Introduction Web Technologies

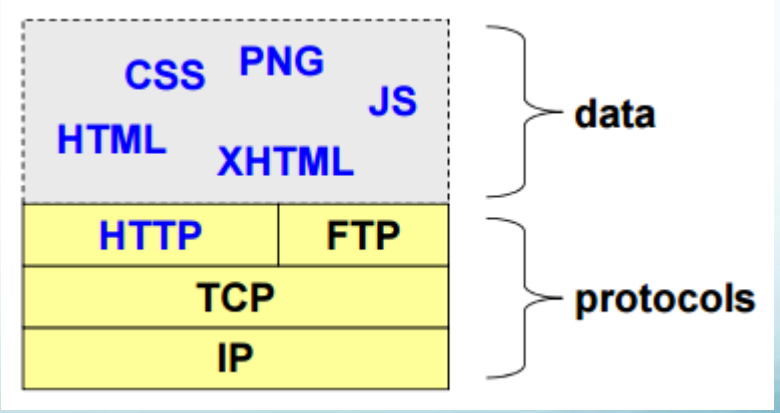
(HTTP, HTML, Dynamic Web)

www.dominusoft.co

Table of content

World Wide Web (www)

- The World Wide Web (www, W3) is an information system of interlinked hypertext documents and other digital resources that are accessed via the Internet.
- Set of
  - *communication protocols*
  - *data formats*
- Build on top of TCP/IP channels



Protocols for the web

- Several existing protocols can be used
  - HTTP
  - FTP
- The application protocol determines which functions are available (e.g. with FTP only GET and PUT of files)
- HTTP is an application layer protocol
- HTTP functions as a request-response protocol in the client-server computing model

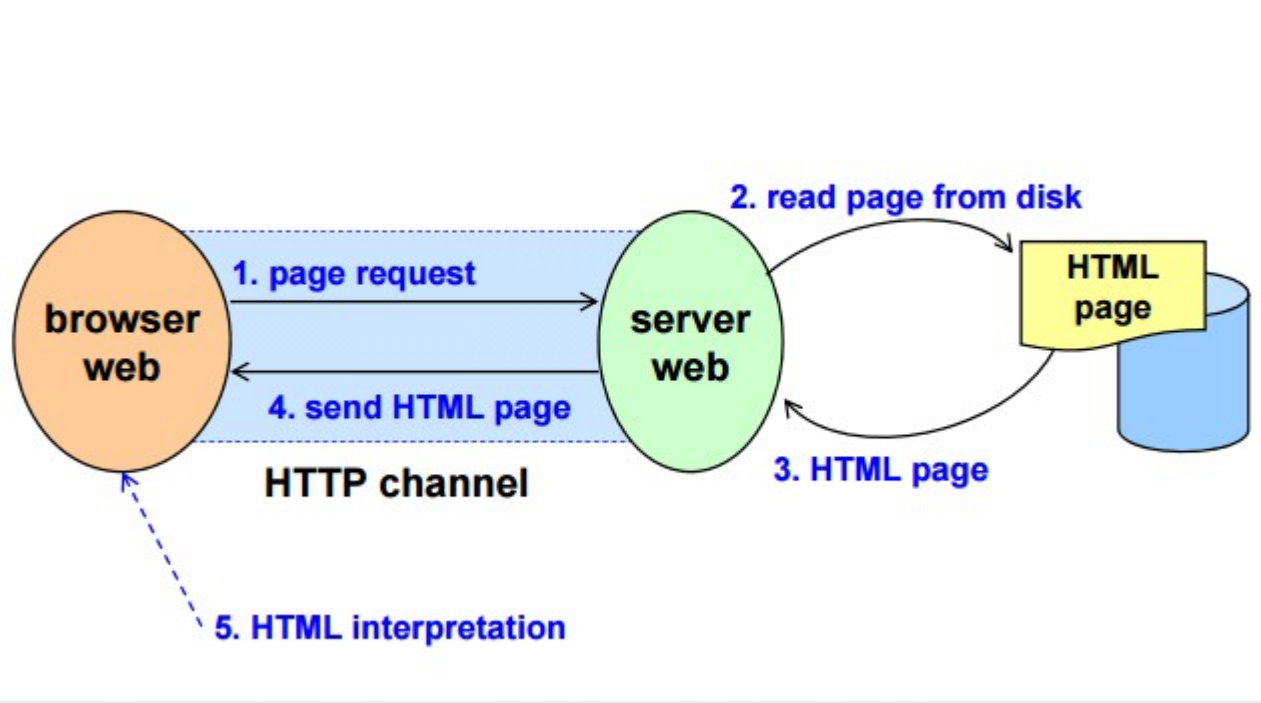
HTTP

- HTTP defines methods to indicate the desired action to be performed on the identified resource.
  - GET - Requests a representation of the specified resource.
  - POST - Requests that the server accept the entity enclosed in the request as a new subordinate of the web resource identified by the URI
  - PUT, DELETE, OPTION
- An HTTP session is a sequence of network request-response transactions.

