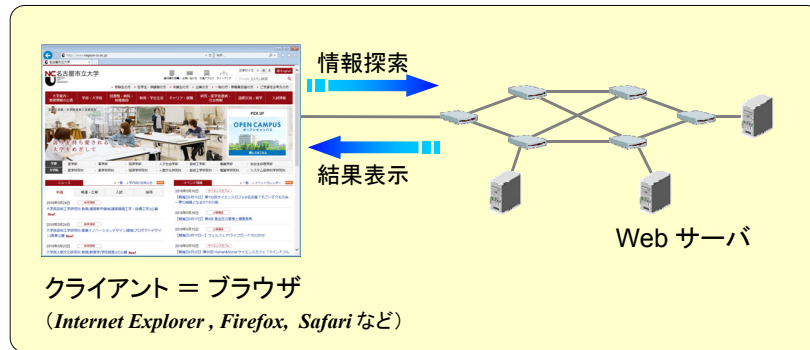


Web とは

インターネット上のハイパーテキスト

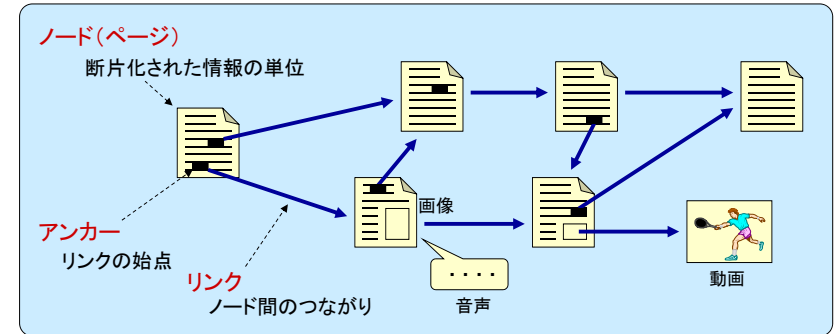


World-Wide Web

ハイパーテキスト

👉 気の向くままに情報にアクセスするための手段

- ✓ 1945年 Vannevar Bush が、論文 As We May Think にて提唱
- ✓ 1965年 Ted Nelson がハイパーテキストという用語を用いる



