

# WWWのおはなし

---

神戸大学理学部惑星学科4回生 伊藤 旭人  
神戸大学理学部惑星学科4回生 入場 柚太

(流体地球物理学教育分野)

# 要旨/目的

- WWW (World Wide Web) について、その登場と変遷について知る.
- その構成と WWW サーバについて基礎的なことを理解する.

# 目次

- **WWWとは**
  - WWW
  - WWW の歴史
- **WWWの特徴**
  - Web ブラウザ
  - ハイパーリンク
  - HTML
  - HTTP, HTTPS
  - URL
- **前半のまとめ**
- **WWWサーバ**
  - Apache
  - Hiki
  - CGI
- **後半のまとめ**

WWWとは

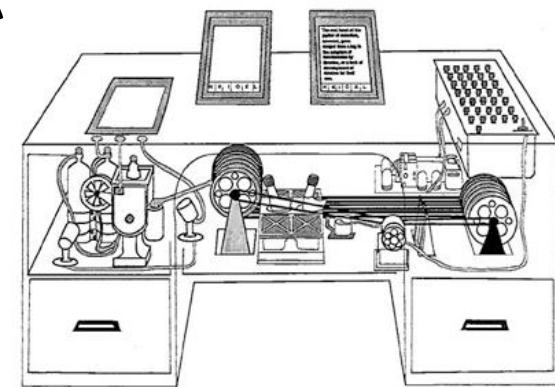
# WWWとは

- WWW(**W**orld **W**ide **W**eb)
  - インターネット上で提供されるハイパーテキストシステム
    - ハイパーリンクでドキュメント間をつなぐ
    - それぞれの文書が必ず住所 (URL) をもつ
  - 直訳すると「世界中に広がった蜘蛛の巣」

# WWWの歴史① ハイパーテキストの歩み

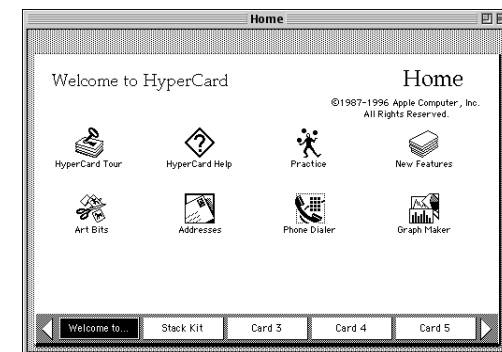
- ヴァネヴァー・ブッシュ:「Memex」の概念  
(1945)

- 従来の図書館的な情報の検索システム(目録)では、現代の情報量の爆発的増加に追いつかなくなってしまうという危機を予測
- 上記の問題に対処するべく考えられた架空の装置が「Memex (Memory Extenderの略)」
- 図書館と電氣的に接続されていて、所蔵する本やフィルムを表示し、さらにその資料同士が相互にリンクしていて、自動的に相互参照を辿って他の作品を表示するというデバイス



# WWWの歴史① ハイパーテキストの歩み

- **テッド・ネルソン: Project Xanaduの立ち上げ (1960)**
  - コンピュータネットワーク上にハイパーテキストをベースにしたシステムを構築しようと計画
  - 「ハイパーテキスト」という用語は, 1965年にネルソンが初めて使用
- **ダグラス・エンゲルバート: NLSシステムの開発 (1965)**
  - 世界で初めてGUIを備え, ハイパーテキストリンクをコンピュータ上で実現したシステム
  - マウス, ラスタースキャン型ディスプレイ, プレゼンテーションソフトウェアなどを実用化
- **ビル・アトキンソン: Hyper Cardの開発 (1987)**
  - カードとカードをリンクするボタンを配置することでハイパーリンクを作成することが可能



# WWWの歴史② ティム・バーナーズ＝リーの功績 —WWWの登場—

- ENQUIRE プロジェクト(1980)
  - スイスの欧州原子核研究機構(CERN)において多くの研究者に効率良く情報を行き渡らせることが重要な課題
  - WWWの根本的なアイデアを多く含んでいた
  - 広く公開はされなかった
- World Wide Webの開発(1990)
  - 世界初のWebサーバhttpdと、世界初のWebブラウザWWWクライアントを開発・発表
  - World Wide Webは、http, URL, HTMLなど現在のWWWの基幹となる仕組みを全て備えた
  - 1993年に、CERNはWorldWideWebを無料で誰にでも開放することを発表



# WWWの歴史③ 普及と規格化

- ブラウザ戦争の勃発(1994～)
  - 様々な Web ブラウザの開発
    - NCSA Mosaic (1993～1997)
      - 画像の表示できるWebブラウザで大人気に
    - Netscape (1994～2008)
    - Internet Explorer (1995～2013)
  - ブラウザごとに独自の機能やタグを導入し始める→標準化の必要性

