CS 316 - Web Programming

Mr	Raphael	Finkel
Dr	Rafi	Goldstein
Prf	ฟิงเคล	
—		

MultiLab

ssh / putty / . . .

abc123

ssh pen.cs.uky.edu
cor

→ login: abc123
Password:

History

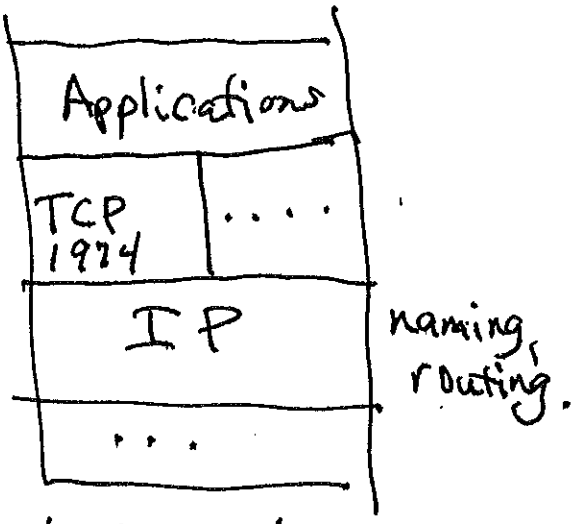
Arpa Net 1973

IP: Internet Protocol

IPv4: names every computer with a 32-bit identifier

128.163.146.21

Protocol stack



TCP: reliable, in-order transmission

Applications:

FTP: File Transfer 1971
SMTP: Simple Mail Transfer 1971
NTP: Network Time 1985
HTTP: Hypetext Transfer 1989
DNS: Domain Name System 1983
SSH: Secure Shell 1995

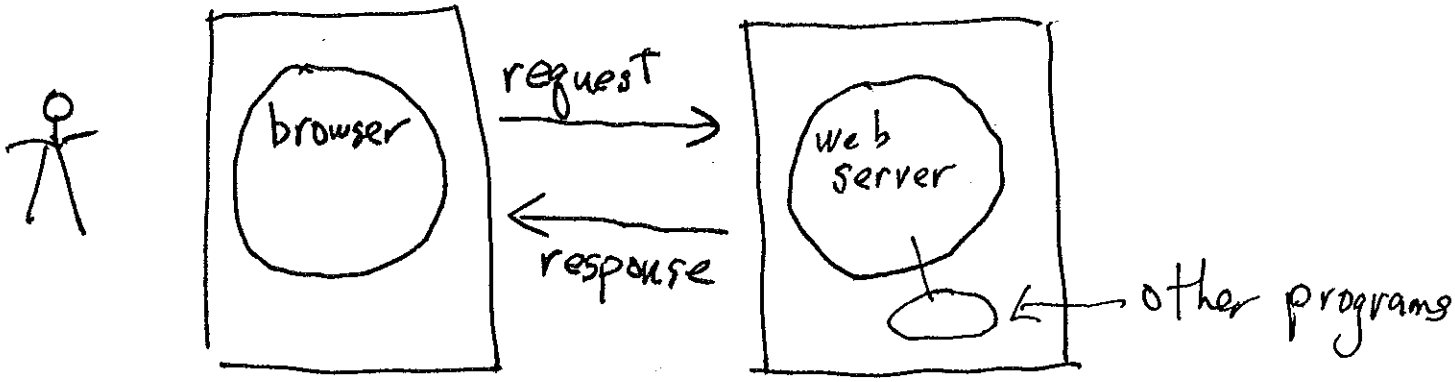
Internet Engineering Task Force

IETF

specifications: documents called RFCs

↓

Request for comment



client

Firefox
Chrome
Safari
IE

HTTP

server

apache 2
nginX
IIS
...

Each request is self-contained.

Server can ask the browser to store cookies (opaque identifiers)

Request asks for a page.

static — Typically, a page is a whole file.

dynamic — If the file is "special", it is a program; the server runs the program. Its output is the response.