HTTP

- The first line of the HTTP response is called the status line and includes a numeric status code and a textual phrase
  - 1xx Informational : **100** Continue, **101** Switching Protocols
  - 2xx Success: **200** OK, **201** Created, **202** Accepted, **204** No Content
  - 3xx Redirection: **301** Moved Permanently, **304** Not Modified
  - 4xx Client Error: **400** Bad Request, **403** Forbidden, **404** Not Found
  - 5xx Server Error: **500** Internal Server Error, **501** Not Implemented

The Static Web



The Static web

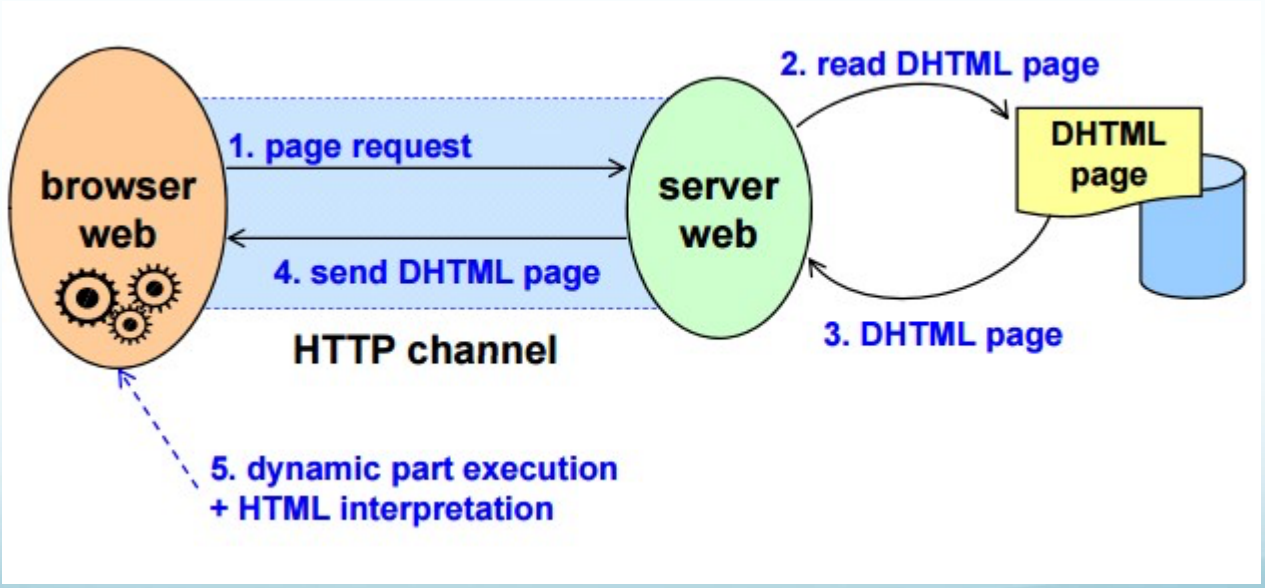
- the web page never changes its content
- … until the author does not change it explicitly
- the content of the page:
  - does not depend on the interaction with the user
  - does not depend on the information sent by the client to the server
  - does not depend on the time it is requested
- Page implemented in HTML / CSS

Static web: pros and cons

- every web page is associated with an HTML file
- (+) maximum efficiency (low load on CPU)
- (+) possibility to perform page caching:
  - in RAM (by the server)
  - on disk (by client)
- (+) pages can be indexed by search engines
- (–) data is static
- (–) no adaptation to clients and their capabilities

Static web with dynamic pages

- the client evaluates the dynamic content of the page (script, or Java applet, or Active-X control)