# WWWの歴史③ 普及と規格化

- World Wide Web Consortium(W3C)の設立(1994)
  - CSS, CGI, HTMLなどの規格を標準化
  - 中心組織は, MIT/**CSAIL**(MIT Computer Science and Artificial Intelligence Laboratory), **ERCIM**(the European Research Consortium for Informatics and Mathematics), **慶應義塾大学**

# WWWの特徴

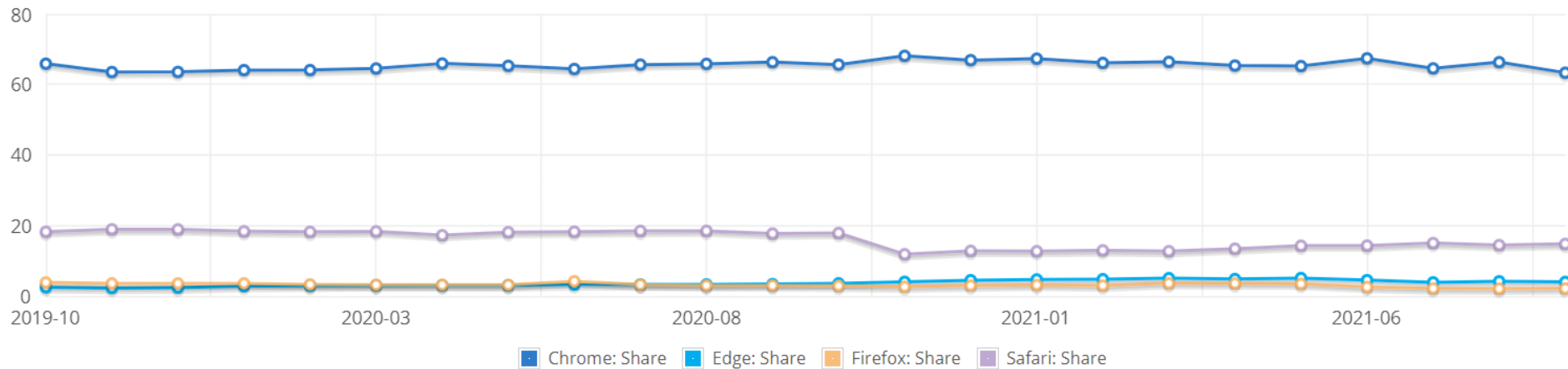
# WWWの特徴

- Web ページの閲覧には Web ブラウザ を使用
- ハイパーリンク によって複数の文書がインターネット上で結びついている
- ドキュメント記述言語に HTML を使用
- プロトコルには HTTP, HTTPS を使用
- それぞれの文書が URL をもつ

# Webブラウザ

- Web ページを閲覧するためのアプリケーションソフトのこと
- 単にブラウザと呼ばれることもある
- 例: Internet Explorer, Mozilla Firefox, Google Chrome, Safari, Opera, Microsoft Edge など