Request names the file with a URL.
Universal Resource Locator

URL format:

protocol://hostName:port/file path?parameters

protocol: http
https

hostName: IP 128.163.146.21
DNS www.cs.uky.edu

file path: f1/f2/f3

parameters: var1=value1&var2=value2 ..

Unix permissions.

Owner	Group	World	
RWX	RWX	RWX	
6	6	4	files
110	110	100	
7	7	5	
111	111	101	

R: can see the file names in dir
W: can add/delete files
X: can "open" the directory
} directories

% chmod 664 foo.txt

% ssh abc123@pen.cs.uky.edu

% scp localfile abc123@pen.cs.uky.edu:HTML/foo

```
% cd ~
% mkdir HTML
% chmod 751 HTML
```

HTML Hypertext Markup Language

all browsers understand HTML.

derived from XML (Extensible Markup Language)

HTML 5 : 2014

HTML page

text with markup

Markup

tags: surrounded by brokets

ex: <head> (opening)

</head> (closing)

additional attributes

<tag attr='value' >

↑

separate multiple attributes with space

tagged regions may nest, but not overlap.

↓

subject to some constraints.

HTML tags

required regions

<html>
<head>
<title>
<body>

Comment: <!-- .... -->

inside <head>

<style>] regions
<script>]

<meta>] non-regions
<link>]

inside <body>

<h1> ... <h6>
<p>
<a> (anchor)

 (vertically separate region)
<pre> (pre-formatted)
<table>
<tr>
<td>

Exercise:

On the MultiLab, build a web page
called inClass-01.html
that satisfies the validator
and says: Hello world

---

http://www.cs.uky.edu/~abc123/inClass-01.htm

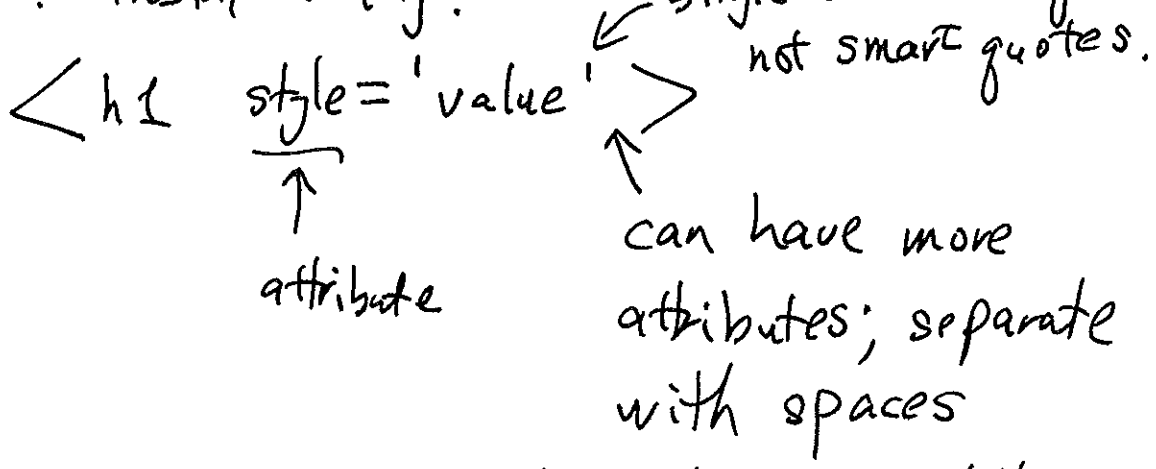
file: ~/HTML/inClass-01.html

at least 001 ↑ ↑ at least 004

% chmod ~~001~~ 711 ~/HTML

% chmod ~~644~~ 664 ~/HTML/inClass-01.html

attributes : inside a tag:



value has a syntax depending on attribute.

```
<img width='20'>
```

some attributes are obligatory

```
<img ~~scr~~ src='...'>
```

Some are recommended

```
<html lang='..'>
<img alt='...'>
```

common attributes

- id unique identifier for element
- style
- class regions that share formatting

unrecognized: ignored.

