

Slide URL

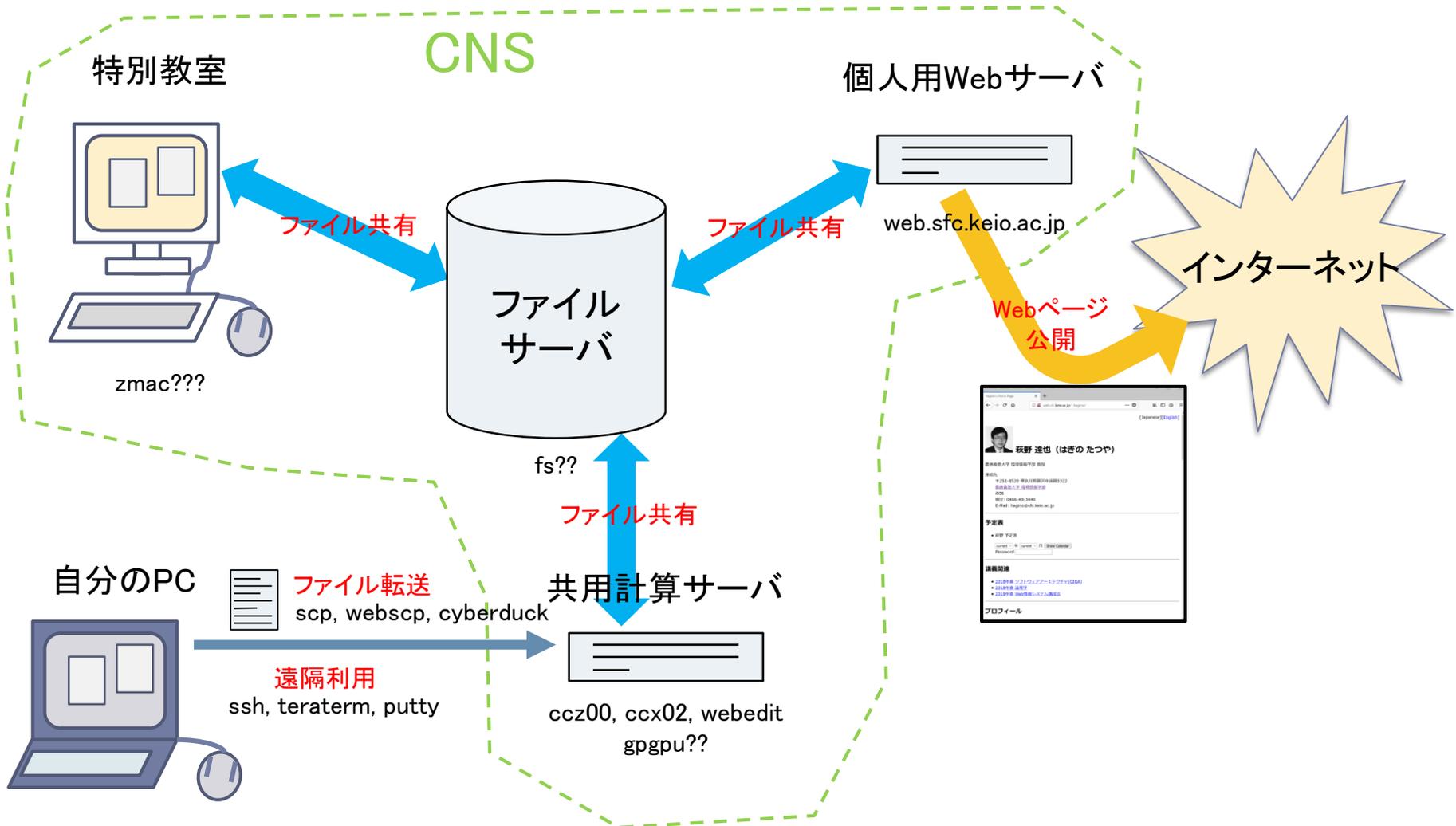
<https://vu5.sfc.keio.ac.jp/slide/>

# Web情報システム構成法

## 第4回 Webサーバ

萩野 達也 (hagino@sfc.keio.ac.jp)