👉 百科事典、観光案内システム等のシステムが個別に構築された

Web の歴史と理念

CERNの研究者

Tim Berners-Lee によって1990年に提案

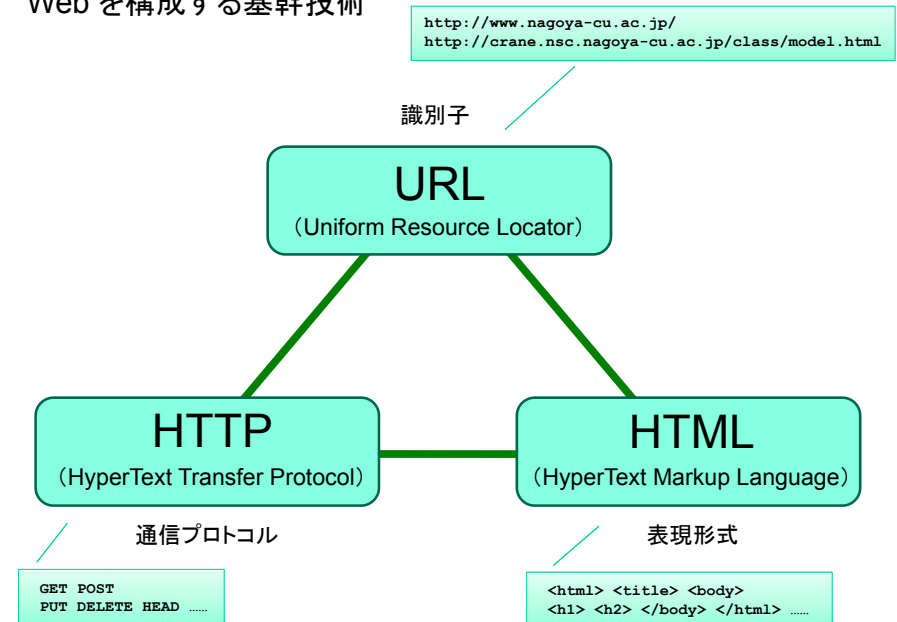
情報共有

インターネット上のハイパーテキスト

1993年、ブラウザ Mosaic の登場とともに世界中で普及しはじめる

リンクによる知のネットワーク化

Web を構成する基幹技術



URL ～ Uniform Resource Locator ～

リソース(情報)のありかを示す記述方法

`http://crane.nsc.nagoya-cu.ac.jp/class/ouyou.html`

スキーム

ホスト名

パス

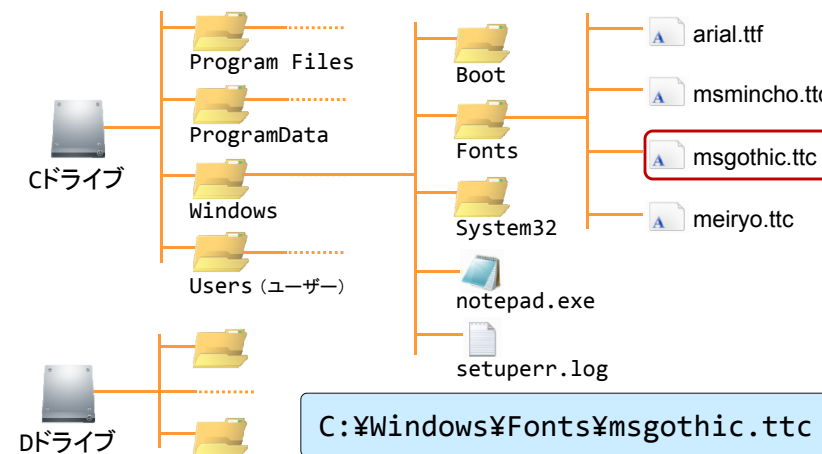
スキーム : アクセスプロトコル

ホスト名 : DNSで解決可能なホスト名 (もしくはIPアドレス)

パス : ホスト上のパス (フォルダ名、ファイル名)

ファイルパス (Path)

CDドライブのWindowsフォルダ, その中にFontsというフォルダがあって, そこにあるmsgothic.ttcというのが, “MSゴシック”のファイルです。



OSによるパス表記の違い

Windows

`C:¥Windows¥Fonts¥msgothic.ttc`

ドライブ名

フォルダ名

ファイル名

区切り記号は「¥」

UNIX系 (Linux, MacOS)

※ドライブ名という概念はない

`/usr/X11R6/ /fonts/gothic.ttf`

フォルダ名

ファイル名

区切り記号は「/」

パスとサーバにおけるフォルダ構造の関係 (1)

[システム自然科学研究科・総合生命理学部]

`http://www.nsc.nagoya-cu.ac.jp/`

ホスト名



宣伝などで見かけるURL

パスがない?

パスとサーバにおけるフォルダ構造の関係 (1)

[サイエンスカフェ]

`http://www.nsc.nagoya-cu.ac.jp/scicafe/`

ホスト名

パス = フォルダ名



宣伝などで見かけるURL

ファイル名?

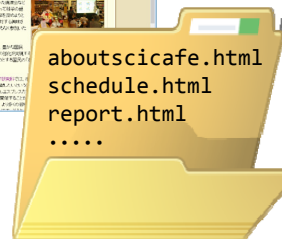
パスとサーバにおけるフォルダ構造の関係 (1)

[サイエンスカフェ → サイエンスカフェとは?]

`http://www.nsc.nagoya-cu.ac.jp/scicafe/` ^{ファイル}
`aboutscicafe.html`
`schedule.html`
`report.html`

ホスト名

フォルダ



サーバ上の「scicafe」フォルダ

パスとサーバにおけるフォルダ構造の関係 (1)

[サイエンスカフェ]

`http://www.nsc.nagoya-cu.ac.jp/scicafe/index.html`

ホスト名

省略されていた



サーバ上の「scicafe」フォルダ

パスとサーバにおけるフォルダ構造の関係 (3)

[宮原研究室]

