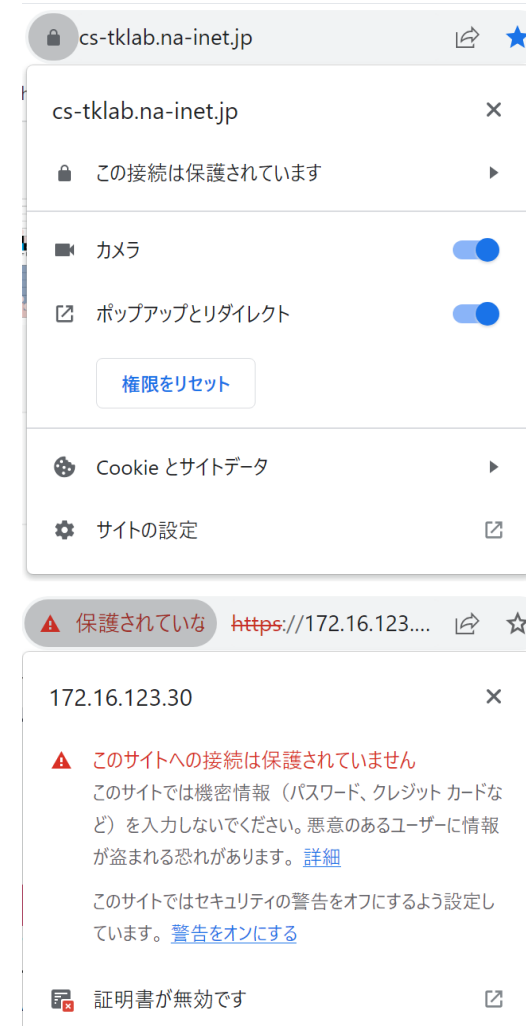
- 基本はTCPでクライアントとサーバが直接HTTP・HTTPS通信する
- プロキシ(Proxy, クライアントの代理)が間に入るケースもある  
→Webサーバ側からはプロキシがアクセスしたように見える。