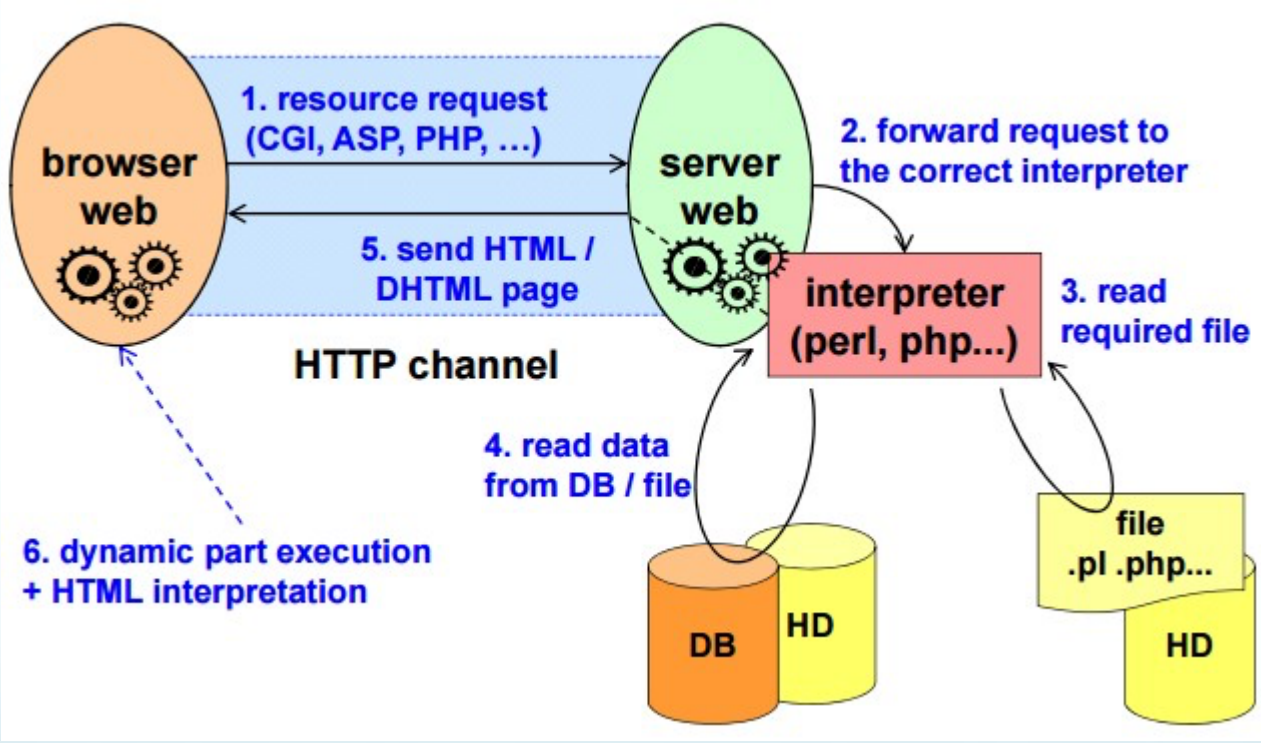
Static web with dynamic pages

- pages change their content depending on user interaction
  - e.g. context menu popping up when the mouse is possitioned over a specific area
- generically known as DHTML:
  - HTML 4.0 or greater
  - CSS (Cascaded Style Sheet)
  - client side scripting languages

Pros and cons

- content presentation is variable
- (+) efficient (low CPU load on servers)
- (–) inefficient (medium-high CPU load on clients)
- (~) possibility to perform page caching
- (~) pages can be indexed by search engines (but only the static data …)
- (–) static data
- (–) functionality depends upon client capabilities

The dynamic web



The dynamic web

- page is dynamically generated by the server
- its information content changes depending on:
  - request sent by the client
  - content of a database
  - date/time of the request
- techniques to implement the dynamic web:
  - CGI
  - server-side scripting language (PHP, PerlScript, Python)
  - servlet, JSP (Java Server Pages)

