

通信プロトコル実装実習

HTTPプロトコル

2022 年 月 日 ()
中国職業能力開発大学校
1号棟4階 1405室

WWW (World Wide Web)

🖱️ Webブラウザが入力するWebサーバへの道筋をURL(Uniform Resource Locator)と呼ぶ

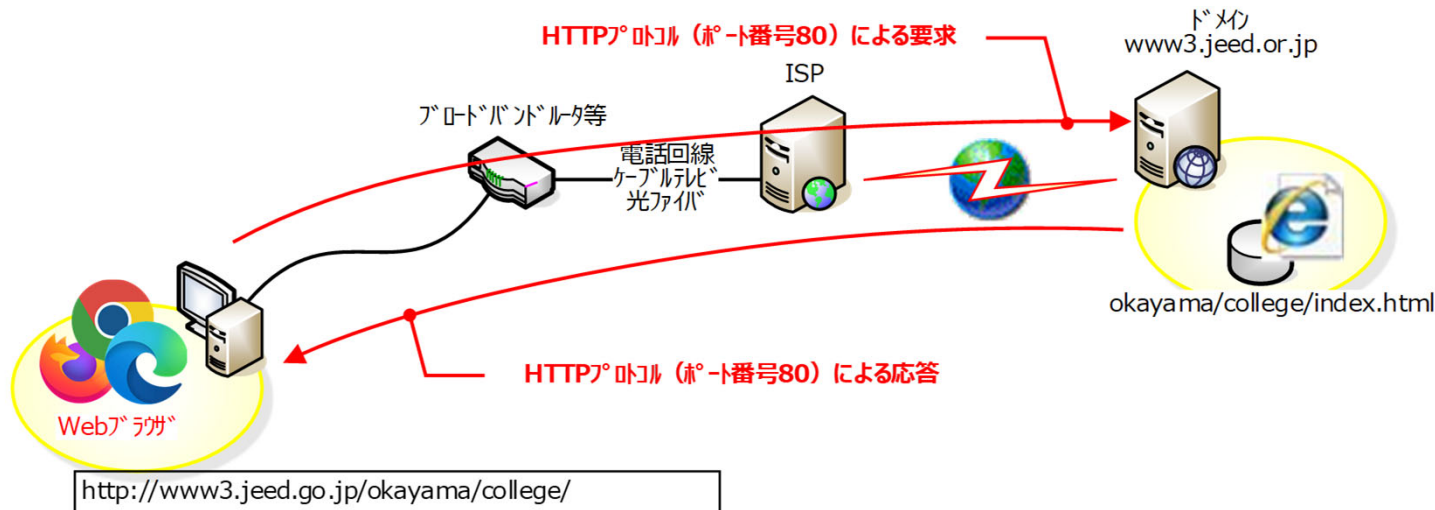







Diagram illustrating the structure of the URL `http://www3.jeed.or.jp/okayama/colledge/index.html`:





- 第3階層: 組織名 (3rd Layer: Organization Name) - `jeed`
- 第2階層: 組織属性 (2nd Layer: Organization Attribute) - `or`
- 第1階層: 国コード (1st Layer: Country Code) - `jp`
- プロトコル (Protocol) - `http`
- ドメイン (Domain) - `www3`
- ディレクトリ (Directory) - `okayama/colledge`
- ファイル名 (Filename) - `index.html`

Webサーバー




IIS(Internet Information Service)

-  WinNTServer上で稼働するアドオンソフトウェア
-  Windows2000 Server登場時にシステムの標準サービスとなり、名称もInternet Information Servicesに
-  HTTP/HTTPS、FTP、SMTP等の基本的なプロトコルをサポート
-  単なるWebサーバというよりも、ASP.NET、XML、SOAP に準拠したアプリケーションサーバという位置づけが強い
-  最新版は「10.0」で、Windows 10 や、Windows Server 2016 に付属





lighttpd

-  高速性が重視される環境に最適化され、安全高速柔軟であることを指向して設計されたWebサーバ
-  メモリの消費量が少なく、CPUへの負荷が少ない高速な動作が目的
-  lighttpdはフリーソフトウェア、オープンソースソフトウェア
-  LinuxやWindowsなどの環境で動作