`http://www.nsc.nagoya-cu.ac.jp/~miya/` ^{チルダ}

ホスト名

パス

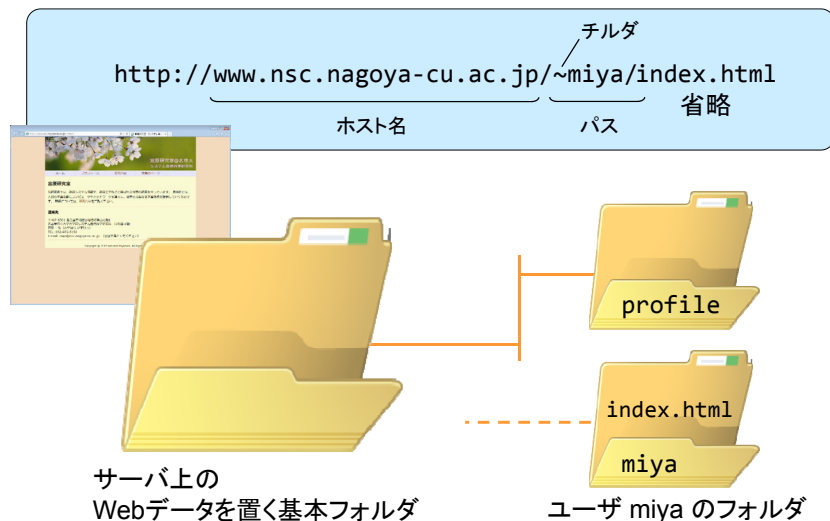


サーバ上の
Webデータを置く基本フォルダ



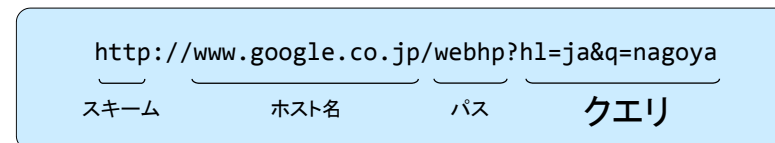
パスとサーバにおけるフォルダ構造の関係 (3)

[宮原研究室]



URL : 補足

クエリ : ブラウザからサーバにデータを渡す

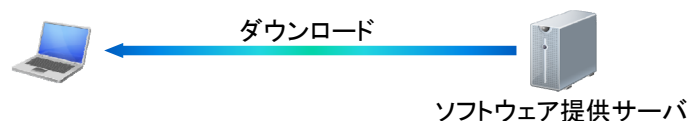


ファイル名の省略

- index.html : 最も一般的・・・か?
- index.htm : 拡張子が3文字だった時代の名残
- index.php : PHPプログラムを含んだHTMLファイル

http 以外のスキーム例

📁 ftp ~ *File Transfer Protocol* ~
ファイル転送のためのプロトコル (とそのソフトウェア群)



📁 mailto 電子メールアドレス
mailto:miya@nsc.nagoya-cu.ac.jp

📁 file ローカルシステム(手元のコンピュータ)上のファイル
file:///Windows/notepad.exe

📁 tel 電話番号
tel:0528725150

HTML ~ *HyperText Markup Language* ~

📄 Webページを記述するマークアップ言語

文書構造 (文書を構成する要素) を <タグ> によって記述する

<タグ> の例

- ```
<html> <head> <title> <meta> <script> <body>
<h1> <h2> <p> <hr>

<div>
```

## HTML のかんたん例 (1)

```

<html>
<head>
<title>宮原のホームページ</title>
</head>

<body>

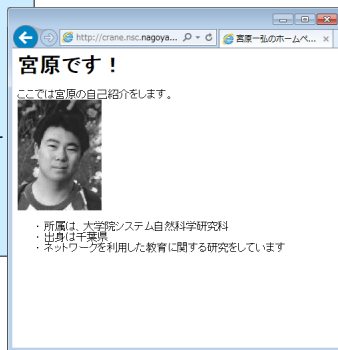
<h1>宮原です！</h1>
<p>ここでは宮原の自己紹介をします。

</p>

所属は、大学院システム自然科学研究科
出身は千葉県
ネットワークを利用した教育支援に関する研究をしています

</body>
</html>

```



※ ただし、構文は不完全

## HTML のかんたん例 (2)

```

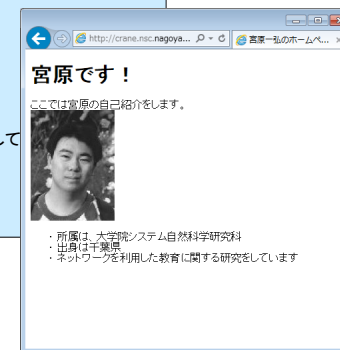
<!DOCTYPE html>
<html>
<head>
<title>宮原のホームページ</title>
<meta charset="Shift_JIS">
</head>
<body>
<h1>宮原です！</h1>
<p>
ここでは宮原の自己紹介をします。

</p>

所属は、大学院システム自然科学研究科
出身は千葉県
ネットワークを利用した教育支援に関する研究をして

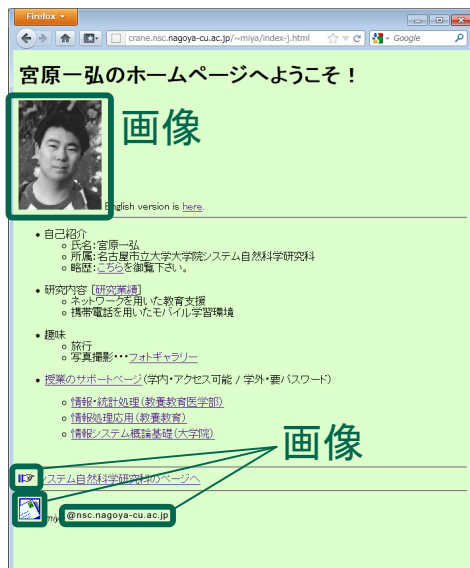
</body>
</html>