# 前回課題：CNSにWebページを置きなさい



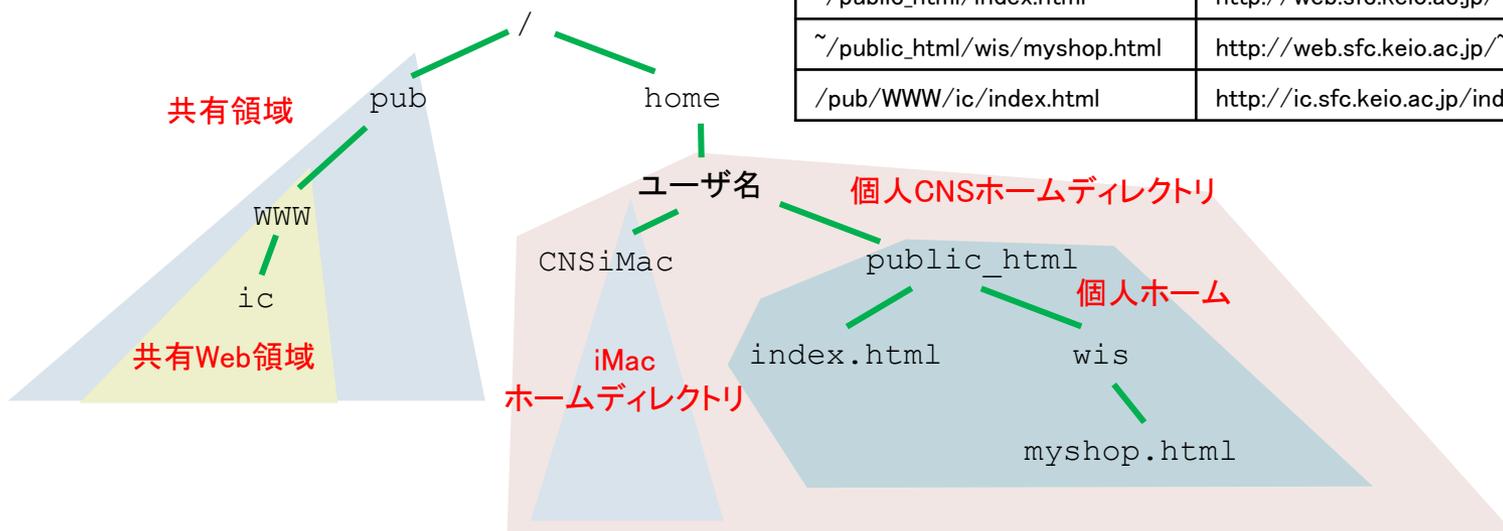
# CNSファイルサーバ

- ▶ CNS内でファイルを共有している
  - ▶ CNSのどのPCでも同じ環境, 同じファイルを利用できる
  - ▶ 個人のホームディレクトリ
    - ▶ 利用者ごとに割り当てられる
    - ▶ 一部が個人Webページとして公開される
  - ▶ 共有ファイル領域
    - ▶ プロジェクトなどで共有して利用
    - ▶ プロジェクトなどのWebとして公開可能

~ は個人のホームディレクトリ  
 ~ユーザ名  
 /home/ユーザ名

## ファイルサーバとURLの対応

パス名	Web URL
~/abc	非公開
~/public_html/index.html	http://web.sfc.keio.ac.jp/~ユーザ名/index.html
~/public_html/wis/myshop.html	http://web.sfc.keio.ac.jp/~ユーザ名/wis/myshop.html
/pub/WWW/ic/index.html	http://ic.sfc.keio.ac.jp/index.html