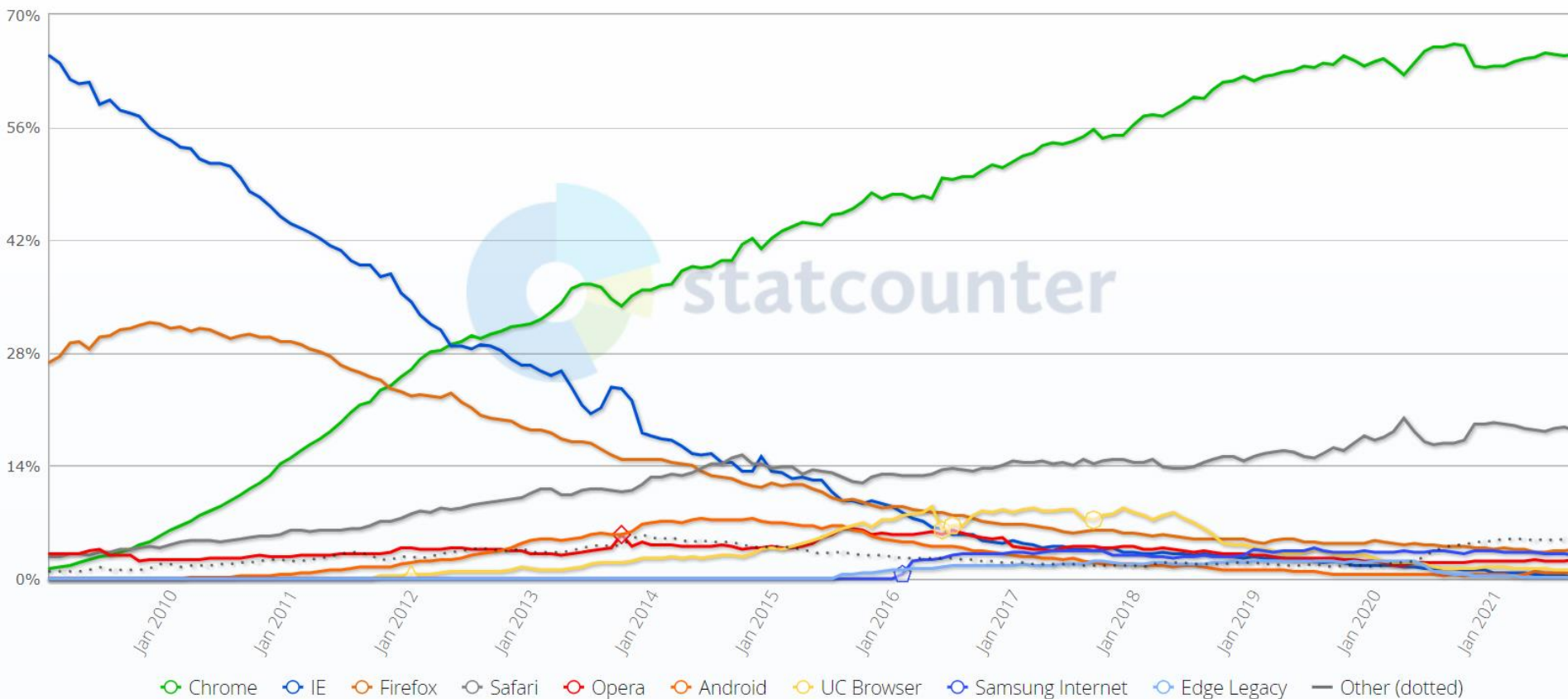
# Web ブラウザのシェア: NetMarketShare



引用元: Desktop Top Browser Share Trend October, 2019 to September, 2021

<http://marketshare.hitslink.com/>

# Web ブラウザのシェア: statcounter



引用元: Browser Market Share Worldwide January, 2009 to October, 2021

<https://gs.statcounter.com/>

# Web ブラウザのシェア率について

- Google Chrome がシェアを伸ばしている
- Firefox は勢いを失っている
- Internet Explorer のシェアは2016年まで急激に下降
- Safari のシェアは少しずつ伸びている
- Opera は停滞している



# Web ブラウザのシェア率について

- Web ブラウザのシェア率が調査によって異なる
  - 対象とする集団や母数が異なるため
    - NetMarketShare: NETMARKETSHARE社の全世界4万サイト以上のアクセス状況をもとに訪問者数を分析
    - StatCounter: StatCounterのトラッキングコードの入った全世界300万サイトのアクセス状況を元にページビューを分析

