



中山大學

SUN YAT-SEN UNIVERSITY

# Lecture 01. Web Basics

**Modern Web Programming**

(<http://my.ss.sysu.edu.cn/wiki/display/WEB/> supported by Deep Focus)

School of Data and Computer Science, Sun Yat-sen University

# Outline

---

- **The Internet**
- The World Wide Web (WWW)
- Web 2.0

# What's the Internet?



- A Chinese town officer
  - “The Internet is an English agents’ net”
- A U.S. Senator
  - “series of tubes” (explanation)
- How many internets are there, anyway? Is The Google one of them?

# The Internet

---



- Wikipedia: <http://en.wikipedia.org/wiki/Internet>
- a connection of computer networks using the Internet Protocol Suite (**TCP/IP**)
- What's the difference between the Internet and the World Wide Web (**WWW**)?
- **WWW = HTML\* + HTTP(S)**
  - \* including CSS, JavaScript, and other browser enabled content*

# Brief History

---

- began as a US Department of Defense network called ARPANET (1960s-70s)
- initial services: electronic mail, file transfer
- opened to commercial interests in late 80s
- WWW created in 1989-91 by Tim Berners-Lee
- popular web browsers released: Netscape 1994, IE 1995
- Amazon.com opens in 1995; Google January 1996
- Chinese First Connection with Internet: Chinese Academics Net, by Computer Applying Technology Institute of Beijing 1986,
- First email, Sep. 14 1987, from CATIB, “Across the Great Wall we can reach every corner in the world”
- Chinese First Full Internet Connection: **NCFC** (National Computing and Networking Facility of China) 1994
- **Baidu** 1999; **Taobao** 2003

# Who can shut down the Internet?

---

Baidu 百度



Google



**GFW**

# Key aspects of the Internet

---

- Internet is for freedom of information
- internet Vs. Internet
- subnetworks can stand on their own
- computers can dynamically join and leave the network
- built on open standards; anyone can create a new device
- lack of centralized control (mostly)
- everyone can use it with simple, commonly available software

# People and Organizations

---

- Internet Engineering Task Force (IETF): internet protocol standards
- Internet Corporation for Assigned Names and Numbers (ICANN): decides top-level domain names
- World Wide Web Consortium (W3C): Web standards