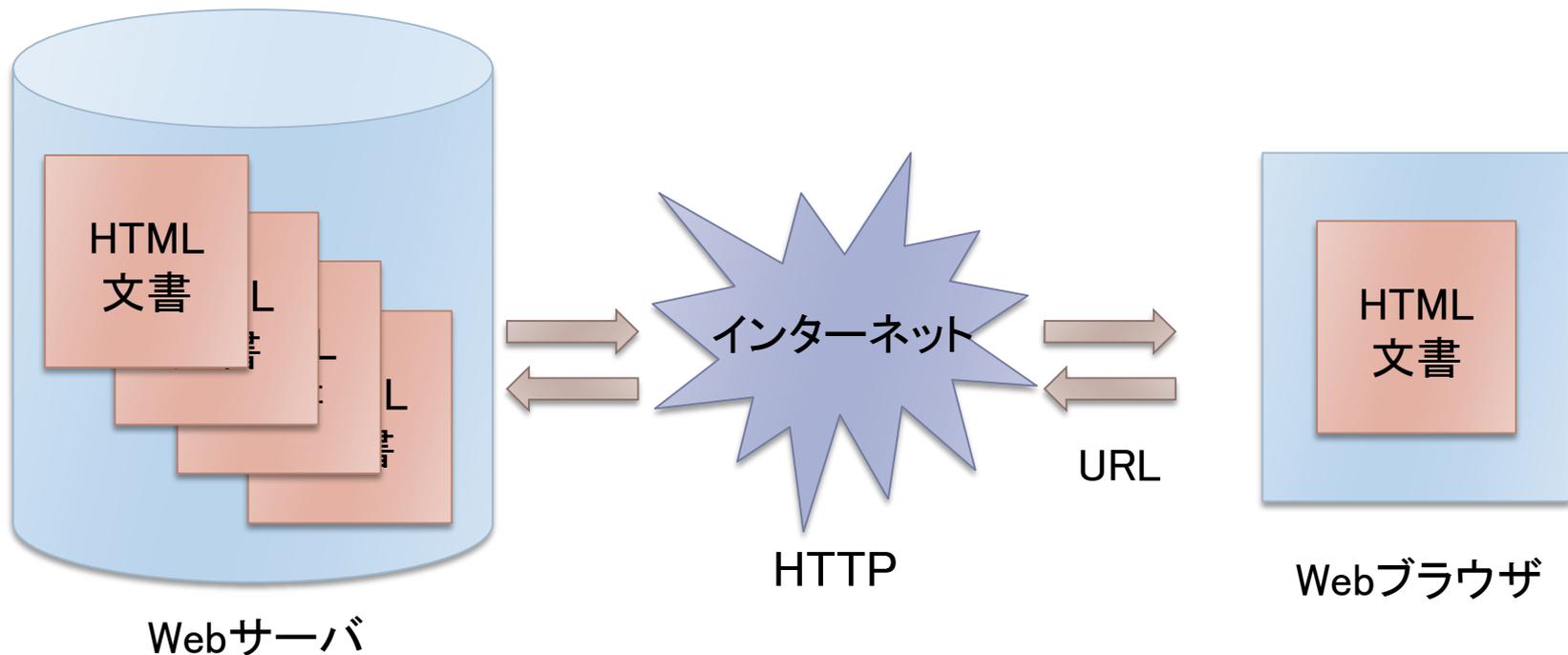




# Webの基本3要素

---

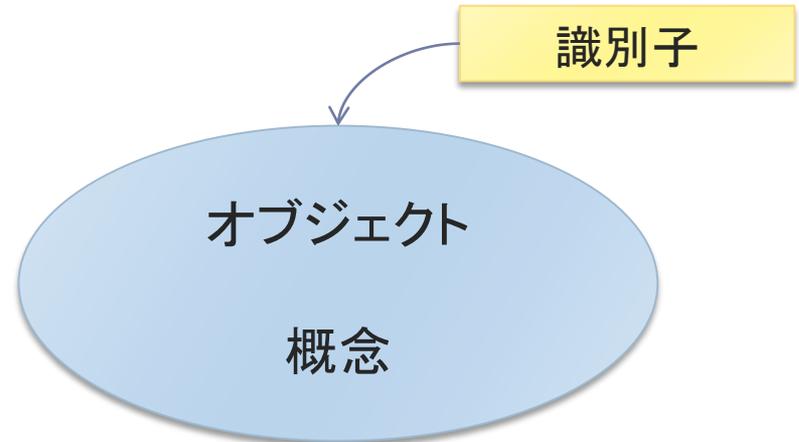
- ▶ 文書をHTMLとCSSで記述する.
- ▶ 文書の位置をURLで指し示す
- ▶ 文書をHTTPで取得する