# Web ブラウザが数多く存在する理由

ブラウザ戦争などの歴史から考えられること

- セキュリティ問題

例: 1998年～2006年 第二次ブラウザ戦争

IE のセキュリティ問題が解決されずにいた頃, 新機能を搭載し, かつ IE のセキュリティ問題の原因を持たない Firefox や Opera などが登場

Windowsの標準ブラウザがIEからMicrosoft Edgeに変更

- 新機能
- 描画速度

# ハイパーリンク

- ハイパーリンク

- 文章内に埋め込まれた, 他の文書や画像などの位置情報
- 単に「リンク」とも呼ぶ
- リンクをクリックするだけで次の文書へ飛ぶことができる

- ハイパーテキスト

- 文章の中にハイパーリンクを埋め込み, 複数の文書を相互に結びつける仕組みのこと

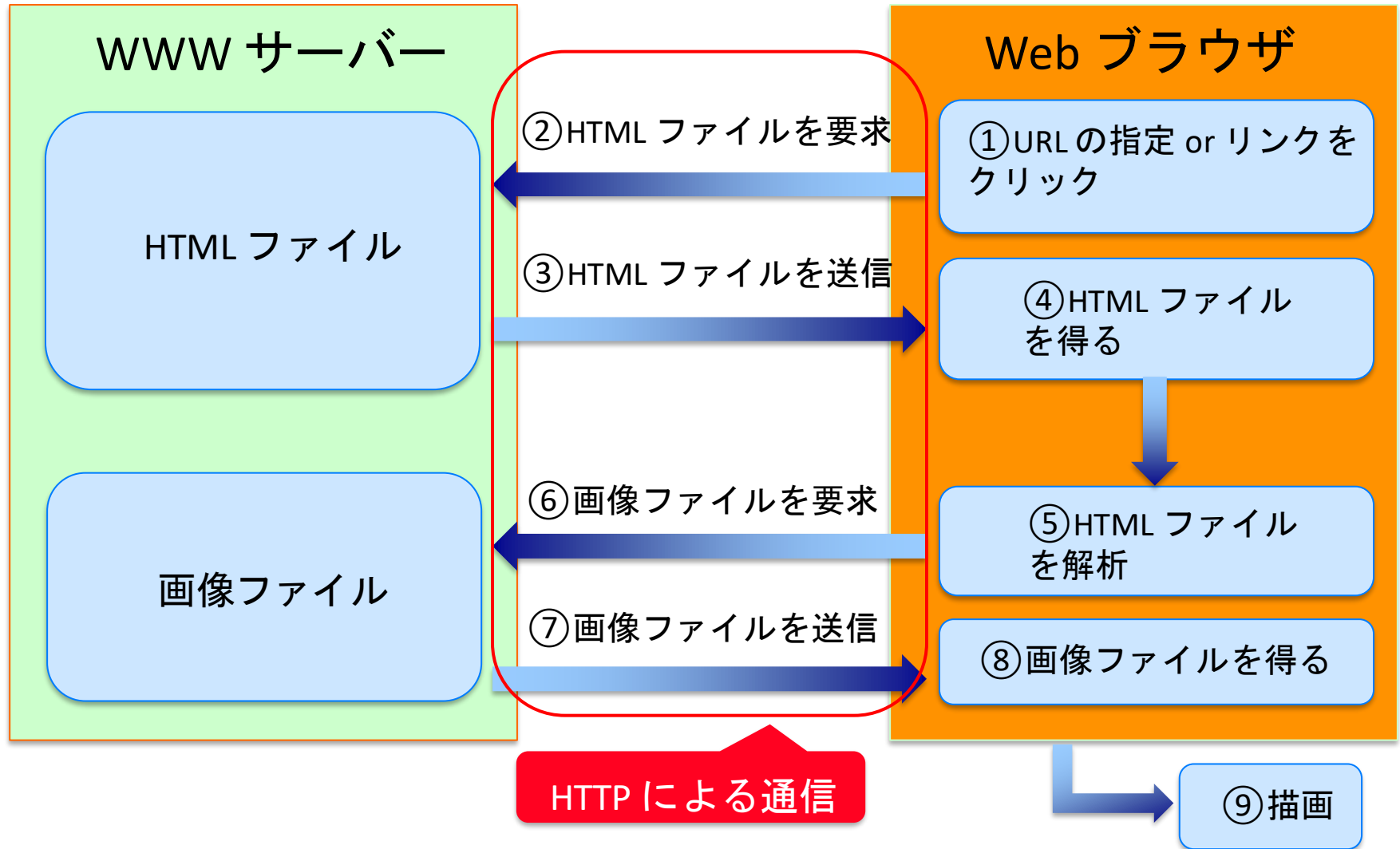
# HTML

- HTML (**H**yper **T**ext **M**arkup **L**anguage)
- Web ページを記述するためのマークアップ言語
- 文章の構造 (段落など) に関する指定を記述する
  - 見栄え (フォントサイズなど) に関する指定は CSS (Cascading Style Sheets) が担う
- 他の情報資源の位置情報 (ハイパーリンク) を文章内に書き入れることができる
- W3C (World Wide Web Consortium) が管理

# HTTP, HTTPS

- **HTTP** (**H**yper **T**ext **T**ransfer **P**rotocol)
  - ハイパーテキスト (HTML) 文書などをやりとりするための通信規約
  - 通常は 80 番ポートを使う
- **HTTPS** (**H**yper **T**ext **T**ransfer **P**rotocol over **SSL**)
  - HTTP に **SSL** によるデータの暗号化機能を付加した通信規約
  - 通常は 443 番ポートを使う

# HTTPでの通信



# HTTP, HTTPS

- **SSL/TLS** (**S**ecure **S**ocket **L**ayer/**T**ransport **L**ayer **S**ecurity)
  - セキュリティーを要求される通信のためのプロトコル
  - 暗号化された安全な通信を提供
  - 機密性の高い情報を安全にやりとりできるようにするために開発