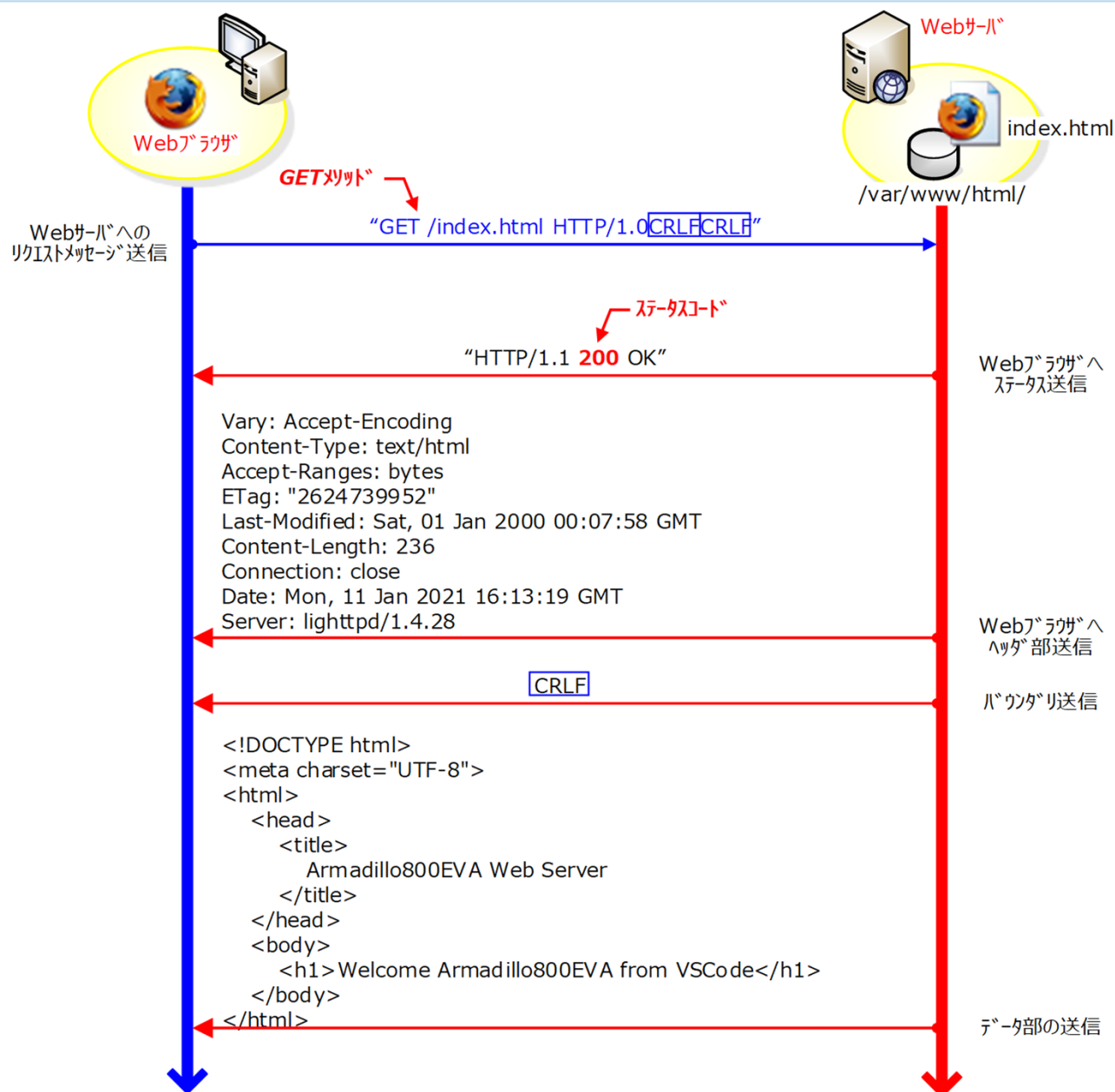
thttpd

-  オープンソースのWebサーバ
-  シンプルで使用メモリが少なく高速に動作するように設計
-  帯域幅調整機能があるため、ある特定のファイルタイプを持つファイルの最大転送速度を制限することができる