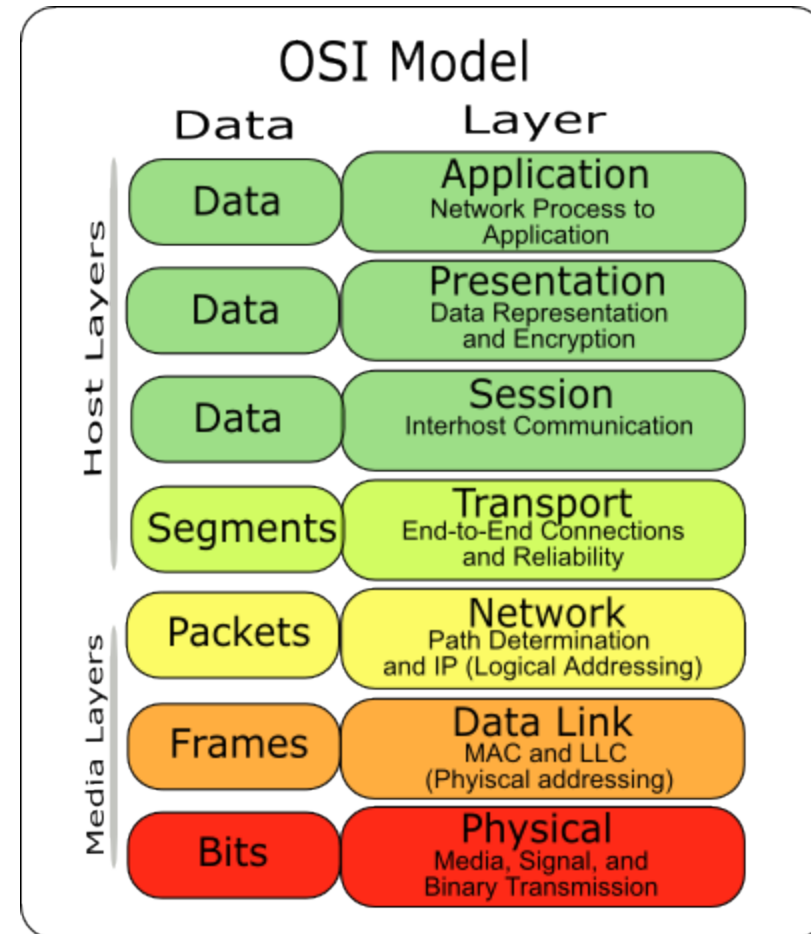




# Layered architecture

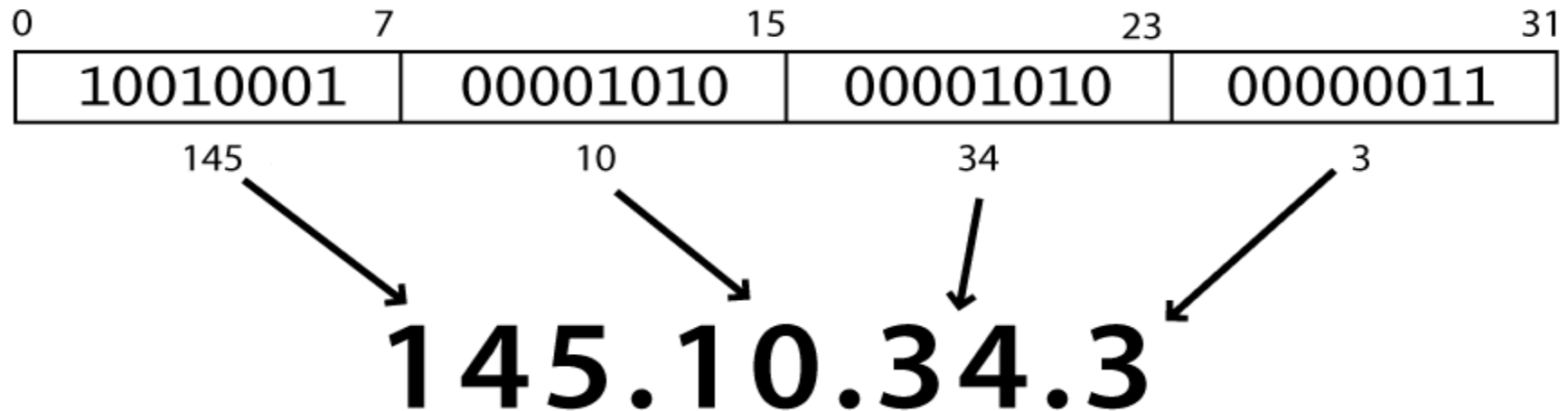
The internet uses a layered hardware/software architecture (OSI model):

- **physical layer**: devices such as coaxial cables, fiber-optic lines, modems
- **data link layer**: basic hardware protocols (Ethernet, Wi-Fi, DSL, ATM, PPP)
- **network / internet layer**: basic software protocol (IP)
- **transport layer**: add reliability to network layer (TCP, UDP)
- **application layer**: implements specific communication for each kind of program (HTTP, POP3/IMAP, SSH, FTP)



# Internet Protocol (IP)

- the **IP** is the underlying system of communication for all data (packets) sent across the internet.
- each device has a 32-bit IP address as four 8-bit numbers



- find out your internet IP address: [whatismyip.com](http://whatismyip.com)
- find out your local IP address:
  - in a terminal, type: **ipconfig** (Windows) or **ifconfig** (Mac/Linux)
- IP v4 vs. IP v6 (32-b vs. 128-b)

# Transmission Control Protocol (**TCP**)

---

- adds multiple, guaranteed message delivery on top of IP
- **multiplexing**: multiple programs using the same IP address
  - **port**: a number given to each program or service
  - 80: Web browser (443 for secure browsing)
  - 25: email
  - 22: ssh
  - 21: ftp
  - more common ports
- some programs (QQ, games, streaming media programs) use simpler UDP protocol instead of TCP
- find out ports used:
  - in a terminal, using **netstat** (Windows) command
  - using CurrPorts