# SSL/TLS の仕組み...の前に

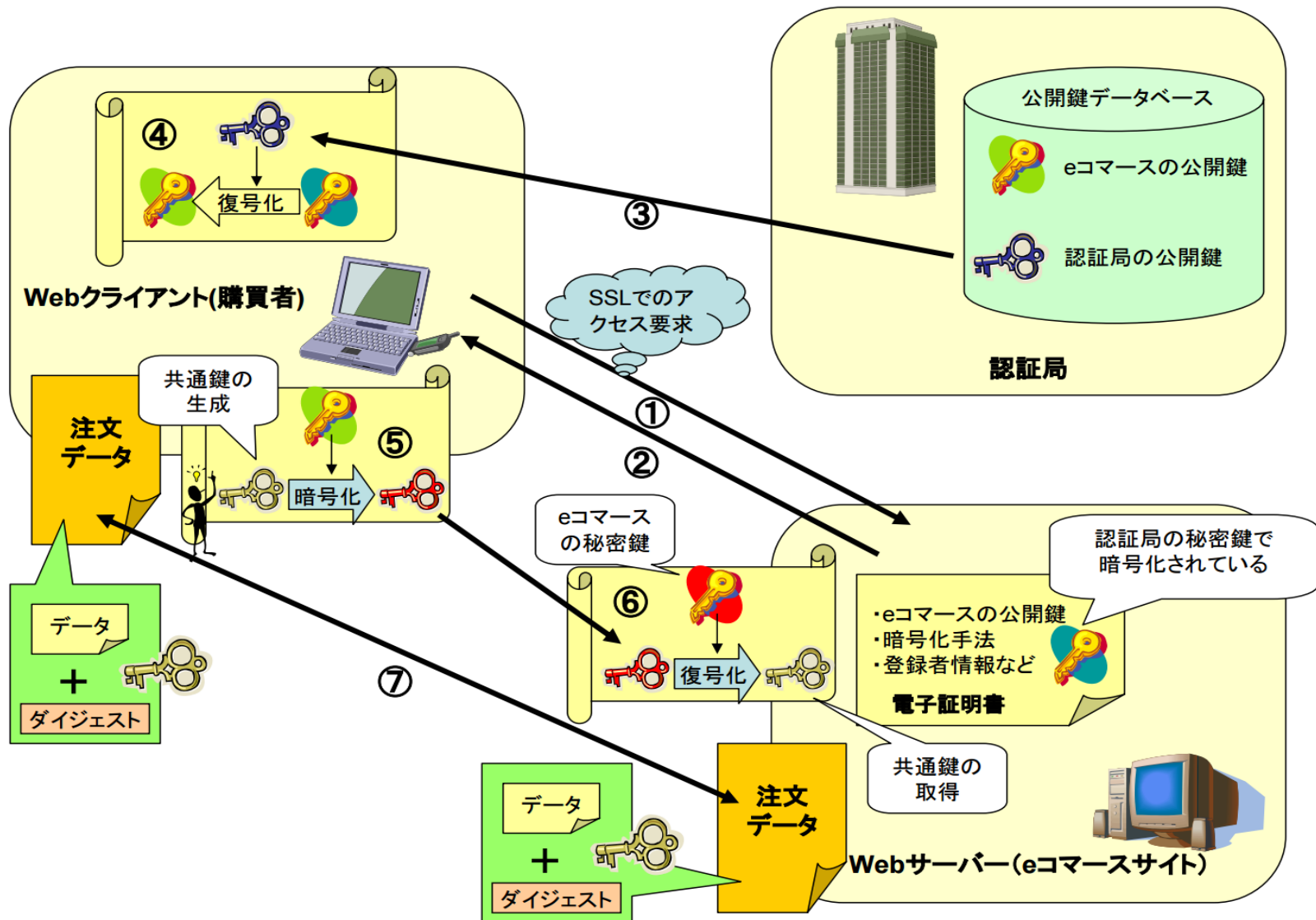
- 公開鍵認証
  - 秘密鍵と公開鍵のペアとして鍵を生成
    - 公開鍵で暗号化されたものはペアの秘密鍵でしか復号できない
    - 公開鍵をWeb上に置く or 相手に送信し, それを用いて暗号化してもらい, 秘密鍵で復号する
- 電子署名 (デジタル署名)
  - メッセージを書いた人がその人自身であることを証明するためのもの
  - ハッシュ関数の計算結果(ダイジェスト)を比較し, 一致していればその署名が正しいことも保証



# SSL/TLS の仕組み...の前に

- 電子証明書
  - 送信者が確実に存在することを保証してくれるもの
    - 信用できるWebサーバであるかどうかを証明しているのではない
  - 第三者機関（認証局＝CA:Certificate Authority）が公開鍵に対して電子署名を行うことにより、公開鍵の所有者の身元を保証
  - 電子証明書の中身は、認証を依頼した本人の公開鍵、暗号化手法や登録者情報など

# SSL/TLSの仕組み



画像元:

[http://www.ibm.com/developerworks/jp/websphere/library/web/web\\_security/pdf/2\\_6.pdf](http://www.ibm.com/developerworks/jp/websphere/library/web/web_security/pdf/2_6.pdf)

# SSL/TLSの仕組み

- ①クライアントがSSL/TLSでのアクセスをサーバに要求
- ②Webサーバは, 電子証明書をクライアントに送信. 電子証明書は, 認証局の秘密鍵で暗号化されている.
- ③クライアントは, 受け取った電子証明書から, その中に登録されている認証局にアクセスして認証局の公開鍵を入手.
- ④クライアントは, 入手した認証局の公開鍵を使用して電子証明書を復号し, Webサーバの公開鍵を入手.
- ⑤クライアントは一時的な共通鍵を生成して, それを取得したWebサーバの公開鍵で暗号化して, Webサーバに送信.
- ⑥Webサーバは, クライアントから暗号化された共通鍵を受け取り, Webサーバが持っている自分の秘密鍵で復号して共通鍵を入手する.
- ⑦ここから同じ共通鍵を使用して, クライアントとWebサーバとの暗号化通信を開始.