# 「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)

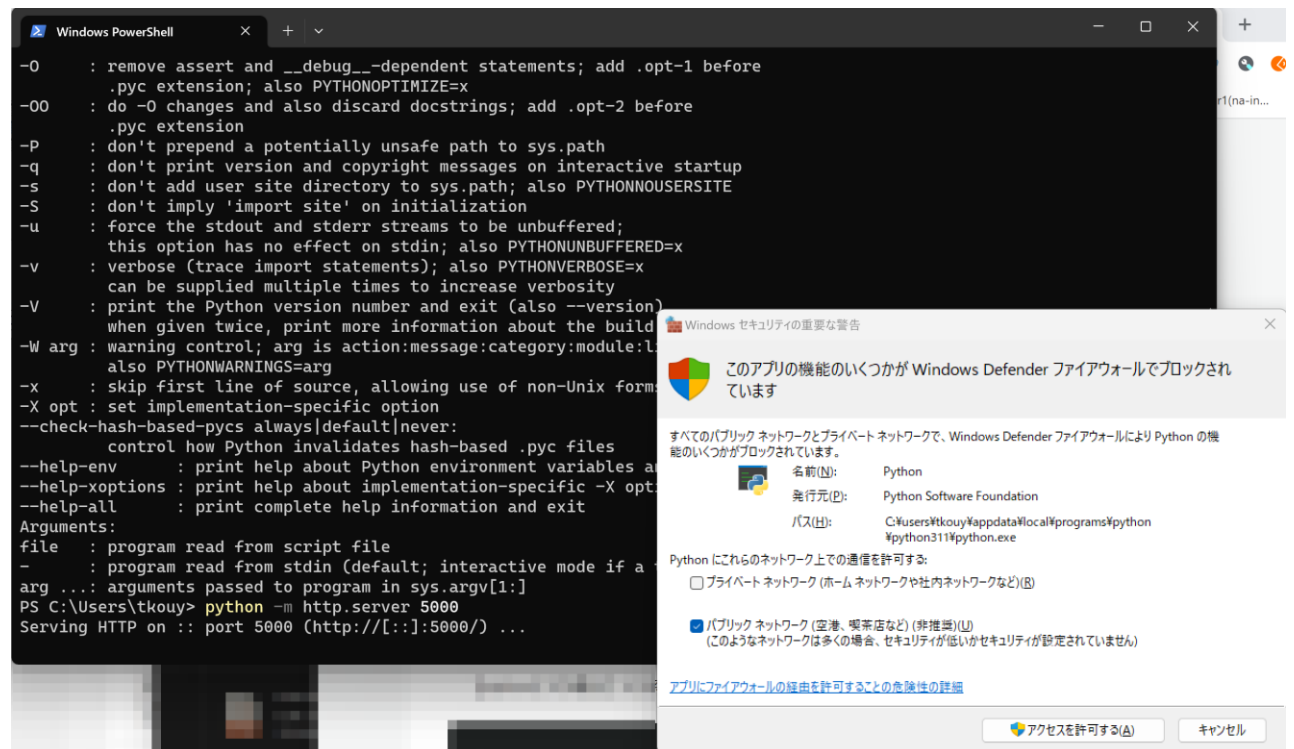
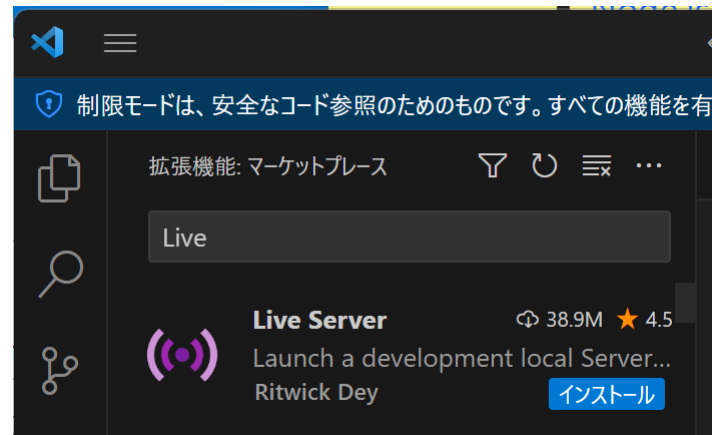
ref) <https://developers.google.com/search/blog/2015/12/indexing-https-pages-by-default>

→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

- Visual Studio Code EditorのLive Server
- 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](https://w3techs.com/technologies/overview/web_server)

1. Nginx(エンジンエックス) 34.2%
2. Apache(アパッチ) 31.1% → 172.16.123.30ではこれを使用
3. Cloudflare (クラウドフレア) 21% (WebサーバというよりCDN)
4. LiteSpeed(ライトスピード) 12.5%
5. Microsoft IIS 5.3%
6. Node.js 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/>

