COMP7306: Web technologies

The World Wide Web

23 January 2013

Pierre Senellart

Outline

The Internet

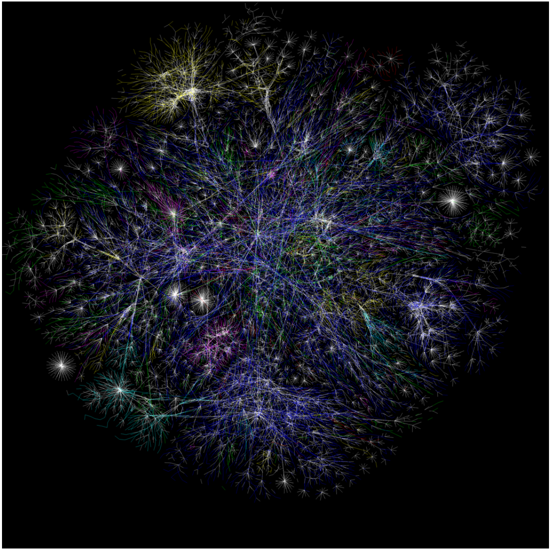
The World Wide Web

HTML

HTTP

Conclusion

A network of networks: interconnected computers



http://www.opte.org/

The Internet protocol stack

A stack of communication protocols, on top of each other.

Application	HTTP, FTP, SMTP, DNS	
Transport	TCP, UDP, ICMP	(sessions, reliability. . .)
Network	IP (v4, v6)	(routing, addressing)
Link	Ethernet, 802.11 (ARP)	(addressing local machines)
Physical	Ethernet, 802.11 (physical)	

IP (Internet Protocol) [IETF, 1981a]

- Addressing machines and routing over the Internet
- Two versions of the IP protocol on the Internet: IPv4 (very well spread) and IPv6 (not that well-spread yet)
- IPv4: 4-byte addresses assigned to each computer, e.g., 137.194.2.24. Institutions are given ranges of such addresses, to assign as they will.
- Problem: only 2^{32} possible addresses (actually, a large number of them cannot be assigned to new hosts, for multiple reasons). This means many hosts connected to the Internet do not have an IPv4 address and some network address translation (NAT) occurs.
- IPv6: 16-byte addresses; much larger address space! Addresses look like 2001:660:330f:2::18 (meaning 2001:0660:0330f:0002:0000:0000:0000:0018). Other nice features (multicast, autoconfiguration, etc.).

TCP (Transmission Control Protocol) [IETF, 1981b]

- One of the two main transport protocols used on IP, with UDP (User Datagram Protocol)
- Contrarily to UDP, provides reliable transmission of data (acknowledgments)
- Data is divided into small datagrams that are sent over the network, and possibly reordered at the end point
- Like UDP, each TCP transmission indicates a source and a destination port number (between 0 and 65535) to distinguish it from other traffic
- A client usually select a random port number for establishing a connection to a fixed port number on a server
- The port number on a server conventionally identifies an application protocol on top of TCP/IP: 22 for SSH, 25 for SMTP, 110 for POP3. . .

DNS (Domain Name System) [IETF, 1999a]

- IPv4 addresses are hard to memorize, and a given service (e.g., a Web site) may change IP addresses (e.g., new Internet service provider)
- Even more so for IPv6 addresses!
- DNS: a UDP/IP-based protocol for associating human-friendly names (e.g., `www.google.com`, `weather.yahoo.com`) to IP addresses
- Hierarchical domain names: `com` is a top-level domain (TLD), `yahoo.com` is a subdomain thereof, etc.
- Hierarchical domain name resolution: root servers with fixed IPs know who is in charge of TLDs, servers in charge of a domain know who is in charge of a subdomain, etc.
- Nothing magic with `www.google.com`: just a subdomain of `google.com`.

Outline

The Internet

The World Wide Web

- Introduction
- The Web: a market

HTML

HTTP

Conclusion

Outline

The Internet

The World Wide Web

- Introduction
- The Web: a market

HTML

HTTP

Conclusion

Internet and the Web

Internet: physical network of computers (or hosts)

World Wide Web, Web, WWW: logical collection of hyperlinked documents

- static and dynamic
- public Web and private Webs
- each document (or Web page, or resource) identified by a URL