Styles

can be multiple words

syntax: property: value; property: value;

spaces are ok

text modifications

color red

#00FF00

r g b

rgb(0,255,0)

background-color

text-align: left, right, justify

vertical-align: top, middle, bottom

text-indent: 20px

font modifications

font-family Times New Roman

Serif

Sans Serif

Helvetica

font-style normal, italic, oblique

font-weight normal bold

Box model

Modify :

Hello world : blue

add a table

hello	there	world
how	are	you
feeling	today	?

blue

bdd.

CSS : Cascading Style Sheets

```
<style>
    p { color:green; }
    :
    :
</style>
```

selector

style information

Selectors :

- tag name — p, li
- class name — .inner, .outer, .foo
- id name — #main
- combinations
- operators — child, parent
- [attribute]

Rules that determine formatting

for each property (font-family, color, backgroundColor...) the browser uses the first value it finds:

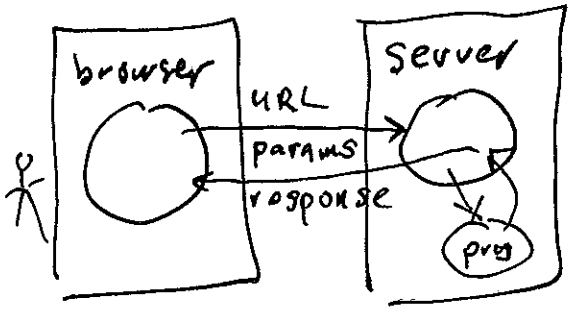
1) style="value" on the element's tag.
2) entry in <style> region that targets
   #id
   .class
   tag
3) repeat steps 1 and 2 for each surrounding element

However: can mark a style !important to override this search.
(sometimes ~~—~~ mystery

Common Gateway Interface (CGI)

requests from browser to server are generated:

1) loading a web page
2) submitting a form
3) AJAX



style='font-family: Free Serif, Serif, Arial'

value

audio: supported mp3, ogg, wav

↓

not in Safari

video: mp4, webM, ogg

CGI program

may write in any language

many lgs have CGI libraries.

Unix permissions

- all directories from the CGI program and up must be world-executable (001 bit)
- the file itself must be executable by the owner (100 bit) and not writable by the group or world
- the program runs with owner's permissions. (suggestion: 754)

File name: ends with .cgi or .php

start of file: #! name of exec.

Parameters come into the CGI program in the environment variables.

REQUEST_METHOD : GET or POST

GET finds its parameters in

QUERY_STRING

"p1:value1&p2:value2 ..."

POST :

CONTENT_LENGTH

how many bytes of parameters query string is in stdin

↑

standard input

POST is better

parameters are not visible

There are non-interactive web clients.

wget

curl

lynx (interactive, not graphical)

Languages for writing CGI

	params	errors	template
Perl	CGI	CGI::Carp	HTML::Template
Python	cgi	cgitb	Chameleon Flask...
C/C++	cgicc		ClearSilver
Java			jsp
PHP	built		built in
Ruby			ERB

PHP

Scripting lg based on templates.

Manual

Program is HTML decorated with

<?php code ?>

Hard to debug.

Many functions: files, gather data from forms, handle cookies, interact with databases, ...