# Outline

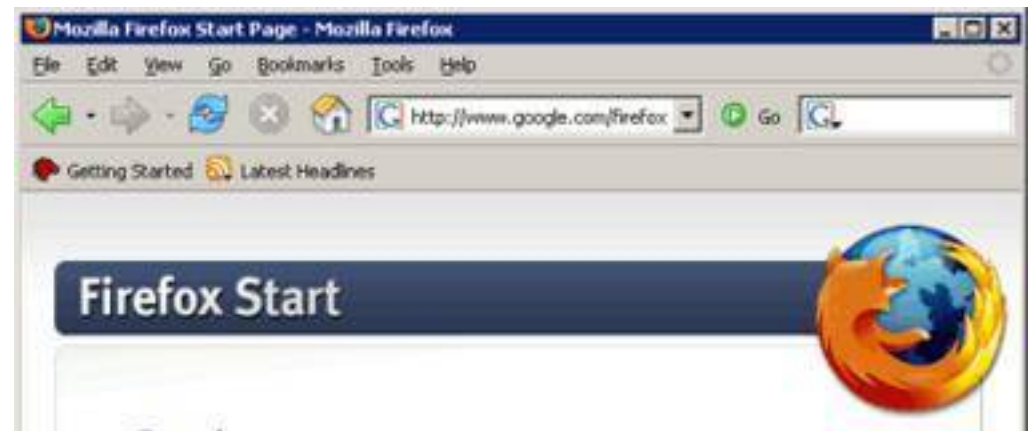
---

- The Internet
- **The Word Wide Web (WWW)**
- Web 2.0

# Web servers and browsers



- **Web server:** software that listens for Web page requests
  - [Apache](#)
  - Microsoft Internet Information Server (IIS) ([part of Windows](#))
- **Web browser:** fetches/displays documents from Web servers
  - Microsoft [Internet Explorer](#) (IE)
  - [Mozilla Firefox](#)
  - Apple [Safari](#)
  - [Google Chrome](#)
  - [Opera](#)



# Domain Name System (**DNS**)

---

- a set of servers that map written names to IP addresses
  - Example: [www.sysu.edu.cn](http://www.sysu.edu.cn) → 202.116.64.9
  - using Windows command **nslookup** to find out IP address
  - non-English languages in domain name [IDN ccTLD Fast Track](#)
- many systems maintain a local cache called a [host file](#)
  - Windows: [C:\Windows\system32\drivers\etc\hosts](#)
  - Mac: [/private/etc/hosts](#)
  - Linux: [/etc/hosts](#)

# Uniform Resource Locator (**URL**)

---

- an identifier for the location of a document on a web site
- a basic URL:

<http://www.aw-bc.com/info/regesstepp/index.html>

~~~ ~~~~~  
 protocol      host                                      path

- upon entering this **URL** into the browser, it would:
  - ask the DNS server for the IP address of `www.aw-bc.com`
  - connect to that IP address at port 80
  - ask the server to GET `/info/regesstepp/index.html`
  - display the resulting page on the screen

# More advanced URLs

---

- **anchor**: jumps to a given section of a web page  
<http://www.textpad.com/download/index.html#downloads>
  - fetches index.html then jumps down to part of the page labeled downloads
- **port**: for web servers on ports other than the default 80  
<http://www.cs.washington.edu:8080/secret/money.txt>
- **query string**: a set of parameters passed to a web program  
<http://www.google.com/search?q=miserable+failure&start=10>
  - parameter `q` is set to "miserable+failure"
  - parameter `start` is set to 10



# Hypertext Transport Protocol (HTTP)

- the set of commands understood by a web server and sent from a browser
- some HTTP commands (your browser sends these internally):
  - **GET** *filename* : download
  - **POST** *filename* : send a web form response
  - **PUT** *filename* : upload
  - **DELETE** *filename*: remove entity
  - **HEAD** *filename*: only status information, not entire content
- simulating a browser with a terminal window:

