

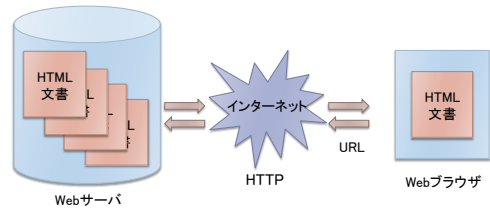
Web Information System Design No.5 Web文書操作

萩野 達也 (hagino@sfc.keio.ac.jp)

1

Webの基本要素

- ▶ 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



▶ 2

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



▶ 3

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, epglobal, eurosystem, example, fdc, fipa, geant, gsma, ietf, iptc, isan, isbn, iso, issn, ivis, liberty, mace, mpeg, nbn, nena, newxml, nfc, nzi, 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

▶ 4

URIのシンタックス

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

スキーマ オートリティ 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
- ▶ オートリティ
 - ▶ ホスト名
 - ▶ サーバ名
- ▶ パス
 - ▶ オートリティ内での場所
 - ▶ 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

▶ 5

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ができない
 - ▶ 資源の表現を調べないと種類等はわからない

▶ 6

Webにおける概念・識別子・表現

概念

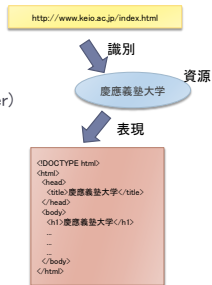
- Web資源(Resource)

識別子

- URI (Uniform Resource Identifier)

表現(Representation)

- HTML+CSS
- XML
- RDF
- GIF



▶ 7

URIの利用

Webページの場所

- URL

http://www.sfc.keio.ac.jp/about_sfc/video.html

仕様

- DTD
- RFC

<http://www.w3.org/TR/html4/loose.dtd>
[urn:ietf.rfc.2141](http://ietf.rfc.2141)

Web資源

- 人
- 都市

<http://ja.dbpedia.org/page/東京都>
<http://ja.dbpedia.org/page/安倍晋三>

▶ 8

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



▶ 9

HTTPとFTPの違い

FTP

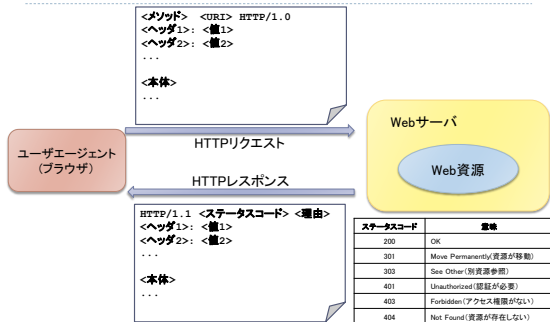
- 遠隔ファイル操作
- インターネットの初期の段階から利用
- ユーザは認証のためFTPサーバにログインする
- フリーソフトなどの配布のためにAnonymous FTPが存在
- 制御とデータ転送のための2つのTCPコネクションを利用
- データの種類の2つのみ: テキスト・バイナリ

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

▶ 10

HTTPリクエストとレスポンス



▶ 11

GETとHEADメソッド

GET

- 資源の一つの表現を取得する
- コンテンツネゴシエーション
- 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の一部分

GETの性質

- GETは何度行っても安全
- GETは幂等 (Idempotent)
- GETは副作用がない

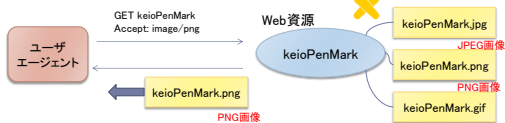


$$\text{GET} \times \text{GET} = \text{GET}$$

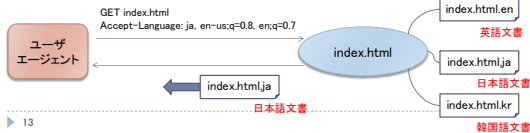
▶ 12

Contentネゴシエーション

▶ 文書/画像形式



▶ 言語



▶ 13

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
- ▶ 資源はデータを処理する
- ▶ 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



▶ 14

Other Features of HTTP

▶ 転送

- ▶ 資源が別のところに移動

▶ TCP/IPの効率的利用

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

▶ 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

▶ Proxyキャッシング制御

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

▶ 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

▶ WebDAVへの拡張

- ▶ COPY, MOVE, LOCK, UNLOCK

▶ 15

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

▶ 16

グループワーク

▶ グループワーク

- ▶ 3人一組のグループで共同で行うこと

▶ 課題「HTML6の提案」

- ▶ 現在のHTMLで不満に思っていることを挙げ、それを解決するHTML6を提案してください。
- ▶ HTMLの文書形式に関することでも、機能に関することでも、サーバやPHPのようなプログラミング環境に関することでも構いません。

▶ 準備するもの

- ▶ 5分間のプレゼンテーション
- ▶ SFO-SFSIに発表日の正午までにプレゼンテーションのPPTあるいはPDFをアップすること

▶ 発表

- ▶ 8月1日に各グループ5分で発表してもらいます
- ▶ 早慶戦で8月1日の授業がなかった場合には8月8日になります

▶ 注意

- ▶ プレゼンテーションのタイトルページに(貢献した)グループメンバーを書くこと

▶ 17