Apache HTTP Server

-  WebサーバといえばApacheと言われるぐらい世界中でもっとも使われているWebサーバ
-  大規模な商用サイトから自宅サーバまで幅広く利用されている
-  開発は、Apacheソフトウェア財団のApache HTTPサーバプロジェクトで行われている
-  Apacheライセンスの下でソースコードが公開および配布

HTTPプロトコルシーケンス



Webサーバーのレスポンスメッセージ

```

user@vbubuntu:~$ telnet 10.22.91.114 80
Trying 10.22.91.114...
Connected to 10.22.91.114.
Escape character is '^]'.
ステータス → GET /index.html HTTP/1.0
HTTP/1.0 200 OK ← ステータスコード
Vary: Accept-Encoding
ヘッダ部 { Content-Type: text/html
Accept-Ranges: bytes
ETag: "1639539525"
Last-Modified: Tue, 04 Jan 2022 09:27:36 GMT
Content-Length: 310
Connection: close
Date: Tue, 18 Jan 2022 12:06:13 GMT
Server: lighttpd/1.4.28
バウンダリ → CRLF
データ部 { <!DOCTYPE html>
<html lang="jp">
<head>
<meta charset="UTF-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-
scale=1.0">
<title>Armadillo800EVA Web Server</title>
</head>
<body>
<h1>Welcome Armadillo800EVA from VSCode</h1>
</body>
</html>Connection closed by foreign host.
  
```

HTTPプロトコルのメソッドとステータスコート

📌 メソッド

メソッド	1.0	1.1	説明
OPTIONS		○	通信オプションの通知など
GET	○	○	URIで指定した情報を要求。 URIがファイル名の場合はそのファイルの中身を、プログラム名の場合はそのプログラムの出力を返す
HEAD	○	○	GETと同じだがHTTPヘッダのみを返す
POST	○	○	クライアントからデータを（名前と値）のセットで渡す。フォームデータ送信で使用
PUT	△	○	URIで指定したサーバ上のファイルを置き換える
DELETE	△	○	URIで指定したサーバ上のファイルを削除する
TRACE		○	サーバが受け取った要求行とヘッダ行をそのまま送り返す

📌 実際に利用するのは、GET と POST

📌 ステータスコード 100番台

	意味	説明
100	Continue	クライアントは、そのリクエストを継続してもよい。
101	Switching Protocols	サーバは、この接続で使用されているアプリケーション・プロトコルの変更に対するクライアントのリクエストに応じようとしている。

📌 HTTPステータスコードの100～199番は通知コード

📌 これらのコードを実際に目にはめったにない

📌 Webブラウザがサイトにアクセスしてこれらのコードが返されたとしても通常それが画面に表示されることはなく内部コードとして
いるから

📌 最初のHTTP規格において100番台のコードが認められていなかったため

HTTPプロトコルのメソッドとステータスコード

📌 ステータスコード 200番台

	意味	説明
200	OK	リクエストは成功した。レスポンスと共に返される情報は、リクエストで使用されたメソッドに依存し、例えば、GET、HEAD、POST、TRACEのようになる。
201	Created	リクエストは果たされ、結果として新しいリソースが生成された。
202	Accepted	リクエストは処理用に受け入れられたが、処理は完了していない。CGIのように別の場所でデータを生成する時など。
203	Non-Authoritative Information	エンティティ・ヘッダーで返されたメタ情報は、オリジナル・サーバから入手できる確定的な集合ではなく、ローカルあるいはサード・パーティーのコピーから集められたものである。
204	No Content	サーバはリクエストを果たしたが、送り返すべき情報が存在しない。空のページにアクセスしようとした時など。
205	Reset Content	サーバはリクエストを果たしたが、ユーザ・エージェントは、リクエストを送信させたドキュメント・ビューをリセットするべきである。
206	Partial Content	サーバは、リソースに対する部分的なGETリクエストを果たした。