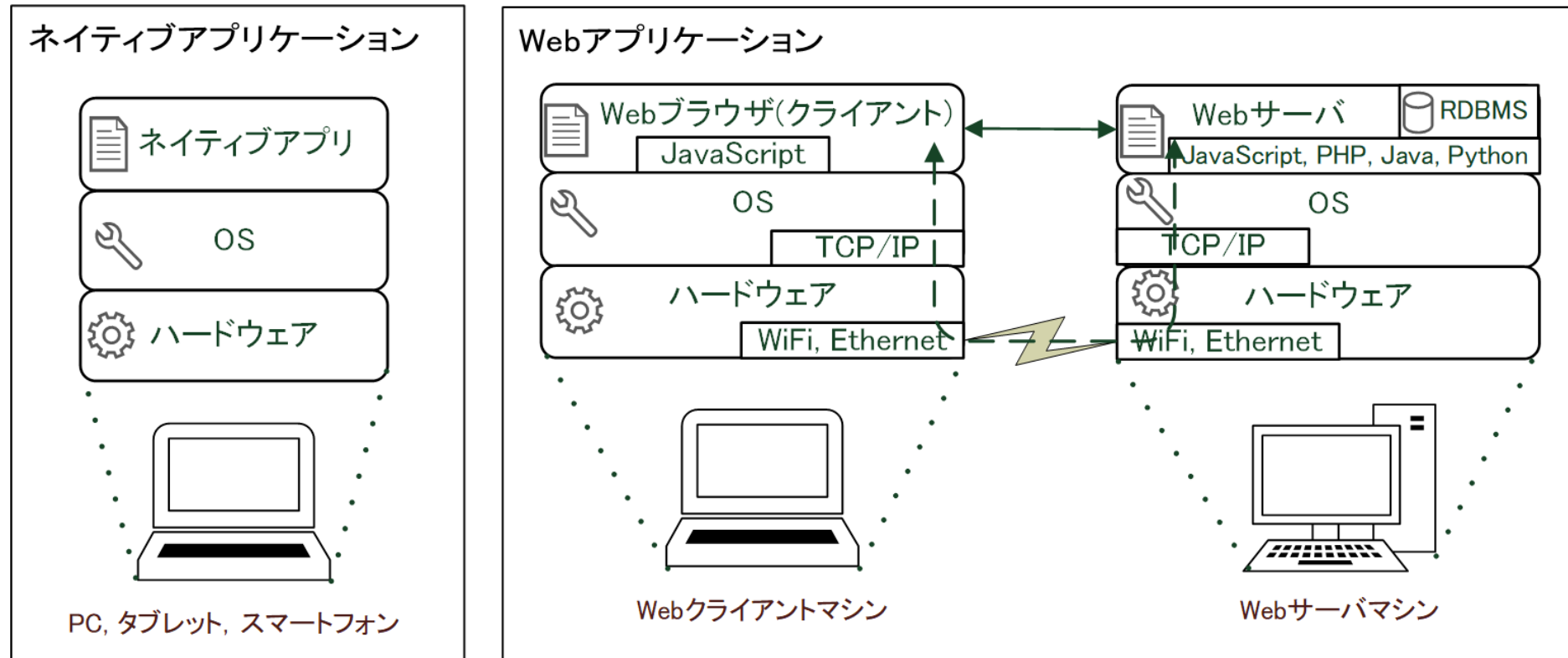
The screenshot shows the Apache HTTP Server 2.4 documentation page. At the top, there is the Apache logo and the text 'APACHE HTTP SERVER PROJECT Apache HTTP サーババージョン 2.4'. Below this, there is a navigation bar with links for 'モジュール | ディレクティブ | FAQ | 用語 | サイトマップ'. The main heading is 'Apache HTTP サーババージョン 2.4 ドキュメント'. There is a search bar with the text 'Google 検索'. Below the search bar, there are three columns of links: 'リリースノート', 'ユーザの手引', and 'How-To / チュートリアル'. The 'リリースノート' column includes links for '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', '2.2 から 2.4 へのアップグレード', and 'Apache ライセンス'. The 'ユーザの手引' column includes links for 'Getting Started', 'アドレスとポートのバインド', '設定ファイル', 'セクションの設定', 'キャッシュ機能', 'コンテンツネゴシエーション', '動的共有オブジェクト (DSO)', '環境変数', 'ログファイル', 'URL をファイルシステムにマップする', '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', and 'バーチャルホスト'. The 'How-To / チュートリアル' column includes links for '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', 'よくある質問 (FAQ)', 'サイトマップ', '開発者のためのドキュメント', 'ドキュメンテーションプロジェクトへの協力', 'その他', and 'ウィキ'. At the bottom, there is a footer with the text '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。' and a language selection bar with links for 'da | de | en | es | fr | ja | ko | pt-br | ru | tr | zh-cn'.

