

## 03 – Web 1.0 (part 1)

## ■ What is Web 1.0?

- Web 1.0 is a *retronym*, referring to the World Wide Web *before Web 2.0*
  - Web 1.0 begins with the birth of the WWW (1991)...
  - ... but its end depends on the beginning of Web 2.0!
  - The term Web 2.0 has been coined in 2004, however its “date of birth” is still rather fuzzy
- Web 1.0 definition is not only temporal, but also refers to some particular characteristics which differentiate it from its successors
  - Static, non interactive websites
  - More value in the (closed) app than in the data
  - Less centered on users
  - No social networks! ;-)

- *Some sources* also try to spot characteristic Web 1.0 design patterns such as:
  - framesets
  - tables + spacers to position elements on a page
  - proprietary HTML extensions such as `<blink>` or `<marquee>`
  - ...
- This is somehow true, but trying to identify a website as Web 1.0 or not only from the design is reductive at best
  - the absence of those (bad) patterns does not mean the site is new or 2.0
  - their presence only gives hints about the (bad) designer

- In the beginning was the hypertext
  - A way for constructing documents that reference other documents
  - A block of text in a hypertext can be tagged as a *hyperlink* which points to another document
- Characteristics of the first hypertexts on the Web:
  - Everyone can link anything: no matter whether you own a content or not, you can always link it from your Web page
  - Backlinks are not explicit: you never know if someone links your page (this has not changed much)
  - Few multimedia contents: due to scarcity of bandwidth and file formats, Web pages were rather poor if compared to current ones


# Web 0.something?


- Same age as the WWW, *Gopher* is another system for document sharing across the Internet
  - Hypertextual
  - Uses another TCP-based protocol (name is Gopher)
  - Mainly relies on textual documents
  - Has its own search engines (i.e. Veronica)
  - Requires much less bandwidth
  - Requires much, much less computational power, archiving space, RAM memory etc. (a Gopher server has been seen running on a Nokia phone!)
- Not every document is a hypertext: this technology is used to manage hierarchical menus which contain documents
  - With its hierarchical organization, Gopher actually looks more like a distributed filesystem
  - The interface, as seen from the browser, recalls an “FTP with rich directories”

## Indice di gopher://gopher.floodgap.com/


Welcome to Floodgap Systems' official gopher server.  
Floodgap has served the gopher community since 1999  
(formerly gopher.ptloma.edu). \*\* OVER A DECADE OF SERVICE! \*\*

We run Bucktooth 0.2.8 on inetd as our server system.  
gopher.floodgap.com is an Apple Power Macintosh 7300 with  
a G3/500 processor and 1GB RAM running NetBSD/macppc.  
Send gopher@floodgap.com your questions and suggestions.

 [Does this gopher menu look correct?](#)  
(plus using the Floodgap Public Gopher Proxy)

 [Super-Dimensional Fortress: SDF Gopherspace](#)  
Get your own Gopherspace and shell account!

--- Getting started with Gopher -----

 [Getting started with gopher, software, more](#)  
(what is Gopherspace? We tell you! And find out how  
to create your own Gopher world!)

# HTTP (again?)

- HTTP is the protocol on which the Web (1.0 and following) relies and is documented in [RFC 2068](#)
- The client sends an HTTP request (GET/POST/HEAD) to the server and this one sends an HTTP reply (telnet to port 80 to see that live)
- Different information is exchanged between the two:
  - Server: HTTP version, date, software+version, etc.
  - Client: requested page, referer, user agent, etc.
  - ... and of course in both cases there are also *IP addresses*
- Client information can be used by the server to provide different contents according to
  - provenance (both geographical and referer Web page)
  - Web client application (user agent)
  - ... and of course the requested page ;-)

- HyperText Markup Language is the first and main format used to create Web pages
  - Currently at version 5... but *that* is definitely not Web 1.0!
  - First versions were not even real standards (search the Web for “*browser wars*” to better understand it)
- HTML *marks up* portions of text and specifies their characteristics (originally mostly appearance, lately more their purpose within the document)
- (good?) old HTML:
  - Describes a document in terms of a HEAD (metadata) and a BODY (actual content)
  - Provides markup for section headers (h1, h2, ...), fonts (type, face, size), text alignment, internal and external anchors, page organization with frames and tables, embedding of objects (images or binary files in general)



# How does HTML work?

HTML pages are plain text files following a specific format

## ■ Page creators

- Can use many different applications to create them, from a simple text editor to a more complex graphical development environment

## ■ Page readers

- See pages rendered in their browsers (which automatically take care of the “techie part of the Web”)
- When a browser finds an embedded object, it downloads and displays it within the page
- Can always request the page HTML source, read it, and save it (not on the server, of course)