- 📌 200～299番のステータスコードは成功を示すコード
- 📌 これらのコードも、内部コードとしているため通常画面に表示されることはない
- 📌 通常は表示されないこれらのコードを読み出せる診断ツールもある。
- 📌 他のHTTPステータスコードと同様、200番台のコードも診断プロセスで貴重な情報を与えてくれることがある。

HTTPプロトコルのメソッドとステータスコード

📌 ステータスコード 300番台

	意味	説明
300	Multiple Choices	リクエストされたリソースは、それぞれが固有のロケーションをもつ表現セットの一つに対応し、ユーザ（あるいはユーザ・エージェント）が、より望ましい表現を選別してそのリクエストをそのロケーションにリダイレクトできるように、エージェント駆動型ネゴシエーション情報が提供されている。
301	Moved Permanently	リクエストされたリソースは新しい恒久的なURIを割り当てられたので、このリソースへの今後の参照は、返されたURIの一つを使用するべきである。
302	Moved Temporarily	リクエストされたリソースは、一時的に異なるURIに属する。
303	See Other	リクエストに対するレスポンスは異なるURIの下で発見でき、そのリソースをGETメソッドを使って検索することが望ましい。
304	Not Modified	リクエストされたリソースが指定された日付以降に更新されていない。クライアントが条件付きGETリクエストを実行し、アクセスは許可されたが、ドキュメントが更新されていない場合には、サーバはこのステータス・コードでレスポンスを行なうべきである。
305	Use Proxy	リクエストされたリソースは、ロケーション・フィールドによって与えられるプロキシを通じてアクセスされなければならない。

- 📌 300～399番のステータスコードは リダイレクションコード
- 📌 これらのコードは基本的にリクエストを実行するためには、ほかの処理を実行する必要があることをブラウザに通知
- 📌 これらの処理は、自動的に実行される処理もあれば、ユーザーによる追加入力を要求する処理もある。
- 📌 例えば、ステータスコード301は、特定のリソースが恒久的に移動し、以後そのリソースに対するすべての呼び出しは特定のURLを参照しなければならないことを示す

HTTPプロトコルのメソッドとステータスコード

ステータスコード 400番台

	意味	説明
400	Bad Request	リクエストは不正な構文であるために、サーバに理解されなかった。打ち込んだURLに変な間違いがあった時など。
401	Unauthorized	リクエストは、ユーザ認証を必要とする。認証に失敗した時など。
402	Payment Required	(予約)
403	Forbidden	サーバはリクエストを理解したが、そのリクエストの実行を拒否した。アクセス権限がない時など。
404	Not Found	サーバは、リクエストURIと一致するものを見つけられなかった。アドレスが無くなった時など。
405	Method Not Allowed	リクエスト・ラインで指定されたメソッドは許されていない。
406	Not Acceptable	リクエストによって識別されるリソースは、リクエスト中に送信された受理ヘッダに従って、受理できない内容特性をもつレスポンス実体を生成することができるのみである。受信の条件が適合していないため、リクエストされるリソースは受理できない。
407	Proxy Authentication Required	クライアントは最初にプロキシに対する認証を行わなければならないことを示している。
408	Request Time-out	クライアントは、サーバの待機時間内にリクエストを発行しなかった。
409	Conflict	リクエストは、リソースの現在の状態と衝突するために完了できなかった。
410	Gone	リクエストされたリソースは、もはやそのサーバでは入手できなくなっており、転送先のアドレスも不明である。
411	Length Required	サーバは、定義された Content-Lengthのないリクエストの受理を拒否した。
412	Precondition Failed	一つ以上のリクエスト・ヘッダ・フィールドで与えられた前提条件が、サーバでテストされたときに偽（不正）であると評価された。
413	Request Entity Too Large	リクエスト実体がサーバの想定よりも、あるいは処理可能なものよりも大きい場合、想定処理を拒否している。
414	Request-URI Too Large	リクエストURIがサーバの想定よりも、あるいは処理可能なものよりも大きい場合、サーバはサービスを拒否している。
415	Unsupported Media Type	リクエスト実体は、リクエストされたメソッドに対してリクエストされたリソースがサポートしていないフォーマットであるため、サーバがリクエストのサービスを拒否している。