To check syntax:

% php -l FILENAME

To get command-line parameters

% php FILE.php 'param1=v1&param2=v2'

Debugging CGI

1) Always test on command line
2) Error log on server
3) Some lgs have a way to reflect errors to browser.
4) #! /usr/bin/python3 

5) Test on MultiLab

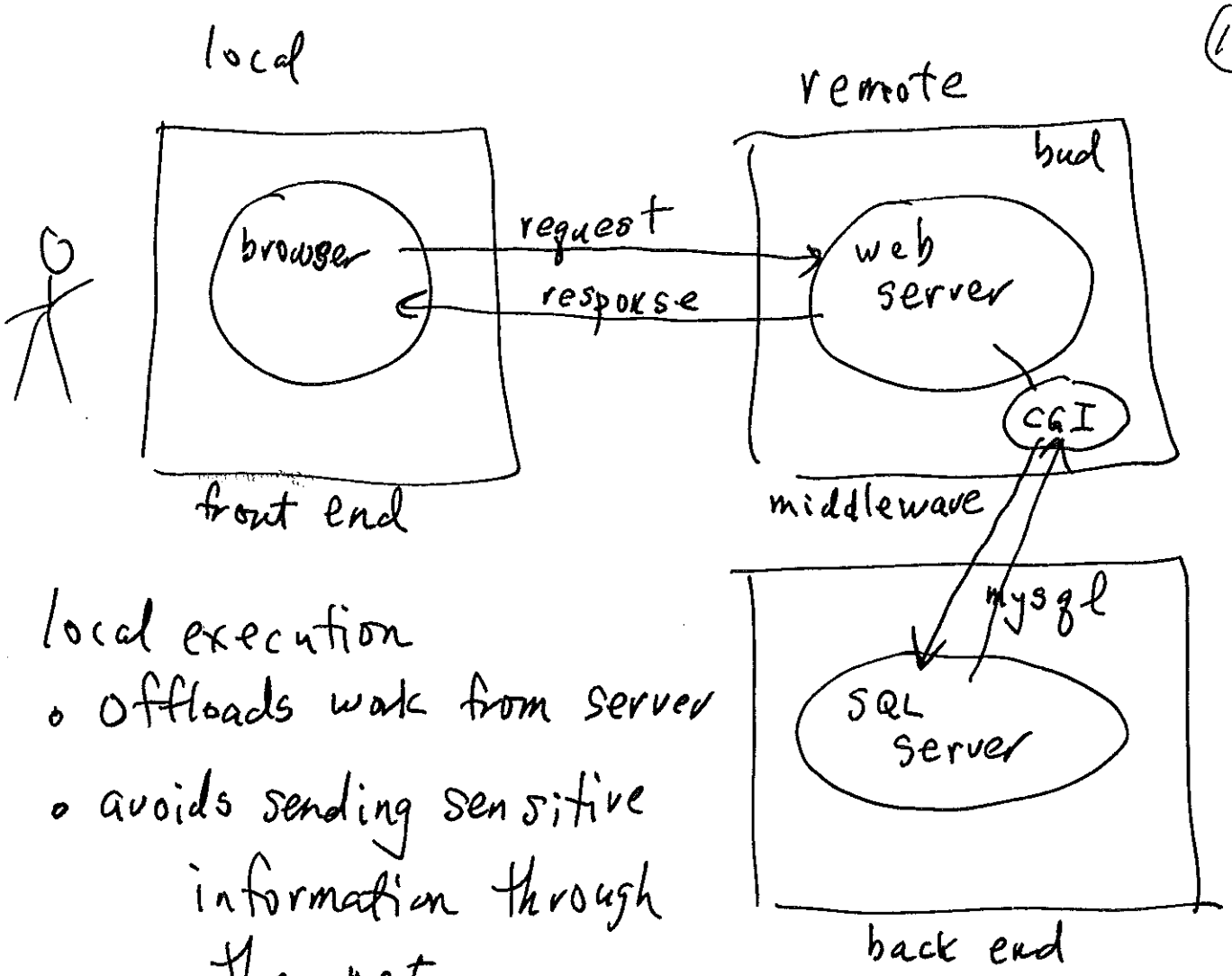
Exercise:

Modify example 8
including Saturday + Sunday
adding 1 more possible message

Local vs Remote execution

↓ in the browser

↓ in the web server



local execution

- offloads work from server
- avoids sending sensitive information through the net
- depends on local facilities: JavaScript

---

remote execution

- uses full power of server
- less portable to other servers.