```
$ telnet www.sysu.edu.cn 80  
Trying 202.116.64.9... Connected to 202.116.64.9  
(202.116.64.9). Escape character is '^'.  
GET /2009/xxgk.html  
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML  
4.0 ..."> <html> ...
```

# HTTP error codes

---

- when something goes wrong, the web server returns a special "error code" number to the browser, possibly followed by an HTML document
- common error codes:

| Number                        | Meaning                                     |
|-------------------------------|---------------------------------------------|
| 200                           | OK                                          |
| <a href="#">301-303</a>       | page has moved (permanently or temporarily) |
| <a href="#">403</a>           | you are forbidden to access this page       |
| <a href="#">404</a>           | page not found                              |
| 500                           | internal server error                       |
| <a href="#">complete list</a> |                                             |

# Internet media (**MIME**) types

---

- sometimes when including resources in a page (style sheet, icon, multimedia object), we specify their type of data

| MIME type                | file extension             |
|--------------------------|----------------------------|
| text/html                | .html , .htm, shtml, .shtm |
| text/plain               | .txt                       |
| image/gif                | .gif                       |
| image/jpeg               | .jpg                       |
| video/quicktime          | .mov                       |
| application/octet-stream | .exe                       |

- Lists of MIME types: [by type](#), [by extension](#)
- .html vs. .htm

# Web languages / technologies

---

- Hypertext Markup Language ([HTML](#)): used for writing web pages
- Cascading Style Sheets ([CSS](#)): stylistic info for web pages
- PHP Hypertext Processor ([PHP](#)): dynamically create pages on a web server – *of course, there are many other languages and scripts can do this ...*
- [JavaScript](#): interactive and programmable web pages
- Asynchronous JavaScript and XML ([Ajax](#)): accessing data for web applications
- eXtensible Markup Language ([XML](#)): metalanguage for organizing data
- Structured Query Language ([SQL](#)): interaction with databases
- Resource Description Frame ([RDF](#)): describing web resources semantically
- .....

# Terms

---

- **Internet Service Provider (ISP)**
  - enterprises or organizations who provide Internet access to you,
  - who? please identify your ISPs
- **Web Hosting**
  - provide a place for consumers to store pages designed to be consumed by the Web surfing public
  - ISPs often offer Web hosting services along with their standard connectivity packages.
- **Client/Server vs. Browser/Server**
- **Presentation Layer**
  - often refers to the top tier of enterprise application architecture
  - in Web, it includes both codes of Web pages and codes of producing Web pages.
- **Client Side Scripting/programming**
  - writing codes consumed by browsers to render Web pages and to respond to user's interactions
- **Server Side Scripting/programming**
  - writing codes used to generate the codes consumed by browsers

# Outline

---

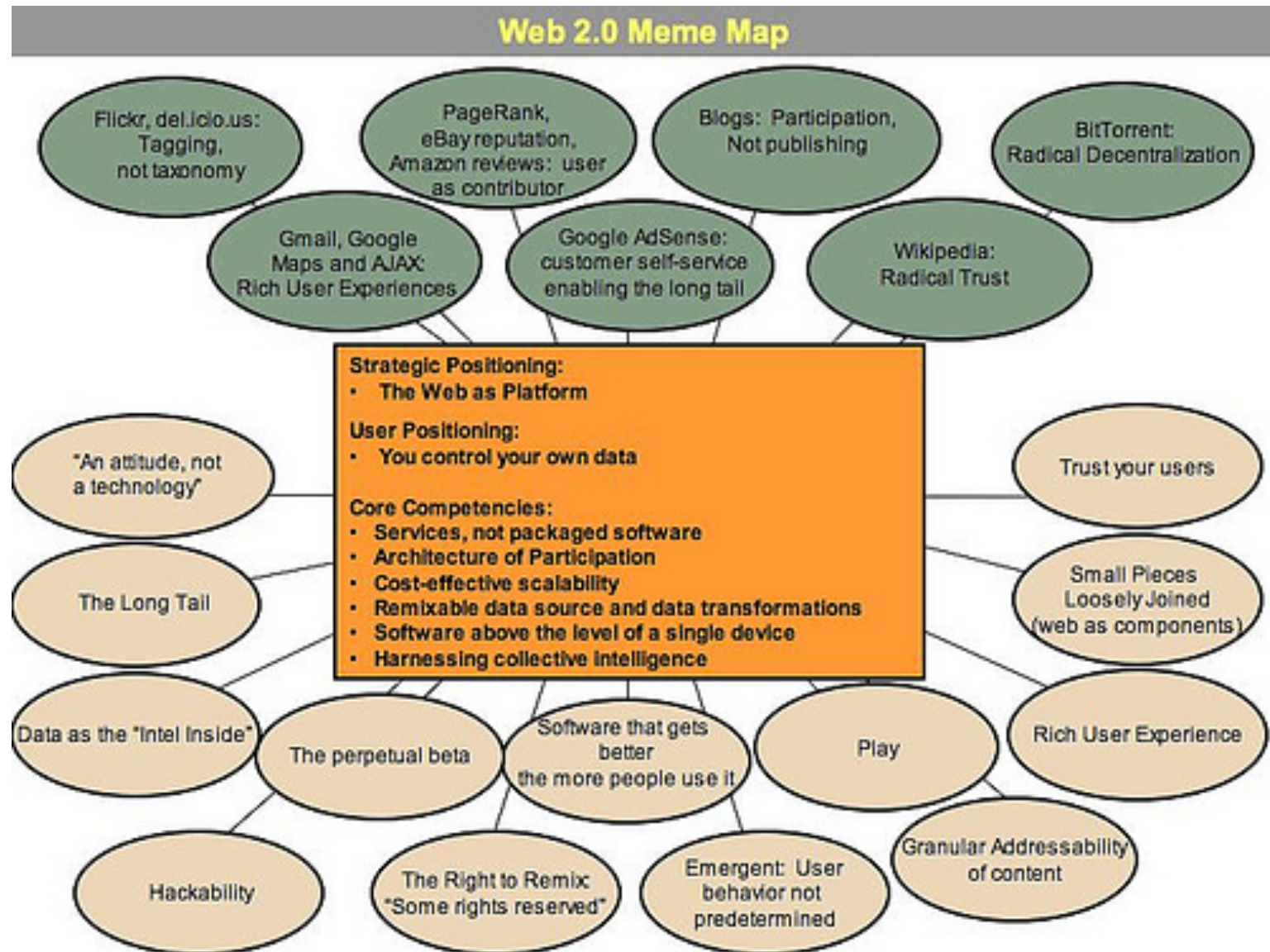
- The Internet
- The Word Wide Web (WWW)
- **Web 2.0**

# Web 1.0 vs. Web 2.0

- **Web 1.0** is about **publishing**
  - users are limited to the passive viewing of information that is provided to them
- **Web 2.0** is about **interaction**