- 400番台は、クライアント側のエラーを示すコード
- これらのエラーコードは、セキュリティに関連している場合が多く、例えば、クライアントがアクセスを許可されていないリソースへのアクセスを試みると、サーバはステータスコード401を返す
- 同様に、クライアントが未許可のリソースへのアクセスを試み、クライアントの認証ステータスによっても状況が変化しない場合には、サーバはステータスコード403を返すことがある。これは、そのリソースへのアクセスが禁じられていることを示す。
- 400番台のコードは、リクエストが不正な形式であったり、クライアント側でタイムアウトが起きたりすると返されることもある
- 404エラーは、リクエストされたリソースが見つからなかったことを示すが、このエラーがクライアント側で起きた場合、ネットワーク接続の問題が原因であることが多い。
- それ以外にも、リソースがサーバから削除されるか、またはリソースの名前が変更されたためにこの404エラーが発生する場合もある

HTTPプロトコルのメソッドとステータスコード

ステータスコード 500番台

	意味	説明
500	Internal Server Error	サーバは、リクエストの実行を妨げる予期しない状況に遭遇した。 CGIスクリプト・エラーなど。
501	Not Implemented	サーバは、リクエストを実行するために必要な機能をサポートしていない。
502	Bad Gateway	ゲートウェイあるいはプロキシとして動作しているサーバが、リクエストを実行しようとしてアクセスした上位サーバから不正なレスポンスを受信した。 不正なゲートウェイ経由のアクセスなど。
503	Service Unavailable	サーバはサーバの一時的な過負荷あるいはメンテナンスのために、現在、リクエストを扱うことができない
504	Gateway Time-out	ゲートウェイあるいはプロキシとして動作しているサーバが、リクエストを完了しようとしてアクセスした上位サーバからタイムリーなレスポンスを受信できなかった。
505	HTTP Version not supported	サーバは、リクエスト・メッセージで使用されたHTTPプロトコル・バージョンをサポートしていない、あるいはサポートを拒否している。

- 500番台のステータスコードは、サーバのエラー
- 例えば、Webサーバでタイムアウトが発生すると504エラーが返されます
- ただし500番台のエラーは、サーバの問題ではなく、サーバ上で動作しているWebアプリケーションの問題を示していることもある

HTML

HTML (HyperText Markup Language)

📌仕様

- 📌 W3C (World Wide Web Consortium) が定めている

- 📌 <https://www.w3.org/>

- 📌 日本語訳

- 📌 <https://momdo.github.io/html/>

📌HyperText

- 📌 文書中の指定箇所にリンクを貼ることができて他のページに移動することができる機能
- 📌 `cgiled0a.cgiへ`

📌Markup

- 📌 しるしをつける
- 📌 HTMLタグを使うこと

📌HTMLタグ

- 📌 開始タグ `<` と `>` で囲む
- 📌 終了タグ `</` と `>` で囲む
- 📌 開始タグと終了タグは基本セットで記述する
- 📌 終了タグが無いタグもある
 - 📌 ``

HTMLのバージョン

🔗 HTML4.01

- 🔗 1999年12月24日に勧告されたバージョン。
- 🔗 視覚的な表現を行なう為の要素や属性を非推奨（CSSという言語に移行）
- 🔗 文章構造のみを記すというHTML本来の目的に沿ったバージョン

🔗 XHTML1.0

- 🔗 2000年1月26日に勧告されたバージョン。
- 🔗 XHTMLの X は eXtensible : 拡張 という意味
- 🔗 具体的にはXMLという別の言語の仕様に従って定義しなおしたHTML
- 🔗 HTMLとしては大きな変更はない
- 🔗 タグは小文字で記述するなどといったルールがより厳格になったバージョン

🔗 HTML5

- 🔗 2014年10月28日に勧告された現時点での最新バージョン。
- 🔗 Webアプリケーションを開発するための様々な仕様が新たに盛り込まれている
- 🔗 DOCTYPE宣言不要
- 🔗 2021年1月廃止