Dynamic web: pros and cons

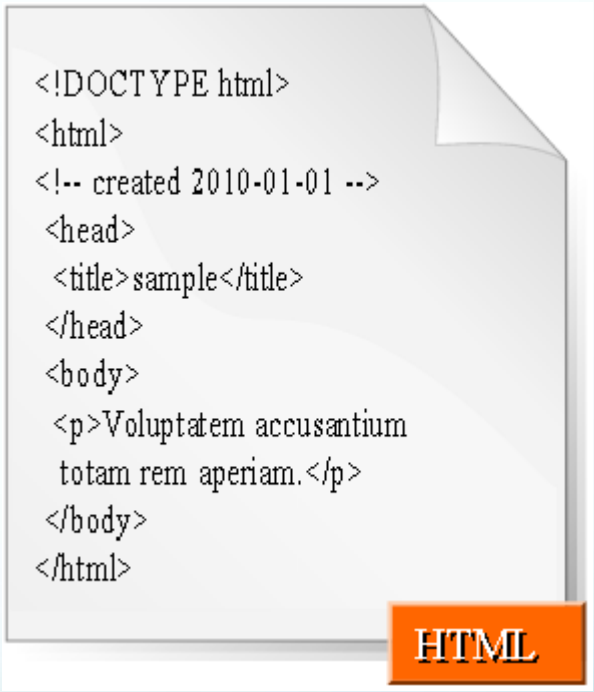
- adaptation of pages to variable conditions
  - input provided by client
  - client capabilities
- (+) maximum dynamicity for the data
- (+) optimal adaptability to clients and their capabilities
- (–) low efficiency (high CPU load)
- (–) pages cannot be indexed by search engines

Server-side or client-side?

- server-side:
  - (pro) higher security
  - (con) server overload
- client-side:
  - (pro) computation on the client
  - (con) client capabilities (functionality and performance)
- in general:
  - better server-side for security and functionality
  - better client-side to improve performance
  - often used together simultaneously

HTML

- HyperText Markup Language, commonly referred to as HTML, is the standard markup language used to create web pages
- Web browsers can read HTML files and render them into visible or audible web pages
- The World Wide Web Consortium(W3C), maintainer of both the HTML and the CSS standards



History

- HTML 2.0 (nov'95 = RFC-1886)
- HTML 3.2 (1996):
  - compatible with 2.0
  - adds tables, applets, superscripts, subscripts, text surrounding images,
- HTML 4.01 (dec' 97 – apr' 98 – dec' 99)
- HTML 5 October 2014
  - was published as a W3C Recommendation

HTML documents

- are normal US-ASCII texts
  - therefore, letters with accents or other “extended” characters are not allowed characters are not allowed
- … enriched with hypertext and hypermedia links
- … and with limited text formatting capabilities
- all these additional capabilities are achieved through annotations expressed with tags

The tags

- enclosed between the symbols “less than" and “greater than“
- usually they are paired (start tag – end tag)

  <h1> … </h1>

- but can also be standalone

- general rule: the final tag is the same of the initial one, preceded by the symbol /
- they are case insensitive in HTML

The attributes

- You can better characterize a tag by using a set of attributes
- Attributes provide additional information about an element
- Attributes come in name/value pairs like: name="value"
- every attribute is placed inside the opening tag

<hr width="90%">

The browsers

- Visualizing HTML documents (and navigate them) requires an appropriate program: an HTML browser
- a browser is an interpreter:
  - reads the source code (HTML + extensions)
  - tries to understand it (hoping there are no errors …)
  - does its best to visualize what is described by the source code
- attention! not every browser visualizes a given attention! not every browser visualizes a given document in the same way
- graphical browsers: Firefox, Chrome, Opera, Internet Explorer, Safari

The browsers

- Visualizing HTML documents (and navigate them) requires an appropriate program: an HTML browser
- a browser is an interpreter:
  - reads the source code (HTML + extensions)
  - tries to understand it (hoping there are no errors …)
  - does its best to visualize what is described by the source code
- attention! not every browser visualizes a given attention! not every browser visualizes a given document in the same way
- graphical browsers: Firefox, Chrome, Opera, Internet Explorer, Safari

General structure of HTML documents

```
<!DOCTYPE HTML PUBLIC ...>
<html>
    <head>
      <title> title </title>
      ... other headers ...
    </head>
    <body>
      text of the document
    </body>
</html>
```

Notes

- browsers do not signal errors: they ignore them!
- white spaces and end of lines:
  - multiple spaces are treated as a single space
  - end-of-lines has no effect on the formatting
- the title (and in general the data inside the head) is very important since it is the element most used by automatic indexing services
- HTML is an extensible language
  - often new tags are added
  - browsers ignore unrecognized tags (or attributes) … but visualize the text enclosed inside the tag