# URI (Uniform Resource Identifier)

---

- ▶ 識別子
  - ▶ モノ(オブジェクト)や概念を名前や数字で識別する
- ▶ 身近な識別子:
  - ▶ 学生を識別
  - ▶ 本を識別
  - ▶ 電話を識別
  - ▶ PCを識別
  - ▶ インターネットでの識別



# URL • URN • IRI

---

## ▶ URL (Uniform Resource Locator)

- ▶ 資源の場所を識別するURI
- ▶ 例: http, ftp, file, mailto

## ▶ URN (Uniform Resource Name)

- ▶ urn:<nid>:<nss>
- ▶ NIDはIANAに登録
- ▶ <http://www.iana.org/assignments/urn-namespaces/urn-namespaces.xml>
- ▶ 50 URNが登録されている(2014-04-17)
- ▶ 例: 3gpp, cablelabs, cgi, clei, dgiwg, dvb, ebu, epc, epcglobal, eurosystem, example, fdcc, fipa, geant, gsma, ietf, iptc, isan, isbn, iso, issn, ivis, liberty, mace, mpeg, nbn, nena, newsml, nfc, nzl, oasis, ogc, ogf, oid, oipf, oma, pin, publicid, s1000d, schac, service, smpte, swift, tva, uci, ucode, uuid, web3d, xmlorg, xmpp

## ▶ IRI (Internationalized Resource Identifier)

- ▶ 国際化されたURI
- ▶ URIではアスキー文字しか使うことができない
- ▶ IRIではユニコード文字を使うことができる

# URIのシンタックス

---

`http://www.sfc.keio.ac.jp/teacher/hagino.html?title=web#lecture`

スキーマ

オーソリティ

パス

問い合わせ

フラグメント

## ▶ スキーマ

- ▶ URIの種類
- ▶ プロトコル

## ▶ オーソリティ

- ▶ ホスト名
- ▶ サーバ名

## ▶ パス

- ▶ オーソリティ内での場所
- ▶ ファイルのパス

## ▶ 問い合わせ

- ▶ 問い合わせパラメータ
- ▶ インタラクション

## ▶ フラグメント

- ▶ 文書内での位置

# URIに関する公理

---

- ▶ 普遍性(Universality)
  - ▶ すべてのWeb資源はURIを持つ
- ▶ 大域性(Global Scope)
  - ▶ URIはどこでも同じ意味を持つ
  - ▶ 一意性
- ▶ 同一性(Sameness)
  - ▶ URIは常に同じものを意味する
  - ▶ 意味が同じであり、内容は異なることもある
- ▶ 不透明性(Opacity)
  - ▶ URIだけから資源の種類を知ることはできない
  - ▶ 資源の表現を調べないと種類等はわからない

# Webにおける概念・識別子・表現

## ▶ 概念

- ▶ Web資源(Resource)

## ▶ 識別子

- ▶ URI (Uniform Resource Identifier)

## ▶ 表現(Representation)

- ▶ HTML+CSS
- ▶ XML
- ▶ RDF
- ▶ GIF

<http://www.keio.ac.jp/index.html>

識別

資源

慶應義塾大学

表現

```
<!DOCTYPE html>
<html>
  <head>
    <title>慶應義塾大学</title>
  </head>
  <body>
    <h1>慶應義塾大学</h1>
    ...
    ...
  </body>
</html>
```

# URIの利用

---

## ▶ Webページの場所

### ▶ URL

[http://www.sfc.keio.ac.jp/about\\_sfc/video.html](http://www.sfc.keio.ac.jp/about_sfc/video.html)