An abridged timeline of Web history

1969	ARPANET (the ancestor of the Internet)
1974	TCP (Vinton G. Cerf & Robert E. Kahn, Turing award winners 2004)
1990	World Wide Web, HTTP, HTML (Tim Berners-Lee, Robert Cailliau)
1993	Mosaic (the first public successful graphical browser, ancestor of Netscape)
1994	Yahoo! (David Filo, Jerry Yang)
1994	Foundation of the W3C
1995	Amazon.com, Ebay
1995	Internet Explorer
1995	AltaVista (Louis Monier, Michael Burrows)
1998	Google (Larry Page, Sergey Brin)
2001	Wikipedia (Jimmy Wales)
2004	Mozilla Firefox
2005	YouTube
2008	Google Chrome

Sources: [Electronic Software Publishing Corporation, 2008], [BBC, 2006]

URL (Uniform Resource Locator) [IETF, 1994]

$$\underbrace{\texttt{https}}_{\text{scheme}}\texttt{://}\underbrace{\texttt{www.example.com}}_{\text{hostname}}\underbrace{\texttt{:443}}_{\text{port}}\texttt{/}\underbrace{\texttt{path/to/doc}}_{\text{path}}\underbrace{\texttt{?name=foo\&town=bar}}_{\text{query string}}\underbrace{\texttt{\#para}}_{\text{fragment}}$$

scheme: way the resource can be accessed; generally http or https

hostname: domain name of a host (cf. DNS); hostname of a website may start with `www.`, but not a rule.

port: TCP port; defaults: 80 for `http` and 443 for `https`

path: logical path of the document

query string: optional additional parameters (dynamic documents)

fragment: optional subpart of the document

Relative URLs with respect to a context (e.g., the URL above):

`/titi`	`https://www.example.com/titi`
`tata`	`https://www.example.com/path/to/tata`

The Web: a mixture of technologies

- For content: HTML/XHTML, but also PDF, Word documents, text files, XML (RSS, SVG, MathML, etc.)...
- For presenting this content: CSS, XSLT
- For animating this content: JavaScript, AJAX, VBScript...
- For interaction-rich content: Flash, Java, Sliverlight, ActiveX, `<canvas>` API...
- Multimedia content: images, sounds, videos...
- And on the server side: any programming language and database technology to serve this content, e.g., PHP, JSP, Java servlets, ASP, ColdFusion, etc.

Quite complex to manage! Being a Web developer nowadays requires mastering a lot of different technologies; designing a Web client requires being able to handle a lot of different technologies!

Outline

The Internet

The World Wide Web

- Introduction
- The Web: a market

HTML

HTTP

Conclusion

Web clients

- Graphical browsers (cf. next slide)
- Text browsers: w3m, lynx, links (free software, Windows, Mac OS, Linux, Unix); rarely used nowadays
- Other browsers: audio browsers, etc.
- But also: spiders for siphoning a Web site, search engine crawlers, machine translation software. . .

A very large variety of clients! Web standards (mainly, HTML, CSS, HTTP) are supposed to describe what their interpretation of a Web page should be. In reality, more complex (tag soup).

Graphical browsers

Browser	Engine	Share	Distribution
Chrome+Android	WebKit	35%	Windows, MacOS, Linux FS
Internet Explorer	Trident	26%	with Windows
Firefox	Gecko	19%	Windows, MacOS, Unix FS
Safari, inc. iOS	WebKit	10%	MacOS, Windows FC
Opera	Presto	4%	Windows, MacOS, Unix, mobiles FC

FC: free of charge (free as a beer)
FS: free software (free as a man)

Market shares: various sources, precise numbers hard to obtain. IE continually decreasing over the last years.

Trident remains the worst standard-compliant rendering engine.

News about graphical browsers

- Google Chrome has known impressive success (only 4 years since its initial release)
- Versions of Internet Explorer 6 to 9 still all commonly used (especially in the enterprise world); IE6 is the browser coming with initial releases of Windows XP browser.
- Versions of Internet Explorer tied with versions of Windows (IE10 recently released with Windows 8).
- Other browsers tend to have recent versions installed, but not always (esp., mobile browsers).

Web servers

Server	Share	Distribution
Apache	60%	Windows, Mac OS, Linux, Unix FS
Microsoft IIS	15%	with some versions of Windows
nginx	12%	Windows, Mac OS, Linux, Unix FS
lighthttpd	1%	Windows, Mac OS, Linux, Unix FS

- Market share: according to various studies, precise numbers do not really mean anything.
- Many large software companies have either their own Web server or their own modified version of Apache (notably, GFE/GWS for Google).
- nginx and lighthttpd are lighter (i.e., less feature-rich, but faster in some contexts) than Apache.
- The versions of Microsoft IIS released with consumer versions of Windows are very limited.