Meta-data

- Inside the HEAD part
- Data useful for:
  - Indexing the HTML page indexing the HTML page
  - Providing information to the web server and / or to the browse
- the title (and in general the data inside the head) is very important since it is the element most used by automatic indexing services
- Syntax:

```
<meta name="author" content="Antonio Lioy">
<meta name="keywords" content="html">

<meta http-equiv="Content-Type"
  content="text/html; charset=ISO-8859-1">
<meta http-equiv="Expires"
  content="Sun, 28 Feb 2010 23:59:00 GMT">
```

Tools for checking HTML

- http://validator.w3.org
  - allows verifying if a page fully satisfies the official syntax
  - can provide detailed explanations on the errors and on how to correct them
- http://tidy.sourceforge.net:
  - “cleans” the HTML code and transforms it to more recent versions recent versions
- problems with dynamically generated HTML (cannot validate an ASP or PHP source page)

Comments

- can be inserted at every point in the text
- can span multiple lines
- enclosed inside **<!--** and **-->**
- Examples:

```
<!-- this is a comment -->

<!--
this comment
spans four lines
-->
```

Headings

- there are six levels of headings or titles:
- should be used according to the logical meaning (semantics), not to achieve a specific formatting
- in particular, it is not correct to use <hN> if not preceded by <hN-1>

- `<h1>...</h1>`
- `<h2>...</h2>`
- `<h3>...</h3>`
- `<h4>...</h4>`
- `<h5>...</h5>`
- `<h6>...</h6>`

Text blocks

- …
  - starts and terminates a paragraph
  - after terminating a paragraph browsers break the after terminating a paragraph, browsers break the current line (and may also insert a small vertical space)
-
 or
  - inserts a line break inserts a line break
- <hr> or <hr/>
  - inserts an horizontal rule (line)

List

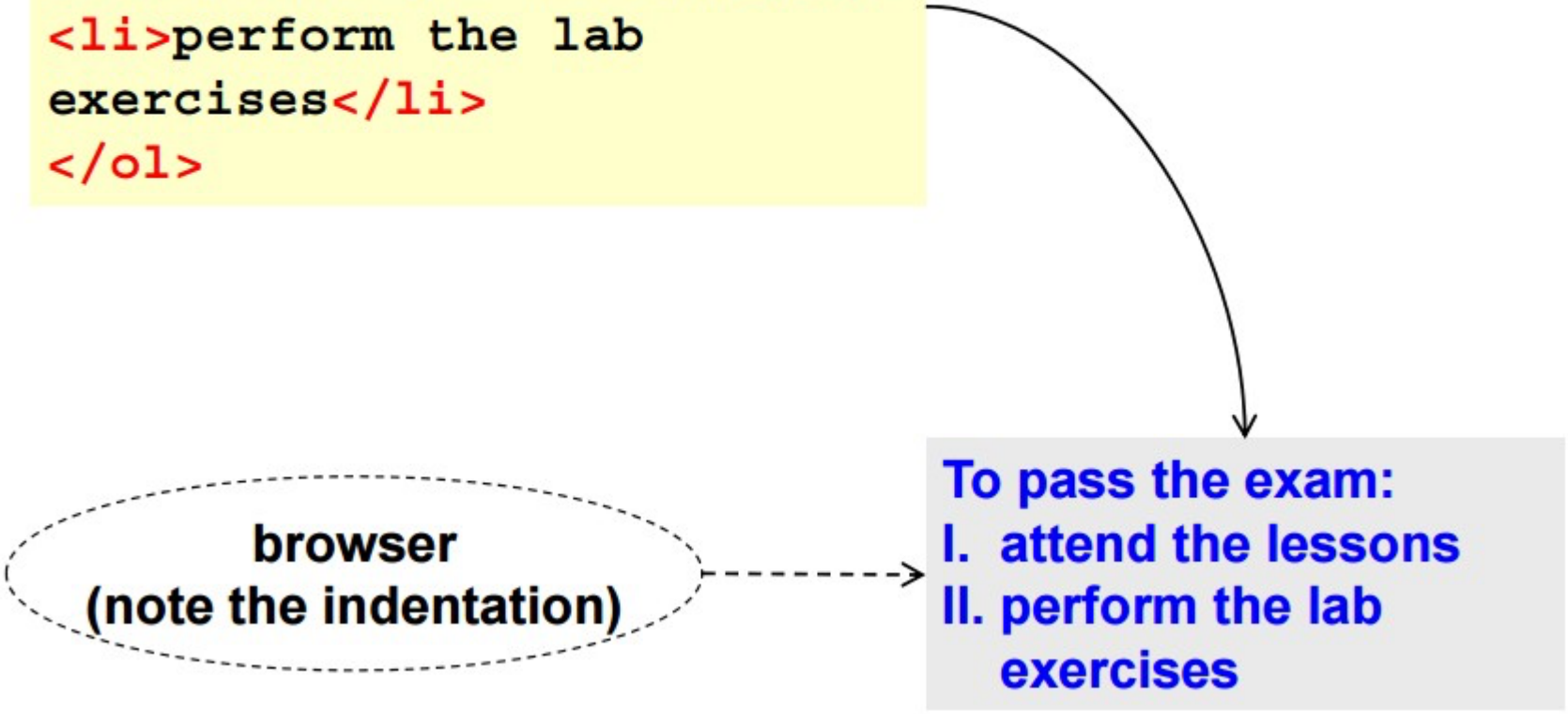
- Unordered list:
  - . . .
- Ordered list:
  - . . .
- An element of (any) list:>
  - . . .