# URL

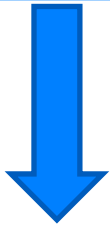
- URL (**U**niform **R**esource **L**ocator)
- 情報資源の場所を示している記述方式
- 「通信方法」と「住所」を指定している

http://

itpass.scitec.kobe-u.ac.jp

exp/fy2012

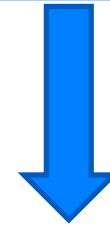
vow.html



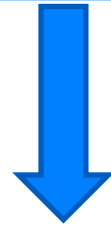
通信方法



サーバ名



ディレクトリ名

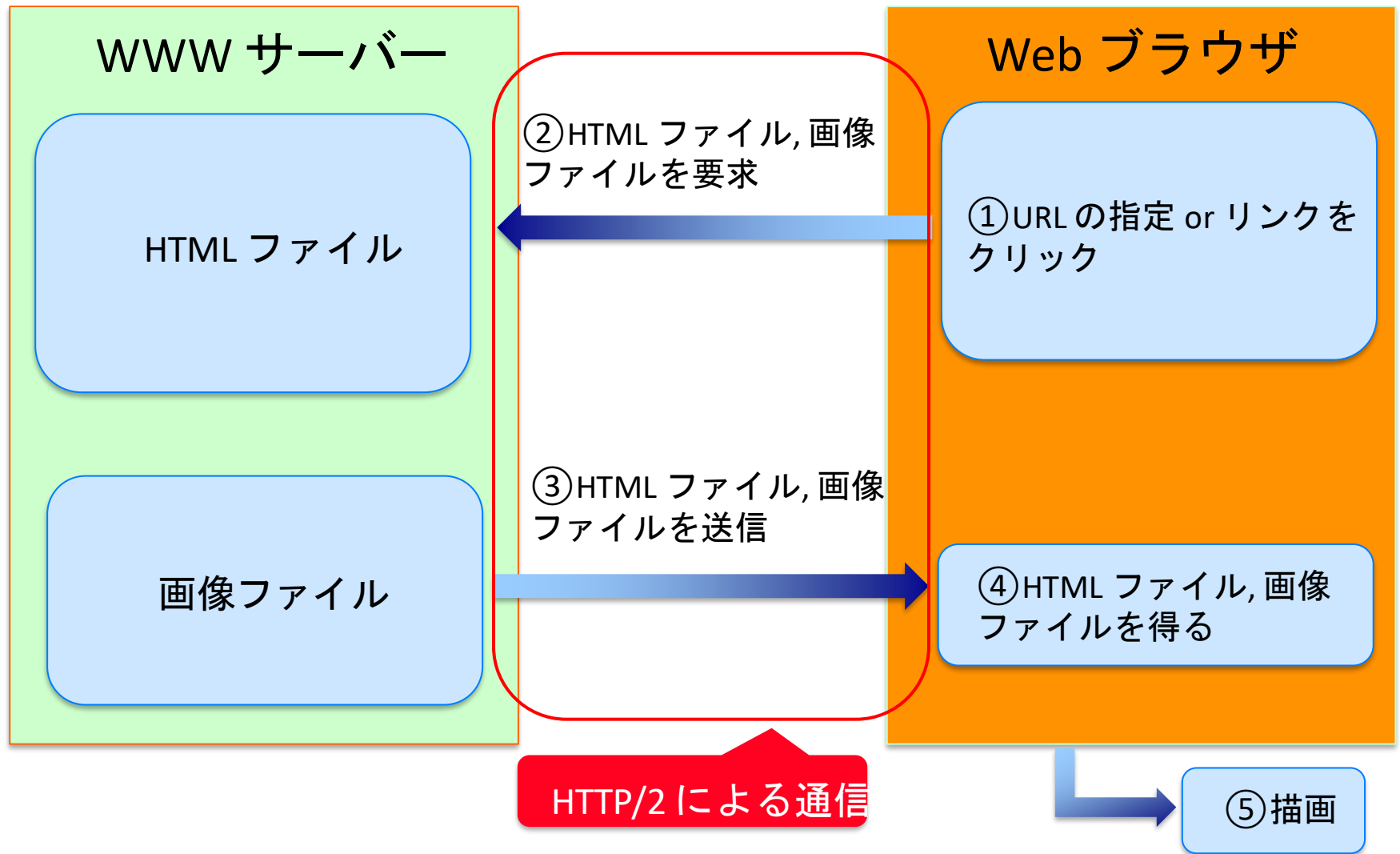


ファイル名

# HTTP/2

- **HTTP/2** (Hyper Text Transfer Protocol version 2)
  - 2015 年に正式に承認され, 公開された HTTP の新しいバージョン
  - 従来 of HTTP の効率化と速度向上を目指して開発された
  - 一度に複数のリクエストを同時に処理することが出来るため, 通信速度が向上した
  - 以前まで使われていた HTTP と完全な後方互換性がある

# HTTP/2での通信



# HTTP/2 の注意点

- SSL/TLS による暗号化が行われた HTTPS の通信上でしか HTTP/2 を使えないブラウザもある
- 全てのサイトが HTTP/2 で高速化されるわけではない
  - 例えば, もともとコンテンツが少ないサイトやコンテンツを複数ドメインに分散して保存しているウェブページなど