HTMLの最新規格（2022/1/18調べ）

📌 W3C と WHATWG の対立

- 📌 WHATWG : Web Hypertext Application Technology Working Group
- 📌 Apple, Mozilla, Opera の 3社が 2004年 に設立



<https://future-architect.github.io/articles/20210621a/> より引用

📌 HTML Living Standard

- 📌 2019年1月29日、WHATWG のサイトで、「W3Cが正式にRecommendation（勧告）する」と発表
- 📌 W3C の規格は全て廃止
- 📌 <https://html.spec.whatwg.org/> に最新規格あり
- 📌 <https://momdo.github.io/html/> に日本語翻訳あるが古い

📌 HTML5 と HTML Living Standard との違い

- 📌 をクリックしたときに、hrefのページに遷移しつつ、別URLにpingを送信するping属性の追加10
- 📌 の遅延読み込みを行う loading の追加11
- 📌 autofocus が <button>や<input>以外の、全てのフォーカス可能な要素にも適用可能に12
- 📌 <h1>は複数使用可能13（ただし <section>も併用すること）

HTMLの編集

📌 Visual Code Editor なら！を書いて Enter すると以下を自動で記述する

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>

</body>
</html>
```

📌 <!DOCTYPE html>

- 📌 DOCTYPE宣言（HTML5はこれだけ）
- 📌 大文字と小文字は区別しない
- 📌 DOCTYPE宣言があれば標準モード、なければ互換モード

📌 <html lang="en">

- 📌 html の開始タグ
- 📌 lang は属性、enは英語圏なので、たいていjpへ変更

📌 <head>

- 📌 ヘッダ部、表示しない要素

📌 <meta charset="UTF-8">

- 📌 metaタグ：文書情報のこと
- 📌 charset属性はWebブラウザへの文字コードの指定
- 📌 htmlファイルの文字コードと一緒にすること

📌 <meta name="viewport" content="width=device-width, initial-scale=1.0">

- 📌 どの端末でも画面幅に合わせたレスポンスなWebデザインをするための仕掛け

📌 <title>

- 📌 タイトル
- 📌 たいていWebブラウザのタブに表示される

📌 <body>

- 📌 Webページ本体
- 📌 Webブラウザで挙動が異なる

よく利用するタグ

📌 デザイン関係のタグ

- 📌 <h1> 見出しレベル1 heading
- 📌 <p> 段落 paragraph
- 📌 強調 bold
- 📌
 改行 break
- 📌 <a href> リンク anchor hypertext reference
- 📌 画像表示 image
- 📌 箇条書き unordered list
- 📌 番号付き箇条書き Ordered List
- 📌 箇条書きの中身 list item
- 📌 <table> 表
- 📌 <th> 表見出し table header
- 📌 <tr> 表の行 rable row
- 📌 <td> 表のセル table data

📌 とほほのWWW入門 逆引きリファレンス

- 📌 <http://www.tohoho-web.com/how2/index.htm>

よく利用するタグ

📌 制御関係のタグ

📌 <form> 入力・送信フォーム

📌 <input> 入力 属性でいろいろカスタマイズ

📌 type="text "

📌 type="password "

📌 type="checkbox "

📌 type="radio "

📌 type="file "

📌 type="hidden "

📌 type="submit "

📌 type="reset "

📌 type="button "

📌 type="image"

📌 <select> セレクトボックス

📌 <button> ボタン 属性で指定

📌 type="submit "

📌 type="reset "

📌 type="button "

📌 <input type=submit>は止めて<button type=submit>を使うべき

📌 <https://qiita.com/irxground/items/c8537d30e9760c5b3e5c>

参考サイト

📌 HTML5 クイックリファレンス

📌 <http://www.htmq.com/>

📌 とほほのWWW入門

📌 <http://www.tohoho-web.com/www.htm>

📌 Webデザインの基礎知識1

📌 <https://www.urban-project.jp/blog/recommended/975/>

📌 WHATWG

📌 <https://html.spec.whatwg.org/>