Options for lists

- symbol preceding the items in unordered lists:
  - type=disc / circle / square
- numbering style in ordered lists:
  - start= index_of_the_first_item
  - type=A / a / I / i / 1
  - that is:
    - alphabetic list (uppercase or lowercase)
    - roman numbers (uppercase or lowercase)
    - decimal numbers
- can be specified for the whole list (ol) and for the single element (li)

List example

```
To pass the exam:
<ol type="I">
<li>attend the lessons</li>
<li>perform the lab
exercises</li>
</ol>
```

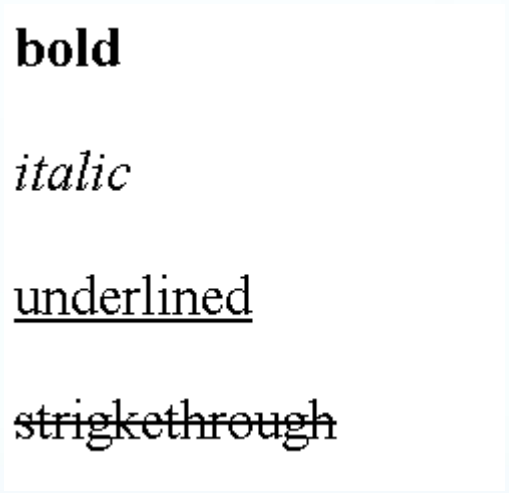


Text formatting

- a text block can be characterized based on the role it plays in the document (logical style) …
- … or based on the way we want to visualize it … or based on the way we want to visualize it physically (physical style)
- best to prefer logical styles and to leave greater freedom to the final user in defining how the text should appear on the screen
- with XHTML (strict), the formatting tags have finally disappeared (you need to use CSS)

Formatting: physical styles

- …
  - bold text
- or
  - italic text
- <u> . . . </u>
  - underlined text
- <s> . . . </s>
  <strike> . . . </strike>
  - Strikethrough text



Formatting: text blocks

- <address>. . . </address>
  - address (typically e-mail)
- <blockquote> . . . </blockquote>
  - Long Citation
- . . .
  - Centered text
- <pre> . . . </pre>
  - Preformatted text (spacing is preserved)

Links (Hyperlinks)

- By using hyperlinks you can move automatically from a resource to another
- the HTML tag identifying the presence of a link is named anchor, and is identified with <a>