Web search engines

- A large number of different search engines, with market shares varying a lot from country to country.
- At the world level:

Google	vastly dominating (around 80% of the market; more than 90% market share in Western Europe!)
Yahoo!+Bing	still resists to its main competitor (perhaps 10% of the market)

- In some countries, local search engines dominate the market (Baidu with 75% in China, Naver in Korea, Yahoo! Japan in Japan)

Recent changes

In July 2009, Microsoft and Yahoo! announced a major agreement:

- Yahoo! stops developing its own search engine (launched in 2003, after the buyouts of Inktomi and Altavista) and will use Bing instead;
- Yahoo! will provide the advertisement services used in Bing.

Operational, but does not concern Yahoo! Japan, which on the contrary uses Google as engine.

Outline

The Internet

The World Wide Web

HTML

HTTP

Conclusion

HTML (HyperText Markup Language) [W3C, 1999]

- normalized by the W3C (World Wide Web Consortium) formed of industrials (Microsoft, Google, Apple. . .) and academic institutions (ERCIM, MIT, etc.)
- open format: possible processing by a wide variety of software and hardware
- text files with tags
- describes the structure and content of a document, focus on accessibility
- (theoretically) no presentation information (this is the role of CSS)
- no description of dynamic behaviors (this is the role of server-side languages, JavaScript, etc.)

The HTML language

- HTML is a language alternating text and tags (`<blabla>` or `</blabla>`)
  - Tags allow structuring each part of a document, and are used for instance by a browser to lay out the document.
- HTML files
  - are structured in two main parts: the header `<head> ... </head>`) and the body `<body> ... </body>`)
- In HTML, blanks (spaces, tabs, carriage returns) are generally equivalent and only serve to delimit words, tags, etc. The number of blanks does not matter.

Tags

- Syntax: (opening and closing tag)

```
<tag attributes>content</tag>
```

or (element with no content)

```
<tag attributes>
```

tag keyword referring to some particular HTML element

content may contain text and other tags

attributes represent the various parameters associated with the element, as a list of `name="value"` or `name='value'`, separated by spaces (quotes are not always mandatory, but they become mandatory if `value` has “exotic” characters)

Tags

- Names of elements and attributes are usually written in lowercase, but `<head>` and `<HeAd>` are equivalent.
- Tags are opened and closed in the right order (`<b><i></i></b>` and not `<b><i></b></i>`).
- Strict rules specify which tags can be used inside which.
- Under some conditions, a tag can be implicitly closed, but these conditions are complex to describe.
- `<!--foobar-->` denotes a comment, which is not to be interpreted by a Web client.

Structure of a document

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"
  "http://www.w3.org/TR/html4/strict.dtd">
<html lang="en">
  <head>
    <!-- Header of the document -->
  </head>
  <body>
    <!-- Body of the document -->
  </body>
</html>
```

- The doctype declaration `<!DOCTYPE ...>` specify which HTML version is used.
- The language of the document is specified with the `lang` attribute of the main `<html>` tag.

Header

- The header of a document is delimited by the tags `<head> ... </head>`.
- The header contains meta-informations about the document, such as its title, encoding, associated files, etc. The two most important items are:
  - The character set of the page, usually at the very beginning of the header

    ```
    <meta http-equiv="Content-Type"
          content="text/html; charset=utf-8">
    ```

  - The title of the page (the only required item inside the header). This is the information displayed in the title bar of Web browsers.

    ```
    <title>My great website</title>
    ```

Character sets

Unicode: character repertoire, assigning to each character, whatever its script or language, an integer number.

Examples

A	→	65	ε	→	949
é	→	233	$\aleph$	→	1488

Character set: concrete method for representing a Unicode character.

Examples (é)

iso-8859-1	11101001		only for some characters
utf-8	11000011	10101001	
utf-16	11101001	00000000	

utf-8 has the advantage of being able to represent all Unicode characters, in a way compatible with the legacy ASCII encoding.

The body of a HTML document

- `<body> ... </body>` tags delimit the body of a document.
- The body is structured into sections, paragraphs, lists, etc.
- 6 tags describe sections, by decreasing order of importance:
  - `<h1>Title of the page</h1>`
  - `<h2>Title of a main section</h2>`
  - `<h3>Title of a subsection</h3>`
  - `<h3>Title of a subsubsection</h3>`
  - ...
- `<p> ... </p>` tags delimit paragraphs of text. All text paragraphs should be delimited thusly.
- Directly inside `<body> ... </body>` can only appear block elements: `<p>`, `<h1>`, `<form>`, `<hr>`, `<ul>`, `<table>`...in addition to the `<div>` tag which denotes a block without precise semantics.

Images

- To add an image into an HTML document, one uses the `<img>` tag.
  - The `src` attribute specifies the location of the image (URL).
  - The `alt` attribute is a textual alternative when the image is unavailable. Compulsory, so that every user agent (screen readers, text browsers, technical issues, robots) that cannot see the image has a replacement text.