## ▶ 仕様

### ▶ DTD

<http://www.w3.org/TR/html4/loose.dtd>

### ▶ RFC

<urn:ietf:rfc:2141>

## ▶ Web資源

### ▶ 人

<http://ja.dbpedia.org/page/東京都>

### ▶ 都市

<http://ja.dbpedia.org/page/安倍晋三>

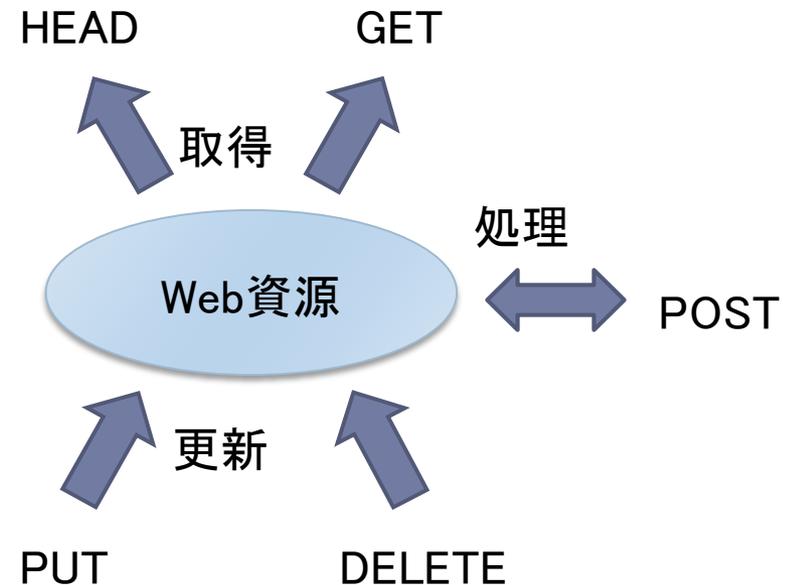
# HTTP (Hypertext Transfer Protocol)

---

## ▶ Web資源を操作するプロトコル

### ▶ 5つの主なメソッドを持つ

- ▶ HEAD
  - ▶ 資源の情報を得る
- ▶ GET
  - ▶ 資源の表現を取得する
- ▶ PUT
  - ▶ 資源を作成あるいは更新する
- ▶ DELETE
  - ▶ 資源を削除する
- ▶ POST
  - ▶ データを処理するために送る