```



※ HTML 5 による完全な構文

## 実際の Web ページでは...



## 昔の宮原のページ

このWebページは  
HTMLファイル × 1  
画像ファイル × 4  
によって構成

## 実際の Web ページでは...



どのような構成か？

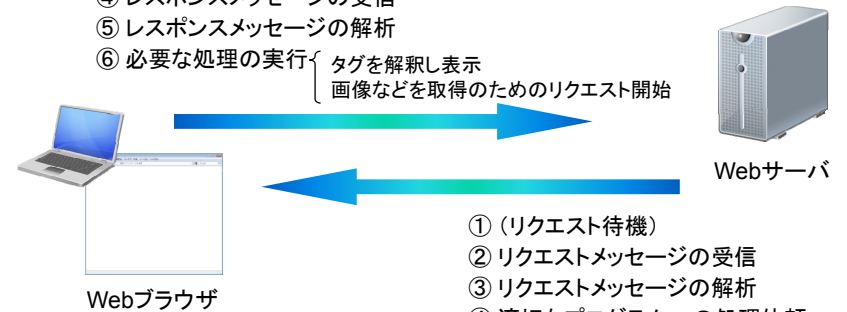
# HTTP ~ HyperText Transfer Protocol ~

リクエスト / レスポンス型のプロトコル



# リクエスト / レスポンス

- ① リクエストメッセージの作成
- ② リクエストメッセージの送信
- ③ (レスポンス待機)
- ④ レスポンスメッセージの受信
- ⑤ レスポンスメッセージの解析
- ⑥ 必要な処理の実行



- ① (リクエスト待機)
- ② リクエストメッセージの受信
- ③ リクエストメッセージの解析
- ④ 適切なプログラムへの処理依頼
- ⑤ プログラムから結果を取得
- ⑥ レスポンスメッセージの作成
- ⑦ レスポンスメッセージの送信

## リクエストメッセージ

http://crane.nsc.nagoya-cu.ac.jp/class/ouyou.html を取得

```
GET /class/ouyou.html HTTP/1.1
Accept: application/x-ms-application, image/jpeg, application/xaml+xml, image/gif,
image/pjpeg, application/x-ms-xbap, application/x-shockwave-flash,
application/vnd.ms-excel, application/msword, application/vnd.ms-powerpoint, */*
Accept-Language: ja-JP
User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1; WOW64; Trident/4.0;
SLCC2; .NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center
PC 6.0; HPDPTDF)
Accept-Encoding: gzip, deflate
Host: crane.nsc.nagoya-cu.ac.jp
Connection: Keep-Alive
```

リクエストライン (先頭の1行)

ステータスライン (先頭の1行)

ヘッダ

ボディ

リクエストライン サーバへのリクエストを記述

```
GET /class/ouyou.html HTTP/1.1
(メソッド=処理) (URL=パス) (プロトコルバージョン)
```

ヘッダ 名前:値という形式で付加的なデータを記述

## レスポンスメッセージ

```
HTTP/1.1 200 OK
Date: Wed, 12 May 2013 03:46:37 GMT
Server: Apache/2.0.55 (Unix) PHP/5.1.4
Last-Modified: Wed, 10 May 2013 03:46:31 GMT
ETag: "7d2f-476d-7cd113c0"
Accept-Ranges: bytes
Content-Length: 18285
Keep-Alive: timeout=15, max=100
Connection: Keep-Alive
Content-Type: text/html

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd"> <html>
<head>
<title>情報処理応用(2018年度・前期)</title>
```

ステータスライン サーバのリクエスト処理状況を記述

```
HTTP/1.1 200 OK
(プロトコルバージョン) (ステータスコード) (テキストフレーズ)
```

ヘッダ 名前:値という形式で付加的なデータを記述

ボディ HTMLなどのデータ本体

## ステートレスなHTTP



リクエスト / レスポンスは1回で完結