How to insert an hyperlink

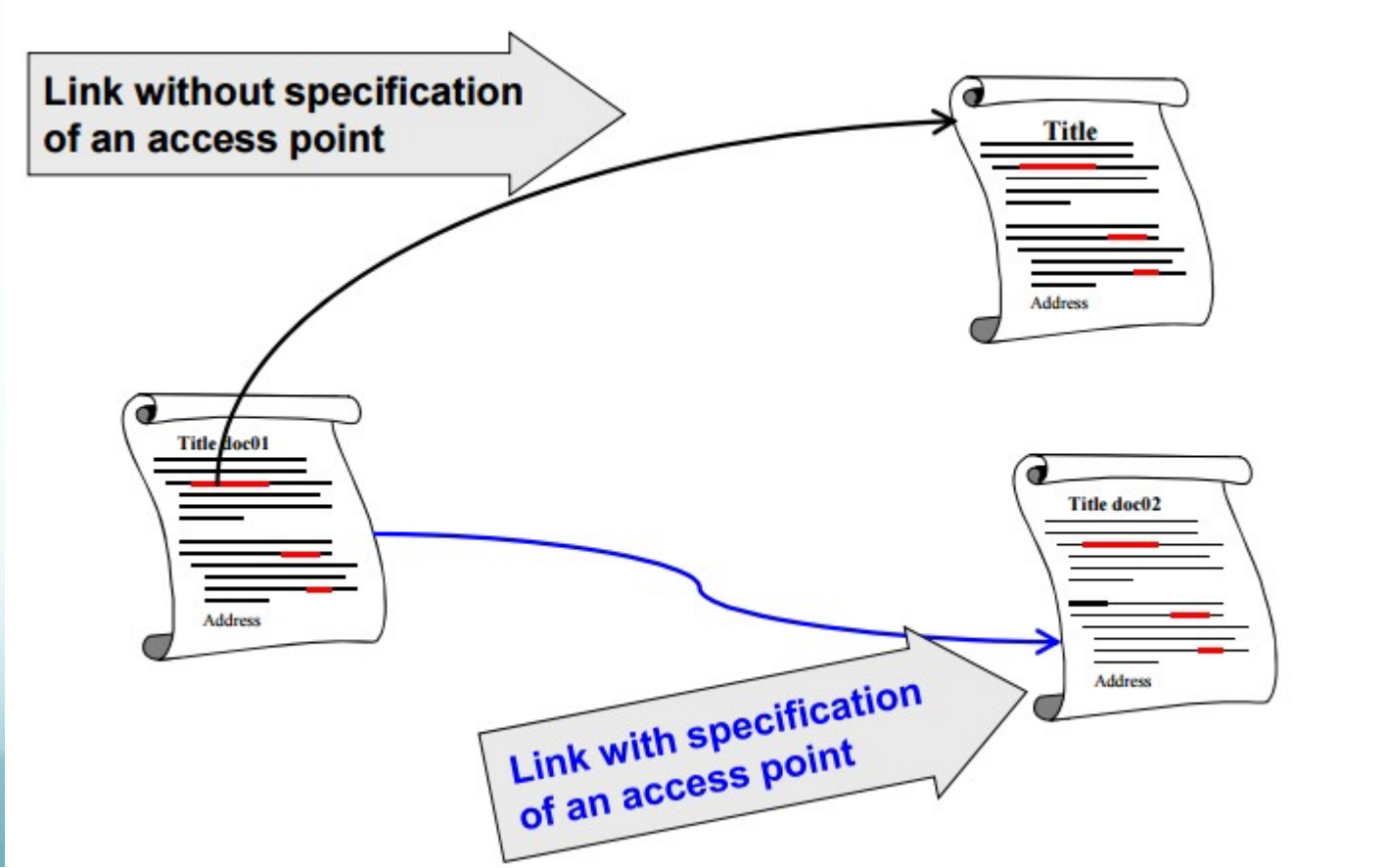
- open the anchor opening tag: <a
- insert a space
- insert the URL of the resource, preceded by href= and enclosed by apices
- close the opening tag with >
- insert the text to highlight (the one associated with the anchor, called "hot word")
- close the anchor:
- Google

Absolute and relative links

- it is possible to omit parts of the URL
  - in this case, it is called a “relative” link
  - the missing parts assume the same value of the the missing parts assume the same value of the current page
- examples of relative links (supposed to be placed inside the page [http://www.lioy.it/01eny/exam.html)](http://www.lioy.it/01eny/exam.html)

relative *link*	absolute *link*
`biblio.html`	`http://www.lioy.it/01eny/biblio.html`
`../cv.html`	`http://www.lioy.it/cv.html`
`res/a1.html`	`http://www.lioy.it/01eny/res/a1.html`

Document access points



Document access points

- in the target document, define the access point through an anchor with the attribute NAME
  -
    Hello world

- in the origin document, include the name of the access point in the URL
  - <a href="doc2.html#hello">
- the access point can also be any element identified through its "id"
  - <h1 id="hello">
    Say Hello Again
    </h1>

Frame navigation

- links should indicate into which frame (or window) the target page should be visualised:
  - . . .
- special values for target
  - "_blank" (new window)
  - "_self" (in the same frame) = default
  - "_top" (span the entire window)