# HTTPとFTPの違い

---

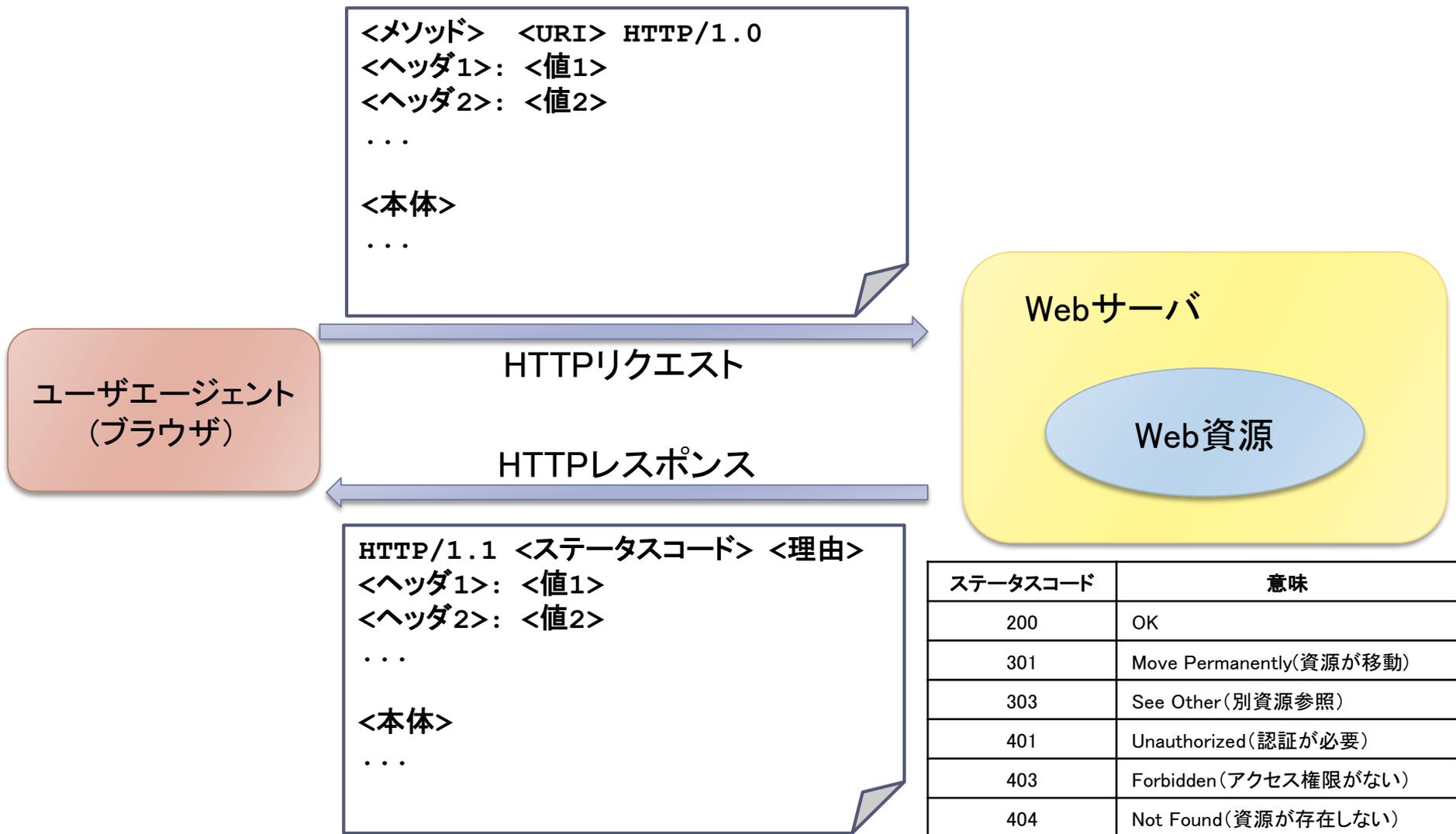
## ▶ FTP

- ▶ 遠隔ファイル操作
- ▶ インターネットの初期の段階から利用
- ▶ ユーザは認証のためFTPサーバにログインする
- ▶ フリーソフトなどの配布のためにAnonymous FTPが存在
- ▶ 制御とデータ転送のための2つのTCPコネクションを利用
- ▶ データの種類は2つのみ:テキスト・バイナリ

## ▶ HTTP

- ▶ Web資源の操作
- ▶ 認証は原則行わない
- ▶ 一つのTCP接続を利用
- ▶ マルチメディアをサポート

# HTTPリクエストとレスポンス



# GETとHEADメソッド

## ▶ GET

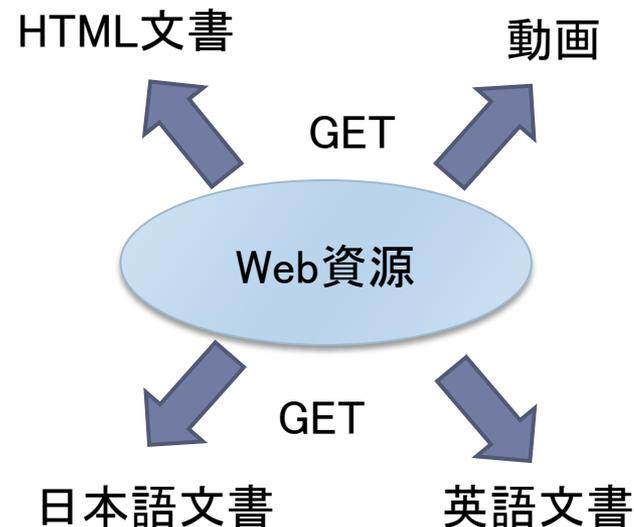
- ▶ 資源の一つの表現を取得する
- ▶ コンtentネゴシエーション
- ▶ 言語ネゴシエーション