# 前半のまとめ（WWWの特徴）

- Web ページの閲覧には Web ブラウザ を使用
- ハイパーリンク によって複数の文書がインターネット上で結びついている
- ドキュメント記述言語に HTML を使用
- プロトコルには HTTP, HTTPS を使用
- それぞれの文書が URL をもつ



WWWサーバー

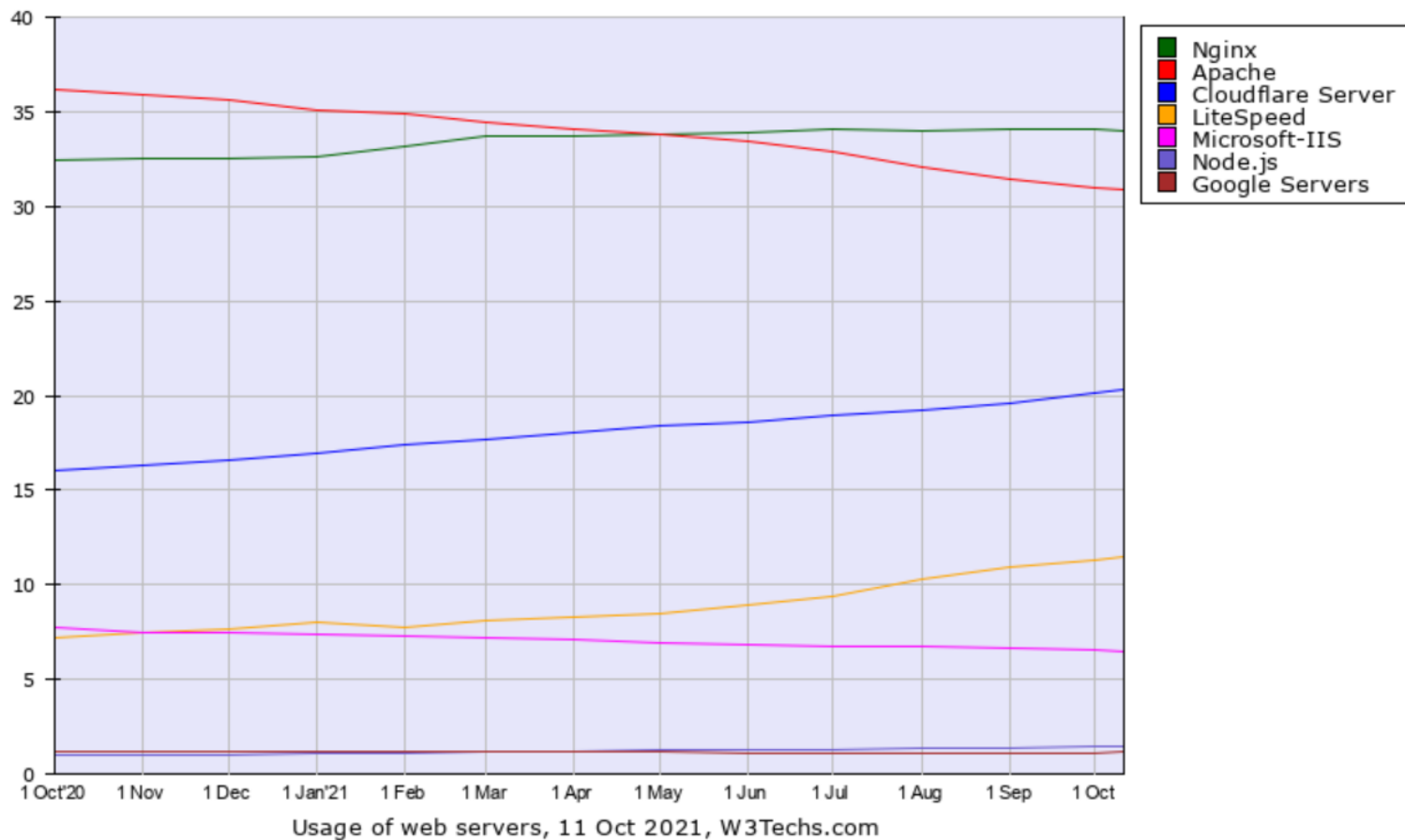
# WWWサーバの仕事

- 様々な情報資源を格納, 配信
  - HTML ファイル, 画像ファイルなど
    - クライアントの要求に応じてファイルを配信
- **静的処理**
- プログラム
  - クライアントの要求に応じてサーバ側でプログラムを実行し, その結果を配信
- **動的処理**
- WWWサーバソフトウェアをインストールすることでWWWサーバとして機能する

# WWWサーバソフトウェア

- **Apache** (Apache HTTP Server)
  - 2021年5月まで世界中で最も使われていたサーバソフトウェア
  - オープンソースソフトウェア
  - カスタマイズが容易
    - 最低限の機能: Apache コア, ApacheAPI
    - 付加的な機能: Apache モジュール
  - 多彩な動作環境
    - Linux, Windows, Macなどで動く
- **nginx**
  - オープンソースソフトウェア
  - 最近シェアを伸ばし、世界中で最も使われているサーバソフトウェアとなった
    - 特にアクセス数の多いサイトでのシェアが高い