# What can (old) HTML do?

- HTML 5 provides new exciting functionalities to the standard and the browsers that support it. To better understand them, it is important to understand what “old” HTML was able to provide or not
- Plain old HTML can
  - Describe a hypertext
  - Display pages with embedded multimedia contents or links to binary files for download
  - Trigger the execution of external apps when needed (i.e. open pdf files, run the mail client, etc.)
- Old HTML alone cannot
  - Provide “dynamic” contents (i.e. contents that change according to given parameters, or that are built dynamically from an external data source as a DB)
  - Provide a Web application (i.e. a Web page that updates parts of its content according to interactions with the user)

- Every resource on the Web (no matter if it is a Web page, a binary file, an external link or an embedded object) can be located by knowing its Uniform Resource Locator (URL). See [RFC 1738](#).
- URLs are unique “addresses” for resources
  - HTTP URLs always begin with the “http://” prefix, which defines the *scheme* (others exist such as ftp://, gopher://, mailto:, etc.)
  - After the prefix, the *host name* follows (optionally with a *port*)
  - After the domain, a *path* follows (like with filesystems), which might end with a page name or not (defaults can be applied)
  - In the case of dynamic pages, which require parameters, a set of parameters might follow too  
(i.e. <http://example.com/my/path/file?param1=val1&param2=val2&etc> )

# HTML and accessibility problems

- Websites (and consequently HTML pages) should be built for *accessibility*. The W3C [guidelines for accessibility](#) suggest the following patterns, across four macro-areas
  
- 1. “Perceivable”
  - Provide text alternatives for any non-text content so that it can be changed into other forms people need, such as large print, braille, speech, symbols or simpler language.
  - Provide alternatives for time-based media
  - Create content that can be presented in different ways (for example simpler layout) without losing information or structure
  - Make it easier for users to see and hear content including separating foreground from background

# HTML and accessibility problems

## ■ 2. “Operable”

- Make all functionality available from a keyboard
- Provide users enough time to read and use content
- Do not design content in a way that is known to cause seizures
- Provide ways to help users navigate, find content, and determine where they are

## ■ 3. “Understandable”

- Make text content readable and understandable
- Make Web pages appear and operate in predictable ways
- Help users avoid and correct mistakes

## ■ 4. “Robust”

- Maximize compatibility with current and future user agents, including assistive technologies

# From directories to search engines

- Before search engines, when pages were few enough to be manually visited, directories existed to classify and organize Web contents
  - The **first Web page**, by Tim Berners-Lee, was a directory linking to various resources on the Web

## World Wide Web

The WorldWideWeb (W3) is a wide-area [hypermedia](#) information retrieval initiative aiming to give universal access to a large universe of documents.

Everything there is online about W3 is linked directly or indirectly to this document, including an [executive summary](#) of the project, [Mailing lists](#) , [Policy](#) , November's [W3 news](#) , [Frequently Asked Questions](#) .

### [What's out there?](#)

Pointers to the world's online information, [subjects](#) , [W3 servers](#), etc.

### [Help](#)

on the browser you are using

### [Software Products](#)

A list of W3 project components and their current state. (e.g. [Line Mode](#) , [X11 Viola](#) , [NeXTStep](#) , [Servers](#) , [Tools](#) , [Mail robot](#) , [Library](#) )

- Search engines allow a faster access to documents when page number increases

## ■ Advantages of directories

- Websites reviewed by humans (potentially providing additional metadata)
- Contain fewer resources than search engines
- Increase the probability of retrieving relevant results

## ■ Weaknesses

- The hierarchical classification has been built top-down, in an arbitrary way
- Updates are not as frequent as search engines'
- User annotations are subjective (no "wisdom of the crowds" effect)

## ■ Examples

- <http://www.dmoz.org/>
- <http://dir.yahoo.com/>

- Search engines are roughly composed by three components:
  - A *crawler* (or *spider*), that downloads pages from the Web
  - An *indexer*, which *tokenizes* page contents and indexes documents according to the tokens they contain
  - An *interface* to provide search results, *ranked* in the proper way
  
- Main problems:
  - index creation, compression, and update
  - stopwords and stemming
  - relevance ranking (recall vs precision)
  - relevance feedback



## ■ Some Web references:

- Is there a Web 1.0? <http://computer.howstuffworks.com/web-10.htm>
- A tribute to Web 1.0:  
[http://www.yvoschaap.com/weblog/rip\\_a\\_tribute\\_to\\_web\\_10/](http://www.yvoschaap.com/weblog/rip_a_tribute_to_web_10/)
- Web 1.0 revisited: <http://www.complexify.com/buttons/>
- An introduction to “old” HTML:  
<http://www.utoronto.ca/webdocs/HTMLdocs/NewHTML/intro.html>
- Directories and Search Engines on DMOZ:  
<http://www.dmoz.org/Computers/Internet/Searching/>
- Search Engine History: <http://www.searchenginehistory.com/>
- How does Google collect and rank results?  
[http://www.google.com/librariancenter/articles/0512\\_01.html](http://www.google.com/librariancenter/articles/0512_01.html)