JavaScript

ECMAScript 1997

High-level language

Just-in-time compiled.

Advanced features

dynamic typing

object orientation - prototype based

first-class functions

anonymous functions

Standalone version (not in a browser)

- Nodejs
- In developer tods : console

Error-prone

Difference

In browser: No access to files

No access to other programs.

Access to DOM

↓

Document Object Model

SQL ← Structured Query Language

Many DB servers:

- mySQL, mariaDB,
- SQL Server
- MS Access
- Oracle
- Sybase ....

} Relational DBs

- tables
  - rows (entries)
  - columns (fields)

Many languages have "a DB connector" module.

- Java: JDBC
- PHP: PDO
- Perl: DBI
- ⋮

} provide

1) connect to DB
2) make queries ↓ safely, avoiding insertion attacks.

To safely query:

s = "SELECT (?) from TABLE" ← prepare

The DOM ← Document Object Model

Hierarchy of objects in a web page
Predeclared in browser's Javascript.

Scripts inside HTML, invoked by HTML

<script>
⋮
</script>

inside <head> region.
(to define functions)

after </body>
before </html>

↘ for code to run after the page is loaded.

Call functions as a side-effect of an event.

By binding a function to an event.

1) onclick='javascript'

↑ onmouseenter ...

↑ function call, with a parameter this referring to the element

2) addEventListener(event, function)

↑ string

AJAX

Allows a web page to query its server, get a result and use that result to update the page, without reloading page.

On MultiLab:

in ~/HTML/cafe-2.php

	~	HTML	cafe-2.php
atleast	001	001	001
recom.	751	751	751

URL http://www.cs.uky.edu/~abcd123/cafe-2.php

% chmod 751 ~

The JavaScript constructs a request
Sends it to the server
Waits for a response
Manipulates the response to update page content.

Why?
Update page without reloading.
Complete a page content after loading.
Send data to server in background.

Efficiency + security

Every AJAX request causes
a round-trip request/response.
work on the server

Browser usually just waits, not using CPU.

Request must go to the same server
that provided the page that
is making the request.

---

fetch()

fetch(URL, { })
optional
↑
extra

GET: default
parameters: at end of URL
URL?p1=value1&p2=v2

POST: in extra

```
{ method: 'POST',
  body: 'p1=value1&...',
  headers: new Headers({
    'Content-Type':
    'application/x-www-form-urlencoded'})}
```

The response to a request has a header.

If the response is text:

Content-Type = text/html

If JSON

Content-Type = application/JSON

in PHP: header(...)

JQuery

Framework: a library to make it easier to build web pages.

Introduced in 2007

Uniformity across browsers

"polyfill" = workaround for a missing feature.

Easy selection of DOM elements.

Current version: 3.6.0 (3/2021)

Features

Avoids browser-specific things

Manipulation DOM, CSS.

Binding events to code

Handling effects + animations.

Simplifying AJAX

Enables lots of other software.

PHP program

```
<?php
    check what type of output.
    if (want HTML) {
?>
 <DOCTYPE ... >
   :
   :
  </html>
<?php
  } else {
    header ( ... )
    jString = " . . . . . ";
                ↖ requires computation

    echo (jString)
?>
```

REST interface

every request is self-contained.
The server has no retained knowledge of previous requests.

MIDTERM EXAM

1) Make sure a legible copy of your name is on page 1.
2) No computers or phones.

1a) Firefox is a web client.

b) https://www.cs.uky.edu/~abcd123/somewhere/file.php

DNS name of server:
www.cs.uky.edu

c) ~abcd123/HTML/somewhere/file.php

d) Permissions 664

e) <tr> ... <form> ... </form> </tr>
invalid HTML

2) a) html title head body
!DOCTYPE

b) <img ~~scr~~ src = "_____"
alt = "_____" >

c) <span style = "border: ~~1px~~ .
1px solid black" >

d)
<style>
#howdy { font-family : Arial; }
</style>

e) why "integrity" attribute?
To authenticate the ~~validy~~ validity of the received data.