```
<img src="http://www.lri.fr/images/lri.png" alt="LRI">
<img src="../images/eiffel.png" alt="Eiffel Tower">
```

- Image formats usable on the Web are:
  - JPEG (.jpg), for photos and other continuous tone pictures.
  - GIF (.gif) and PNG (.png) for other kind of pictures; PNG is to be preferred (transparency, color depth...) except for animated images (to use sparsely!).

Links

- What differetiates Web pages (hypertext pages) from normal documents: links!
- Introduced with `<a> ... </a>`
- Navigating a link can bring to:
  - a resource on another server or another file of the same server
  - another part of the same document

Links

Links are made using the `href` attribute of the `<a>` tag, whose content will be the link:

```
<a href="http://www.cnrs.fr/">
  <img src="images/cnrs.gif" alt="CNRS">
</a>

<a href="bio/indexbioinfo.html">Bioinformatics</a>
```

Anchors

- Anchors serve to reach a precise point in the document.
  - They are defined, either on an existing tag by using the `id` attribute, or with an `<a id="...">` :

```
<h3 id="tutorials">Tutorials</h3>
<a id="tutorials">
```

  - Then, one can link to this anchor:

```
<a href="#tutorials">tutorials</a>
<a href="http://www.w3.org/#tutorials">tutorials</a>
```

  - Commonly, the old `<a name="...">` syntax is used.

The different versions of HTML

- HTML 4.01 (1999) strict (as described earlier) and transitional

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
```

- XHTML 1.0 (2000) strict and transitional
- XHTML 1.1 and XHTML 2.0: mostly a failure, unusable and unused in today's Web
- HTML5: upcoming standard (W3C candidate recommendation, 2012), partly implemented, continuously updated

```
<!DOCTYPE html>
```

Tag soup

- A lot of HTML documents on the Web date back from before HTML 4.01
- In practice: many Web pages do not respect any standards at all (with or without doctype declarations) $\Longrightarrow$ browsers do not respect these standards $\Longrightarrow$ tag soup!
- When dealing with pages from the real Web, necessary to use all sorts of heuristics to interpret a Web page.

HTML vs XHTML

- XHTML: an XML format
- Tags without content `<img>`, are written `<img />` in XHTML.
- Some elements can be left unclosed in HTML (`<ol><li> one <li>two </ol>`), but closing is mandatory in XHTML.
- Attribute values can be written without quotes (`<img src=toto.png alt=toto>`) in HTML, quotes are required in XHTML.
- Element and attribute names are not case-sensitive in HTML (`<HTMl laNg=fr>`), but are in XHTML (everything must be in lowercase).
- Attributes `xmlns` and `xml:lang` on the `<html>` tag in XHTML.
- And some other small subtleties…

HTML's future: XHTML 2.0 vs HTML 5

- XHTML 2.0: initiative of the W3C, incompatible with HTML 4.01/XHTML 1.0, major changes
- HTML 5: initiative of browser developers, compatible with HTML 4.01/XHTML 1.0, incremental but numerous changes
- XHTML 2.0 abandoned in July 2009
- HTML 5 features have appeared in recent browsers (Internet Explorer 9 included)
- HTML 5 offers the choice between syntactic conventions inherited from both HTML 4.01 and XHTML
- New features: 2D drawing (`<canvas>`), multimedia (`<audio>` , `<video>`), better structuring elements (`<section>` , `footer`), etc.

Outline

The Internet

The World Wide Web

HTML

HTTP

Conclusion

HTTP (HyperText Transfer Protocol) [IETF, 1999b]

- Application protocol at the basis of the World Wide Web
- Latest and most widely used version: HTTP/1.1

- Client request:

```
GET /MarkUp/ HTTP/1.1
Host: www.w3.org
```

- Server response:

```
HTTP/1.1 200 OK
...
Content-Type: text/html; charset=utf-8

<!DOCTYPE html ...> ...
```

- Two main HTTP methods: GET and POST (HEAD is also used in place of GET, to retrieve meta-information only).
- Additional headers, in the request and the response
- Possible to send parameters in the request (key/value pairs).