# WWWサーバソフトウェアのシェア



引用元 : W3 Techs.com

Usage of web servers for websites

[https://w3techs.com/technologies/history\\_overview/web\\_server](https://w3techs.com/technologies/history_overview/web_server)

# Hiki

- Ruby で書かれた Wiki クローンの一種
  - Wikiクローン：  
オリジナルの Wiki (WikiWikiWeb) から派生したソフトウェア
- Web ブラウザを用いて文書の作成・編集が可能
- アクセス制限が可能
- プラグインによる機能拡張
  - プラグイン：  
ソフトウェアに追加され、その機能を拡張するようなプログラム
- CSS を使ったテーマ機能
  - CSS (Cascading Style Sheet) : HTMLの「見栄え」を定義する
  - tDiary用の豊富なテーマがある
- CGI を利用し WWW サーバと連携して動く

# CGI(Common Gateway Interface)

- 動的処理のサービスを提供する際使われる手法のひとつ
  1. ブラウザからの要求を受けたサーバが対応するプログラムを起動
  2. プログラムの実行結果をブラウザ側に返す
- プログラムはRuby, Perl, Java, C言語などでよく作成されている

# CGI(Common Gateway Interface)

```
# /usr/local/apache2/bin/apachectl  
次に,  
  
$ ps aux  
を実行すると、画面左端に  
とがわかった。  
  
itpass ディレクトリに public_html  
前のディレクトリを作成した。その中  
ブラウザで当該 URL にアクセスし、その  
確認した。  
URL に記述するとき、“http://...”と  
ころ、どちらの場合もファイルの内容が  
  
apache の設定ファイルに記述されてい  
るなど、  
denied  
to check  
れた。  
test ディレクトリ内の .uim.d というラ  
イファイル
```

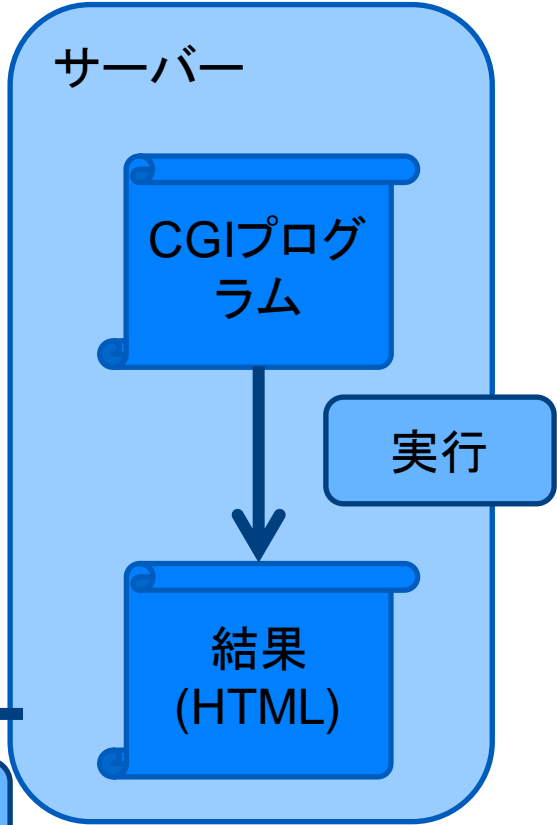
プレビュー 保存  タイムスタンプを更新する

キーワード(1行に1つ記述してください)

保存ボタンがプログラムにリンク

保存

実行結果をブラウザに返す



# 後半のまとめ（WWWサーバ）

- WWWサーバは, 情報資源の格納, 配信を行う
- ITPASS サーバでは, WWWサーバソフトウェアとして **Apache** を使う
- **Hiki** ではウェブ上で文書の作成・編集が可能
- **CGI** を使うことでサーバの動的処理が可能になっている



# 参考資料

- 2020年度ITPASSセミナー「WWWのおはなし」丸山響, 杉本小都
  - [https://itpass.scitec.kobe-u.ac.jp/seminar/lecture/fy2020/201016/pub/www\\_201016.pdf](https://itpass.scitec.kobe-u.ac.jp/seminar/lecture/fy2020/201016/pub/www_201016.pdf)
- IT用語辞典 e-Words
  - <http://e-words.jp/>
- ブラウザのシェアを、3つの情報源で比較してみる。
  - <https://language-and-engineering.hatenablog.jp/entry/20140530/BrowserShareStatCounterVsNetApplications>
- 2021年5月のWebサーバ利用シェア、「Nginx」が「Apache」を初めて上回る
  - <https://atmarkit.itmedia.co.jp/ait/articles/2105/17/news053.html>
- Apacheとnginxどちらを採用すべきかメリット・デメリット比較
  - <https://qiita.com/pink/items/7709218310b5cf11eabe>