3) a) select 2nd <pre>
document.querySelectorAll("pre")[1]
getElementsByTag

b) myElt.style.color = 'red';

c) all <p> elements with class myClass.
document.querySelectorAll("p.myClass")
"p[class = myClass]"

4) a) In <style> border all

```
li { border: 6px dotted green; }
```

b)

```
.myClass {
    background-color : lightblue;
    font-weight : bold;
}
```

c)

```
$(".myClass").css('backgroundColor', 'lightblue').
  css('fontWeight', 'bold');
```

```
$(".myClass").css({
    backgroundColor : 'lightblue',
    fontWeight : 'bold'
})
```

d)

```
$(".myClass").mouseenter(function (){
    $(this).css('fontSize', '30px');

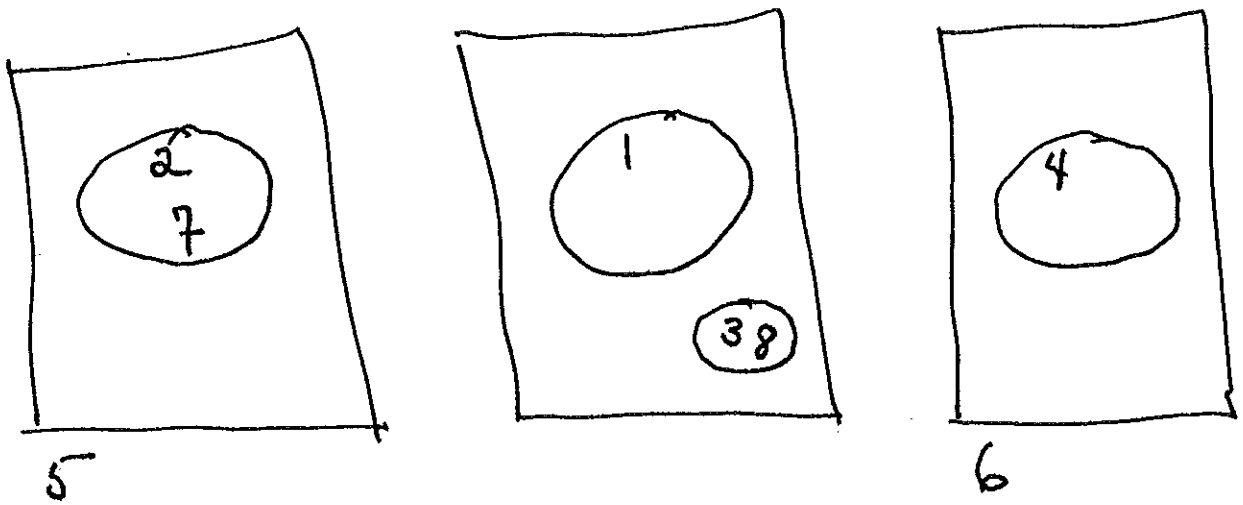
});
```

e)

```
$(".myClass").mouseenter(function () {
  $(this).animate({'fontsize':'30px'}, 2500)

});
```

a)



b) mytable

```
SELECT * FROM mytable;
```

c) Why AJAX?

Allow JavaScript to manipulate response to modify the existing web page without reloading it.
Asynchronously.

d)

```
fetch("echo.cgi?parameter1=Hello world
```

Fonts

A web page may specify its own fonts.
In CSS:

```
@font-face {
    font-family: anyName;
    src: url( somewhere )
         url( ... )
}
```

font-style
font-weight
unicode-range

↑ additional formats

Formats:
woff woff2 ttf otf svg ...

Where are fonts?

fonts.google.com

To build your own icon font:

icomoon.io/app/

download font file (several formats)
CSS file

to use:

More CSS

viewport (responsive web design)

↑ responds to physical limits of the browser.

@media (in <style>)

spellcheck / autocomplete / ...