## ▶ HEAD

- ▶ 資源およびその表現のに関する情報を取得する
- ▶ GETの一部

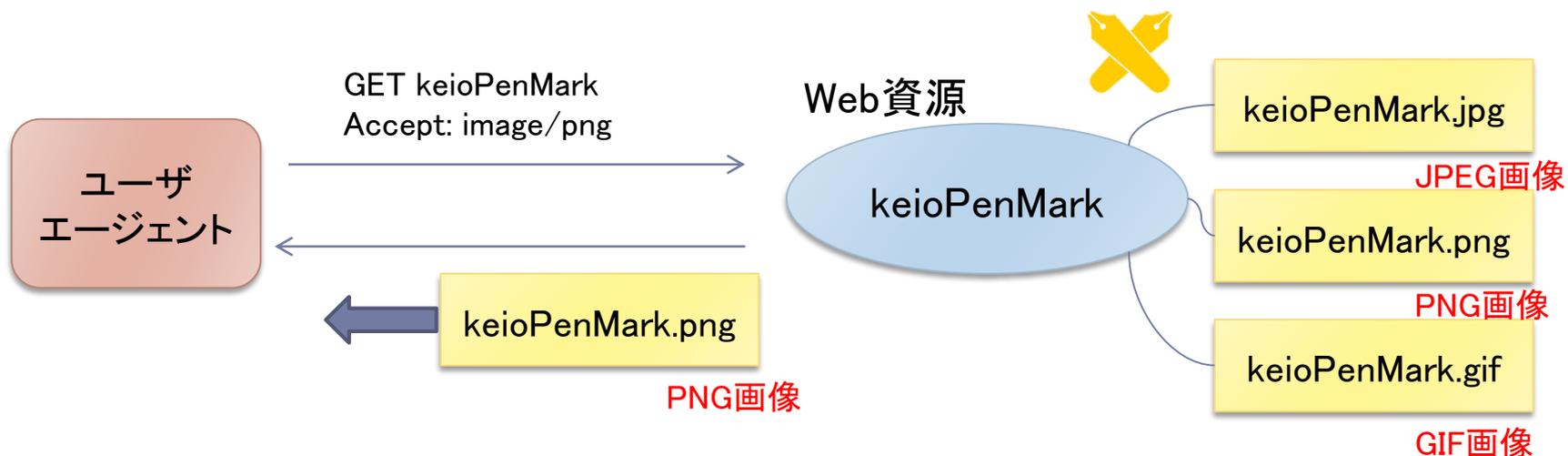
## ▶ GETの性質

- ▶ GETは何度行っても安全
- ▶ GETは冪等 (Idempotent)
- ▶ GETは副作用がない

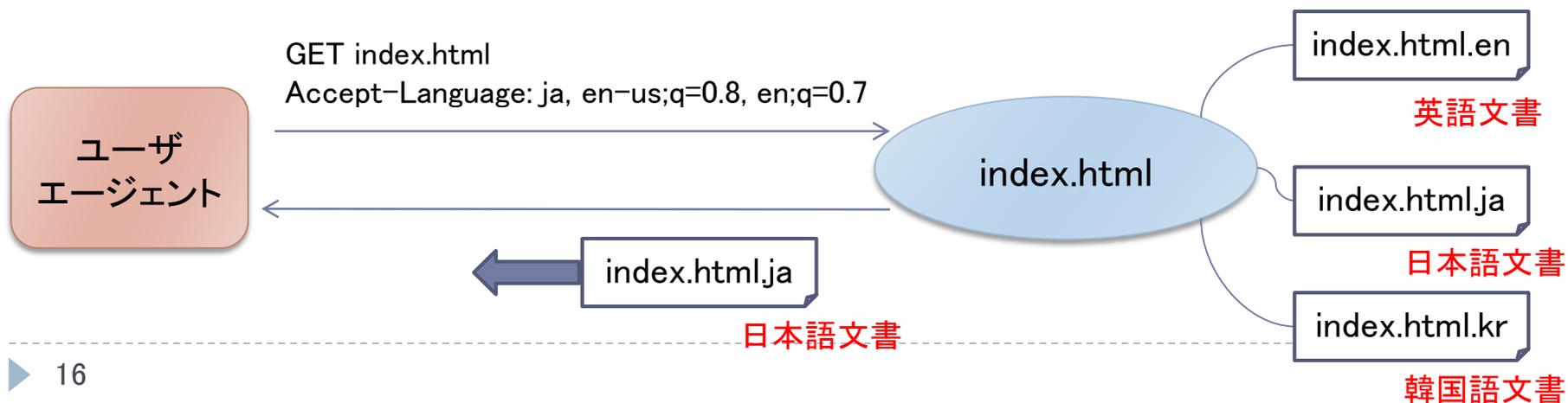


# Contentネゴシエーション

## ▶ 文書/画像形式



## ▶ 言語



# PUTとPOSTメソッド

## ▶ PUTメソッド

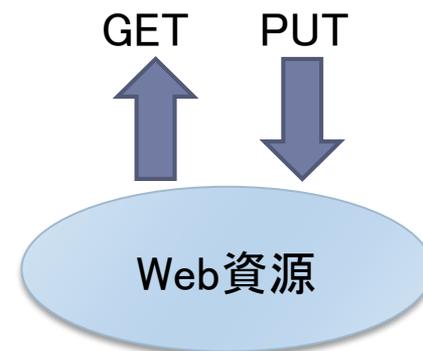
- ▶ 資源を作成あるいは更新する
- ▶ GETの逆
- ▶ 通常のブラウザはPUTを使わない

## ▶ POSTメソッド

- ▶ 資源にデータを送る
- ▶ 資源はデータ进行处理する
- ▶ HTMLのFORMで利用

## ▶ GET vs POST

- ▶ FROMのmethod属性によりGETまたはPOSTを指定
- ▶ 副作用がない場合にGET
- ▶ 資源の更新など副作用があるときにはPOST
- ▶ POSTは幂等ではない



# Other Features of HTTP

---

## ▶ 転送

- ▶ 資源が別のところに移動

## ▶ 認証

- ▶ ユーザ名とパスワードによる認証
- ▶ BasicあるいはDigest認証

## ▶ Virtual Host

- ▶ 一つのサーバで複数のWebサイトをホスト
- ▶ DNSの別名を利用

## ▶ TCP/IPの効率的利用

- ▶ Persistent connection (keep-alive)
- ▶ パイプライン処理

## ▶ Proxyキャッシュ制御

- ▶ max-age
- ▶ no-cache
- ▶ public あるいはprivate

## ▶ WebDAVへの拡張

- ▶ COPY, MOVE, LOCK, UNLOCK

# 課題：多言語対応

---

- ▶ 架空のオンラインショップのトップページを日本語と英語の両方で準備しなさい。
  - ▶ 両方とも同じHTML5で記述すること。
  - ▶ できるだけ同じ内容にしなさい。