GET

- Simplest type of request.
- Possible parameter are sent at the end of a URL, after a '?'
- Not applicable when there are too many parameters, or when their values are too long.
- Method used when a URL is directly accessed in a browser, when a link is followed, and for some forms.

Example (Google query)

URL: `http://www.google.com/search?q=hello`

Corresponding HTTP GET request:

```
GET /search?q=hello HTTP/1.1
Host: www.google.com
```

POST

- Method only used for submitting forms.

Example

```
POST /php/test.php HTTP/1.1
Host: www.w3.org
Content-Type: application/x-www-form-urlencoded
Content-Length: 100

type=search&title=The+Dictator&format=long&country=US
```

Parameter encoding

- By default, parameters are sent (with GET or POST) in the form: `name1=value1&name2=value2`, and special characters (accented characters, spaces...) are replaced by codes such as `+`, `%20`. This way of sending parameters is called `application/x-www-form-urlencoded`.
- For the POST method, another heavier encoding can be used (several lines per parameter), similar to the way emails are built: mostly useful for sending large quantity of information. Encoding named `multipart/form-data`.

Status codes

- The HTTP response always starts with a status code with three digits, followed by a human-readable message (e.g., 200 OK).
- The first digit indicates the class of the response:

1 Information
2 Success
3 Redirection
4 Client-side error
5 Server-side error

Most common status codes

200	OK
301	Permanent redirection
302	Temporary redirection
304	No modification
400	Invalid request
401	Unauthorized
403	Forbidden
404	Not found
500	Server error

Virtual hosts

- Different domain names can refer to the same IP address, i.e., the same physical machine (e.g., `www.google.fr` and `www.google.com`)
- When a machine is contacted by TCP/IP, it is through its IP address
- No *a priori* way to know which precise domain name to contact
- In order to serve different content according to the domain name (virtual host): header `Host:` in the request (only header really required)

Example

```
GET /search?hl=fr&q=hello HTTP/1.1
Host: www.google.fr
```

Content type

- The browser behaves differently depending on the content type returned: display a Web page with the layout engine, display an image, load an external application, etc.
- MIME classification of content types (e.g., `image/jpeg`, `text/plain`, `text/html`, `application/xhtml+xml`, `application/pdf` etc.)
- For a HTML page, or for text, the browser must also know what character set is used (this has precedence over the information contained in the document itself)
- Also returned: the content length (can be used to display a progress bar)

Example

```
HTTP/1.1 200 OK
Content-Type: text/html; charset=UTF-8
Content-Length: 3046
```

Client and server identification

- Web clients and servers can identify themselves with a character string
- Useful to serve different content to different browsers, detect robots…
- … but any client can say it's any other client!
- Historical confusion on naming: all common browsers identify themselves as Mozilla!

Example

```
User-Agent: Mozilla/5.0 (X11; U; Linux x86_64; fr;
rv:1.9.0.3) Gecko/2008092510 Ubuntu/8.04 (hardy)
Firefox/3.0.3

Server: Apache/2.0.59 (Unix) mod_ssl/2.0.59 OpenSSL/0.9.8e
PHP/5.2.3
```

Authentication

- HTTP allows for protecting access to a Web site by an identifier and a password
- Attention: (most of the time) the password goes through the network uncrypted (but for instance, just encoded in Base64, revertible encoding)
- HTTPS (variant of HTTP that includes encryption, cryptographic authentication, session tracking, etc.) can be used instead to transmit sensitive data

Example

```
GET ... HTTP/1.1
Authorization: Basic dG9Obzp0aXRp
```

Content negotiation

- A Web client can specify to the Web server:
  - the content type it can process (text, images, multimedia content), with preferrence indicators
  - the languages preferred by the user
- The Web server can thus propose different file formats, in different languages.
- In practice, content negociation on the language works, and is used, but content negociation on file types does not work because of bad default configuration of some browsers.