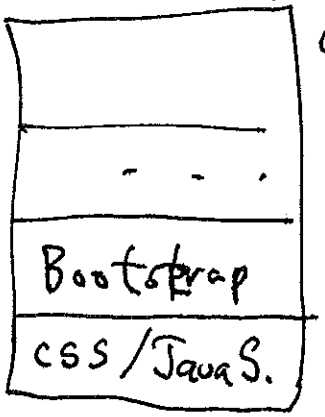
Bootstrap V3 V4 V5

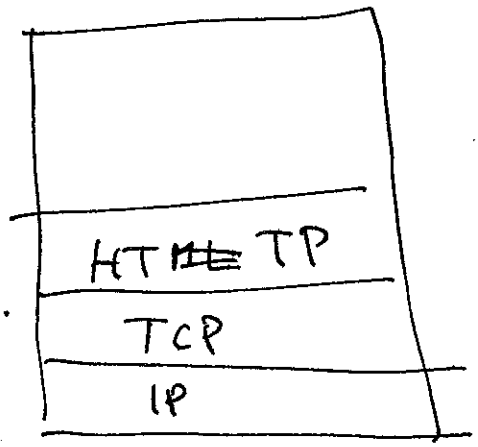
↑ I like ↖ we will see

A library · CSS
JavaScript

V4 needs jQuery
V5 doesn't

A framework: Software upon which you can build code.





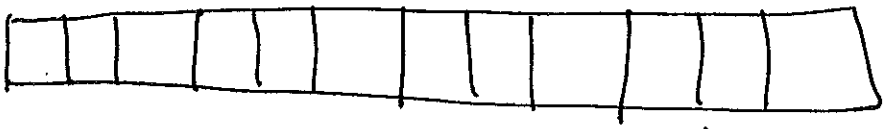
Main features

Design templates for standard elements.

Responsive web design

The grid system

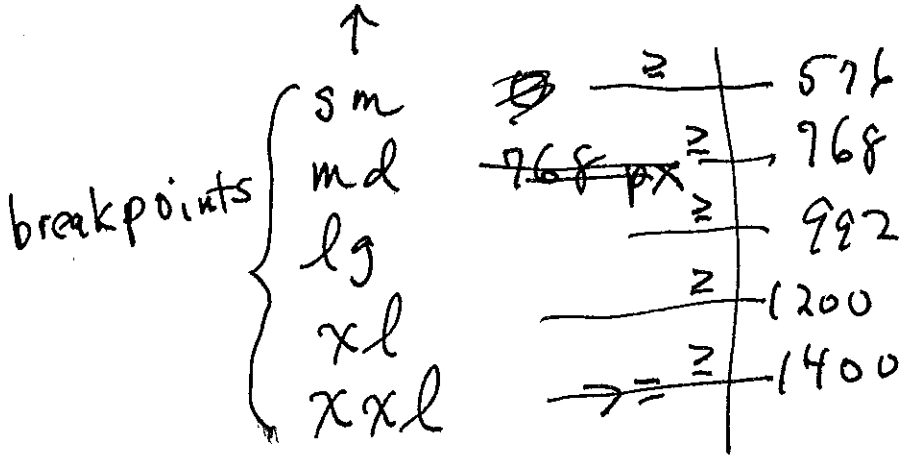
1) regions are understood to be 12 cells wide. (rows)



2) nested regions (~~"rows"~~ "columns")

can use a specified number of cells.

3) The specification can depend on the width of the screen.



col-md-5 ← use 5 cells

↑ a column ↑ at least 768px

offset-sm-2

col-5

To insert content dynamically into a web page.

1) HTML / JavaScript provides

```
document.createElement ( . . . )
document.appendChild (parent, child)
```

2) jQuery has similar methods.