Images

- <img src="image.png">
  - inserts the image contained in the file image.png
- <img src="image.png" alt="Picture of something">
  - Inserts the image, but if the browser does not support graphics or fails to load the URL, It visualizes the text
- difference between linking and inserting an image:
  - <img src="image.png">
    (inserts the image inside the page)
  - <a href="image.png"> (by following the link, you visit a page containing only the image)

Image formatting

- <img width="w" height="h" ...>
  - Image size
  - allows rapid visualization of the page (the browser does not need to download the image before knowing how much space should be reserved to it)
- <img vspace="v" hspace="h" ...>
  - minimum distance between text and the image
- <img border=b ...>:
  - Size of border

Image formatting

- . . .
  - font for the text block included within the tags
  - deprecated (use CSS) deprecated (use CSS)
- Attributes:
  - size=
  - color=
  - font-family=

Colors

- some predefined colors are accessible by name
  - Black, White, Gray, Silver, Yellow, Red, Purple, Fuchsia, Yellow, Red, Purple, Fuchsia, Maroon, Green, Lime, Olive, Aqua, Teal, Blue, Navy
- other colors can be specified through their RGB hexadecimal code (# rr gg bb)
- Example:
  - White!

Standard colors

Tables

- <table . . . > . . . <table>
- Attributes:
  - align= left / center / right
  - border= size
  - width= size
  - cellspacing= size
  - cellpadding=size
  - summary= text
  - frame= void / above / below / hsides / lhs / rhs / vsides / box / border

Table data

- <tr . . . > . . . </tr>
  - A row of the table
  - Contains normal (<td>) or heading (<th>) cells
- <th . . .> . . . </td>
  <td . . .> . . . </td>
  - table data (or heading), which can span multiple cells, horizontally or vertically
  - colspan= number-of-columns
  - rowspan= number-of-rows

Optional elements of a table

- <thead . . . > . . . </thead>
  - heading
- <tbody . . .> . . . </tbody>
  - Content block
- <tfoot . . .> . . . </tfoot>
  - footer
- <caption . . .> . . . </caption>
  - Caption text describing the nature of the table

DIV and SPAN

- introduced in HTML 4.0
- to group parts and apply formatting more easily
- DIV identifies a block (typically, browsers place a line break before and after a block)
- SPAN identifies a part inside a block
- frequently used to create (with an appropriate CSS) a page layout without using tables or frames
- “id” and “class” allow references from the CSS “id” and “class” allow references from the CSS

General attributes of HTML tags

- id=“string”
  - anchor for a link
  - reference to an element from a script
  - reference to a field in a form
  - reference for a specific style in CSS
- class = "class1 class2 ..."
  - list of classes to be used e.g. as CSS selectors
- title = "title”
  - visualised when pointing to the element
- lang = "language"
  - for automatic text reading (values: en it fr de ...)

Favorite icon

- the little icon near the URL
  - a 16 x 16 pixel image
- old browsers:
  - only in MS icon format
  - in a fixed position and with fixed name = /favicon.ico
- first step to standardization:

```
<link rel="shortcut icon" href="/icons/my.ico"
  type="image/vnd.microsoft.icon">
```

- new browsers support the de-facto standard

```
<link rel="icon"
  type="image/png" href="/icons/my.png">
```

<input> Tag

- The <input> tag specifies an input field where the user can enter data.
- The <input> element is the most important form element.
- The <input> element can be displayed in several ways, depending on the type attribute

- The different input types are as follows:

  text, password, number, button, radio, checkbox, date, time, datetime, hidden, email, submit, file…

<select> Tag

- The <select> element is used to create a drop-down list
- The different input types are as follows:

```
<select name="cars" id="cars">
  <option value="volvo">Volvo</option>
  <option value="saab">Saab</option>
  <option value="mercedes">Mercedes</option>
  <option value="audi">Audi</option>
</select>
```

Audio and Video

- The <audio> tag is used to embed sound content in a document, such as music or other audio streams.

- The HTML <video> element is used to show a video on a web page.

```
<audio controls>
  <source src="horse.ogg" type="audio/ogg">
  <source src="horse.mp3" type="audio/mpeg">
  Your browser does not support the audio tag.
</audio>
```

```
<video width="320" height="240" controls>
    <source src="movie.mp4" type="video/mp4">
    <source src="movie.ogg" type="video/ogg">
     Your browser does not support the video tag.
</video>
```

Thank you