  - ▶ 日本とのトップページと英語のトップページをリンクして切り替えることができるようにしなさい。
- ▶ CNSでの設定
  - ▶ `myshop.html` を日英で用意する場合には、
    - ▶ `myshop.html` は作成しない
    - ▶ `myshop.html.ja` に日本語版
    - ▶ `myshop.html.en` に英語版
    - ▶ Webサーバがフォルダ内を探ることができるように、`other/`に`read`パーミッションを与える
- ▶ 提出
  - ▶ <https://vu5.sfc.keio.ac.jp/kadai/>
  - ▶ 作成したHTMLファイルをCNS上に置き、CNS内のみから見えるようにして下記のものを宿題登録システムから提出してください。
    - ▶ トップページの言語自動切換えURL
  - ▶ 締め切り: 5月13日正午

# HTML5によるページの構造化

```
<!DOCTYPE html>
<html lang="ja">
<head>
  <meta charset="UTF-8" />
  <title>なんでもショップ</title>
</head>
<body>
  <header>
    <h1>なんでもオンラインショッピング</h1>
  </header>
  <nav>
    <ul>
      <li><a href="goods1.html">商品1</a></li>
    </ul>
  </nav>
  <article>
    <p>この、オンラインショップでは、なんでも売っています。</p>
  </article>
  <footer>
    <p>問い合わせ: <a href="help.html">ヘルプ</a></p>
  </footer>
</body>
</html>
```

ショップのヘッダ  
ロゴなど

メニュー

このページの本文

フッター  
ヘルプナビなど



# まとめ

---

## ▶ URI

- ▶ Web資源の識別
- ▶ URL, URN, IRI

## ▶ HTTP

- ▶ Web資源を操作するプロトコル
- ▶ HTTP URI
- ▶ メソッド: HEAD, GET, PUT, DELETE, POST