# 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<IfModule mod\_userdir> . . . </IfModule>)でユーザ毎のWebページを置くディレクトリを指定する。

<https://www.hoge.jp/~tkouya/> ↔ /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

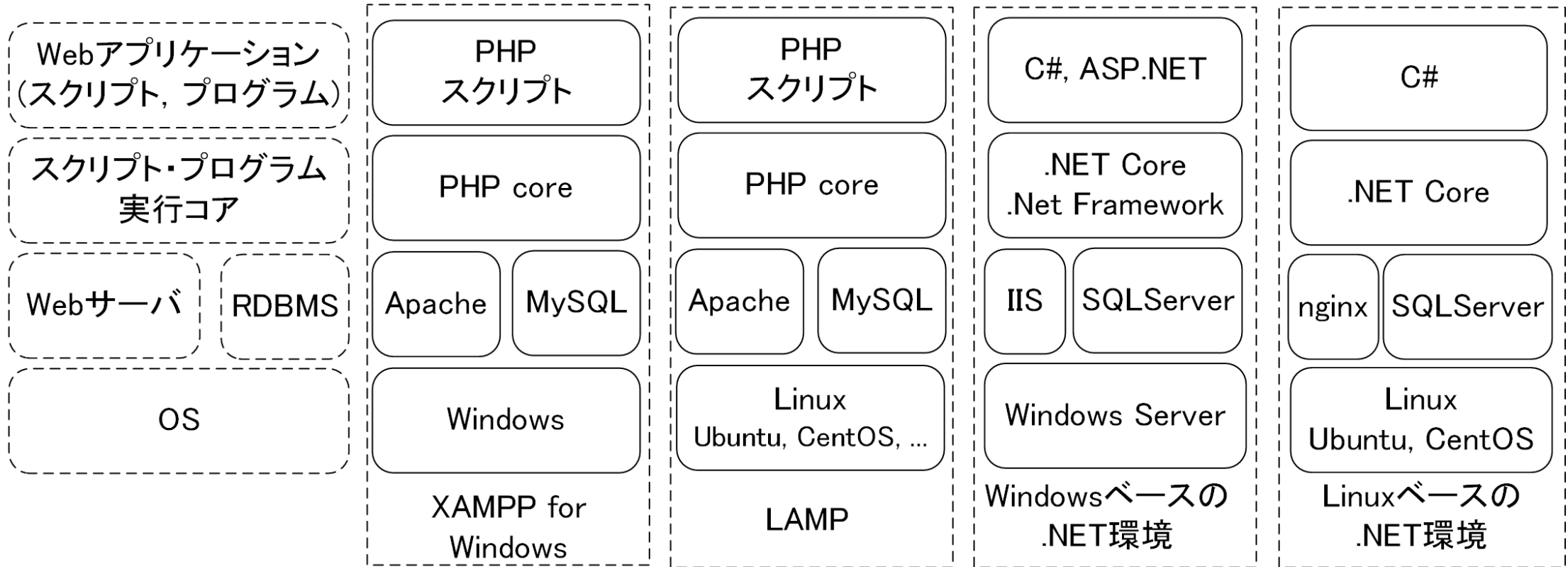
## Webアプリケーションの長所

- 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

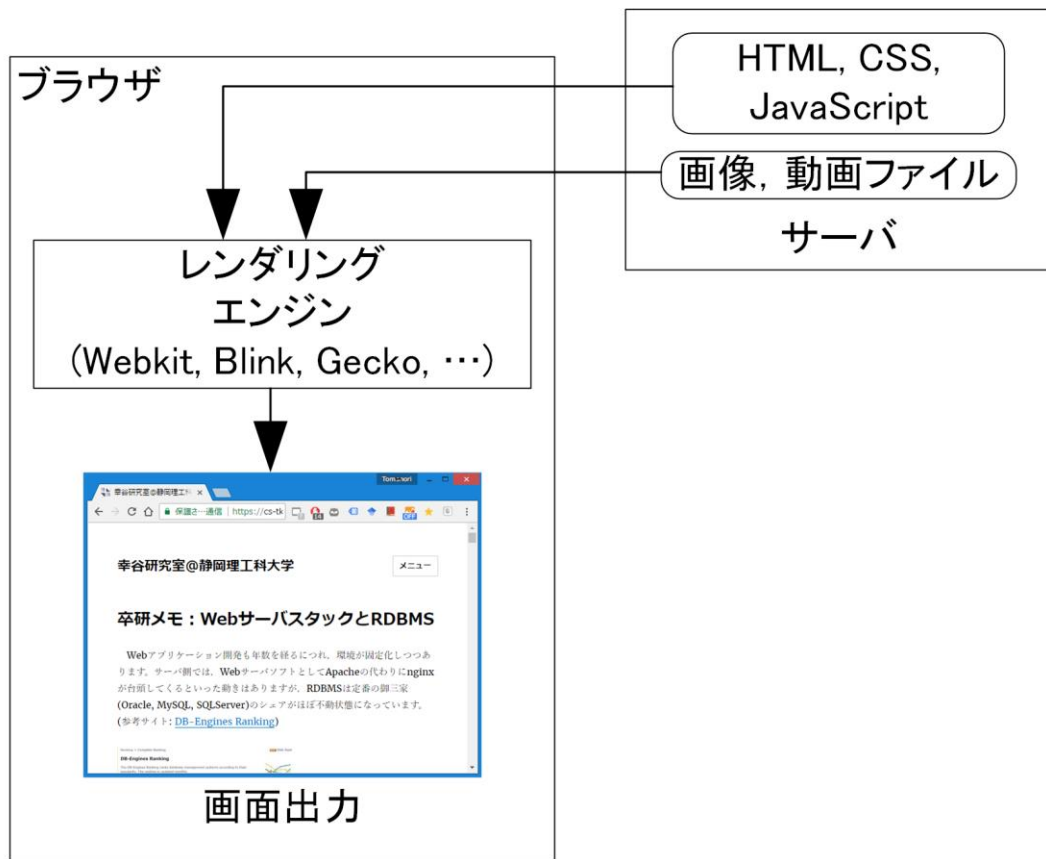
- ネイティブアプリケーションに比べて処理が遅い
- 端末の機能（例：スマートフォンのカメラなど）を活かしづらい

# 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に)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

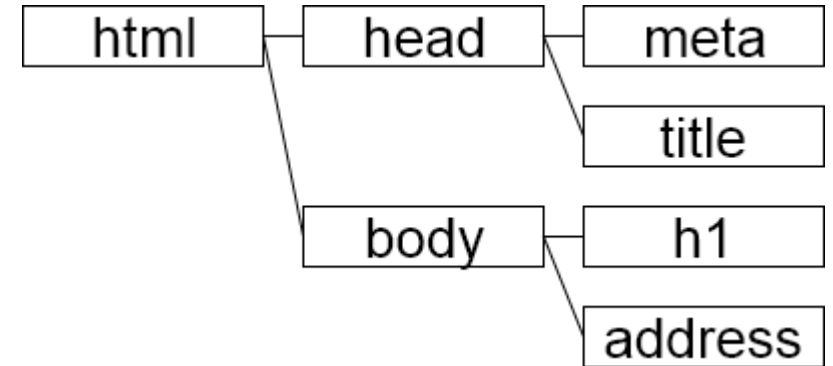
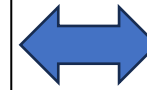
```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta charset="UTF-8"/>
5   <title>CSSサンプル</title>
6
7 <style type="text/css">
8 <!--
9 #aaa { color:red; }
10 -->
11 </style>
12
13 </head>
14 <body>
15 <p>ここの文字は黒です。</p>
16
17 <p id="aaa">ここは文字が赤です。</p>
18
19 <p>ここも文字は黒です。</p>
20
21 </body>
22 </html>
```



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

```
1 <!DOCTYPE html>
2 <html lang="ja">
3 <head>
4   <meta charset="UTF-8" />
5   <title>DOMテスト: 静的HTMLファイル</title>
6 </head>
7 <body>
8 テスト
9 <h1 id="first_h1" onclick="window.alert('h1タグをクリックしました。')">DOMテス
  • ト</h1>
10 <address style="color: red">HPC Lab.@ Shizuoka Institute of Science and
  • Technology</address>
11 </body>
12 </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. <https://172.16.123.30/~自分のユーザー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
4. 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>

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

コンピュータネットワーク 第11回 本日の課題