  - allows its users to interact with other users or to change website content
  - information sharing, interoperability, user-centered design and collaboration
  - hosted services, web applications, social-networking sites, video-sharing sites, wikis, blogs, mashups and folksonomies.
  - coined by Tim O'Reilly because of the O'Reilly Media Web 2.0 conference in 2004



# Web 2.0 memo map





# Web 2.0 examples

| Web 1.0                    |   | Web 2.0                    |
|----------------------------|---|----------------------------|
| DoubleClick                | → | Google AdSense             |
| Ofoto                      | → | Flickr                     |
| Akamai                     | → | BitTorrent                 |
| mp3.com                    | → | Napster                    |
| Britannica Online          | → | Wikipedia                  |
| personal websites          | → | blogging                   |
| evite                      | → | upcoming.org and EVDB      |
| domain name speculation    | → | search engine optimization |
| page views                 | → | cost per click             |
| screen scraping            | → | web services               |
| publishing                 | → | participation              |
| content management systems | → | wikis                      |
| directories (taxonomy)     | → | tagging ("folksonomy")     |
| stickiness                 | → | syndication                |

## 2.0 flurry

---

- Library 2.0, Classroom 2.0, Publishing 2.0,
- Social Work 2.0, Enterprise 2.0, PR 2.0,
- Medicine 2.0, Telco 2.0, Travel 2.0
- Government 2.0
- and even Porn 2.0
- these 2.0s refer to Web 2.0 technologies as the source of the new version in their respective disciplines and areas.

# Web 2.0 technologies

---

## ● browser side

- Asynchronous JavaScript and XML ([Ajax](#)),
- [RIA](#)
  - [Adobe Flash](#)
  - [JavaScript](#)/Ajax frameworks
    - [Prototype](#), [script.aculo.us](#), [Yahoo! UI Library](#), [Dojo Toolkit](#), [MooTools](#), [jQuery](#), [ExtJS](#), ...
  - others
    - XUL, JavaFX, Silverlight, OpenLaszlo, ...

## ● server side

- many of same technologies as Web 1.0
  - [PHP](#), [Ruby](#), [ColdFusion](#), [Perl](#), [Python](#), [JSP](#), [Servlet](#), and [ASP](#)
- addition with providing data in different format
  - [XML](#), [RSS](#), and [JSON](#) , why?

# Summary

---

- The Internet
  - history
  - key aspects
  - people and organizations
  - layered architecture
  - protocols: IP, TCP
- The World Wide Web (WWW)
  - servers and browsers
  - protocols: DNS, URL, HTTP, MIME
  - web languages / technologies
- Web 2.0
  - features, competences, applications, and technologies

# Reading materials

---

- [A Brief History of the Internet http://www.isoc.org/internet/history/brief.shtml](http://www.isoc.org/internet/history/brief.shtml)
- [http://en.wikipedia.org/wiki/Web\\_2.0](http://en.wikipedia.org/wiki/Web_2.0)
- <http://oreilly.com/web2/archive/what-is-web-20.html>

# Thank you!



# Exercises & Assignments

---

- use a terminal shell on your computer to fetch the homepage of our Software School of SYSU
- install the Chrome , using devtools
  - introduction: <https://developer.chrome.com/devtools>
  - tutorial: <http://discover-devtools.codeschool.com/>
- register a Github account for holding your homeworks
- learn using of Git
  - elementary: <https://try.github.io/levels/1/challenges/1>
  - advanced: <https://www.atlassian.com/git/tutorials/>
- learn using of Github
  - <https://guides.github.com/activities/hello-world/>