Example

```
Accept: text/html,application/xhtml+xml,application/xml;
q=0.9,*/*;q=0.8
Accept-Language: fr,fr-fr;q=0.8,en-us;q=0.5,en;q=0.3
```

Cookies [IETF, 2000]

- Information, as key/value pairs, that a Web server asks a Web client to keep and retransmit with each HTTP request (for a given domain name).
- Can be used to keep information on a user as she is visiting a Web site, between visits, etc.: electronic cart, identifier, and so on.
- Practically speaking, most often only stores a session identifier, connected, on the server side, to all session information (connected or not, user name, data. . .)
- Simulates the notion of session, absent from HTTP itself

Example

```
Set-Cookie: session-token=RJYBsG//azkfZrRazQ3SPQhlo1FpkQka2;
path=/; domain=.amazon.de;
expires=Fri Oct 17 09:35:04 2008 GMT

Cookie: session-token=RJYBsG//azkfZrRazQ3SPQhlo1FpkQka2
```

Conditional downloading

- A client can ask for downloading a page only if it has been modified since some given date.
- Most often not applicable, the server giving rarely a reliable last modification date (difficult to obtain for dynamically generated content!).

Example

```
If-Modified-Since: Wed, 15 Oct 2008 19:40:06 GMT
```

```
304 Not Modified
Last-Modified: Wed, 15 Oct 2008 19:20:00 GMT
```

Originating URL

- When a Web browser follows a link or submits a form, it transmits the originating URL to the destination Web server.
- Even if it is not on the same server!

Example

```
Referer: http://www.google.fr/
```

Outline

The Internet

The World Wide Web

HTML

HTTP

Conclusion

Conclusion

What you should remember

- The Web is not the same thing as the Internet!
- Variety of protocols, languages, technologies used on the Web.
- HTML as a markup language for content and structure.
- HTTP as a communication protocol for downloading resources from the Web.

References

Software

- Variety of user agents to get different views of a Web site: text browsers, graphical browsers, screen reader (`http://webanywhere.cs.washington.edu/`)...
- Browser developer options or extensions such as Firebug to inspect HTML page source and HTTP communication
- HTML validation service: `http://validator.w3.org/`
- HTML Parser, TagSoup: Java libraries for parsing real-world Web pages

To go further

- M. Zalewski, The Tangled Web, No Starch Press, November 2011
- Main references:
  - HTML 4.01 recommendation [W3C, 1999]
  - HTTP/1.1 RFC [IETF, 1999b]

Bibliography I

BBC. Fifteen years of the web. http://news.bbc.co.uk/2/hi/technology/5243862.stm, 2006. Accessed March 2009.

Electronic Software Publishing Corporation. Internet & World Wide Web history. http://www.elsop.com/wrc/h_web.htm, 2008. Accessed March 2009.

IETF. Request For Comments 791. Internet Protocol. http://www.ietf.org/rfc/rfc0791.txt, September 1981a.

IETF. Request For Comments 793. Transmission Control Protocol. http://www.ietf.org/rfc/rfc0793.txt, September 1981b.

IETF. Request For Comments 1738. Uniform Resource Locators (URLs). http://www.ietf.org/rfc/rfc1738.txt, December 1994.

IETF. Request For Comments 1034. Domain names—concepts and facilities. http://www.ietf.org/rfc/rfc1034.txt, June 1999a.

Bibliography II

IETF. Request For Comments 2616. Hypertext transfer protocol—HTTP/1.1. `http://www.ietf.org/rfc/rfc2616.txt`, June 1999b.

IETF. Request For Comments 2965. HTTP state management mechanism. `http://www.ietf.org/rfc/rfc2965.txt`, October 2000.

W3C. HTML 4.01 specification, September 1999. `http://www.w3.org/TR/REC-html40/`.

Licence de droits d'usage

Contexte public } avec modifications

Par le téléchargement ou la consultation de ce document, l'utilisateur accepte la licence d'utilisation qui y est attachée, telle que détaillée dans les dispositions suivantes, et s'engage à la respecter intégralement.

La licence confère à l'utilisateur un droit d'usage sur le document consulté ou téléchargé, totalement ou en partie, dans les conditions définies ci-après et à l'exclusion expresse de toute utilisation commerciale.
Le droit d'usage défini par la licence autorise un usage à destination de tout public qui comprend :
– le droit de reproduire tout ou partie du document sur support informatique ou papier,
– le droit de diffuser tout ou partie du document au public sur support papier ou informatique, y compris par la mise à la disposition du public sur un réseau numérique,
– le droit de modifier la forme ou la présentation du document,
– le droit d'intégrer tout ou partie du document dans un document composite et de le diffuser dans ce nouveau document, à condition que :
  – L'auteur soit informé.

Les mentions relatives à la source du document et/ou à son auteur doivent être conservées dans leur intégralité.

Le droit d'usage défini par la licence est personnel et non exclusif.
Tout autre usage que ceux prévus par la licence est soumis à autorisation préalable et expresse de l'auteur : sitepedago@telecom-paristech.fr