3) Build an HTML string with the new content.

```
parent.innerHTML = " that string ".
```

4) Have the new content already in place,
with style: display: none
visibility: hidden
to make it appear:

```
node.style.display = "block"
                   .visibility = "visible"
$(node).show();
```

DataTables

Plugin for jQuery

↖ relies on

Builds nice, searchable, scrollable tables

May use Bootstrap

May options. (there is a manual)

My example (22) also shows

favicon

shaking

```
<table>
    <thead>
        <tr> <th> col. header </th> ... </tr>
    </thead>
    <tbody>
        ignored
    </tbody>
</table>
```

```
$(...).DataTable({
    options
});
```

Exercise:

Build a web page based on

www.cs.uky.edu/~raphael/courses/CS316/examples/example.22.html

/services.cgi

<ctrl-u> (in browser) shows source
<ctrl-a> selects everything
<ctrl-c> copies the selection into clipboard

TypeScript

JavaScript with declarations and types.
+ prevent programming errors
− browsers don't have it.

features:

can infer the type of an expression
Knows about the DOM
introduce programmer-defined types.

Indexed DB

Data are in the browser, accessed by JavaScript.

Same-origin policy enforced by browser.

Data survive browser restarts.

Not organized as "relational tables"

Operations are packaged in "transactions"

failure atomicity

synchronization atomicity

Operations are asynchronous

Bind completion events to functions.

HTML canvas

#pixels

<canvas id="~" width=NN height=NN>

</canvas>

JavaScript fills in the content.

awkward.

Plugin for jQuery called jCanvas.

less awkward.

SVG : Scalable vector graphics

like Canvas, it's in the HTML.

but : does not require JavaScript

scalable { <svg> ← contents are part of DOM </svg>

far more complex than canvas.

usually, people generate pictures for svg with an editor.

REST : Representational State Transfer

(2000 Roy Fielding)

RESTful interfaces

Characteristics

1) Client-server architecture
   Independently programmed.
   following a protocol.
2) Statelessness : each request from client
   contains all data needed to fulfil the
   request.
3) Cacheability : client may store responses.
   Server should indicate whether cache is OK.

Security worries for web programmers.

CGI vulnerabilities

CGI program must sanitize inputs

Cannot depend on client to do that.

Because: curl / wget / developer console

If data are sent via GET, they are part of the URL.

Code-injection attacks.

parameters to ~~f~~ calls can include program fragments.

SQL example

HTML example:

string ... <script anything></script> ..

refer off-site to any JavaScript

Cross-site scripting attacks (XSS)

1) Stored XSS: application stores + retrieves <script> regions.

2) Reflected XSS: <script> embedded in a URL as a GET parameter.

< can be represented as %3C or <

3) DOM-based XSS

JavaScript invokes eval() on data in an input field

to sanitize data:

if data become part of page,
remove < or use .innerText

string.replace(/</g, "");

if to shell:
remove shell metacharacters
; & ( ) > < ' "

if to sql:
use prepared statements or
remove: ' " \ ; ( )

Social engineering (usually email)

phishing: getting private info.

spear phishing: aimed at specific individuals.

scam: getting banking info on promise reward.

spam: unwanted advertisement.

Remote attacks

Buffer overflow: invalid data sent to a server that causes server to do wrong thing.

Brute-force attacks: repeatly connecting to a service to guess passwords.

Denial of Service (DOS): overload a server by repeated (expensive) requests. → 39

React

Framework for building web sites.

2013: Jordan Walke (Facebook)

alternatives:

- bare JavaScript
- jQuery + plugins (DataTables, jCanvas)
- Angular (Google)
- Vue.js

React Native: for building apps apple/android.

Uses JSX heavily

(JavaScript XML)

an expression in JSX can be HTML.

const myString = <input ... >

requires "babel" plugin.

React renders components

↘ places on the page.

DDOS: Distributed denial of service
a coordinated group of clients sending requests to a server.

Defense:

https instead of http.

authentication: client know the identity of server
integrity: no intermediate can modify traffic without detection.
privacy: no intermediate can read data.

passwords

strong
not stored: digest instead.