

View Calendar

Calendar Setup
Apply to Manager
View Calendar
Add Block(s)
Block in Appointment
Search appointments

Help
FAQ
What's New

Before people can make appointments with you, you must add one or more blocks of available time to the calendar. Click Add Block(s) to add blocks, or click Help for details.
[close message](#)

My new calendar

< October 2014 >

Day Week Month

SUN MON TUE WED THU FRI SAT

Sun	Mon	Tue	Wed	Thu	Fri	Sat
			1	2	3	4
			Teaching	Teaching	Teaching	Teaching
5	6	7	8	9	10	11
Teaching	Teaching	Teaching	Teaching	Teaching	Teaching	Teaching
12	13	14	15	16	17	18

October 2014

November 2014

Project Deliverables!

Design Document Template

www.klarity.com

© 2014 Klarity, Inc. All rights reserved.
This document is the property of Klarity, Inc.
It is to be used for the development of
software and other business-related
products.

KLARITY.COM

Advanced Programming Techniques

Web Technologies 1

Christopher Moretti

World Wide Web

- ❖ Network of interlinked hypertext data
- ❖ Developed at CERN '89-'91
 - ❖ Tim Berners-Lee proposed project, wrote HTTP, HTML, the first server, the first browser, and the first pages
 - ❖ Used internally, spread to Stanford '91, by end of '93 there were 623 websites!



URLs/URIs

- ❖ Uniform Resource Locator / Identifier

- ❖ Defined in RFC1738 in 1994 by Berners-Lee

- ❖ Built off of domain name and file path semantics

scheme://user:password@host:port/path?query#fragment

- ❖ Many of the segments are optional





1

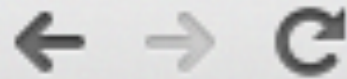
2

3

4

5

6



<http://blog.woorank.com/seo-basics/url-optimization-tips>

- 1 - Protocol
- 2 - Sub-Domain
- 3 - Domain
- 4 - Top-Level Domain
- 5 - Folder/Path
- 6 - Page

HTTP: HyperText Transfer Protocol

```
$ telnet www.google.com 80
Trying 74.125.141.104...
Connected to www.google.com.
Escape character is '^]'.
GET /index.html HTTP/1.1
```

```
HTTP/1.1 200 OK
```

```
Date: Tue, 01 Mar 2016 01:59:02 GMT
Expires: -1
Cache-Control: private, max-age=0
Content-Type: text/html; charset=ISO-8859-1
P3P: CP="This is not a P3P policy! See https://www.google.com/support/accounts/answer/151657?hl=en for more info."
Server: gws
X-XSS-Protection: 1; mode=block
X-Frame-Options: SAMEORIGIN
Set-Cookie: NID=77=qg6VKm-LUQxyY9WBGE80ckifoh0aqdk6-
aW86_KqyHrveUh7NGXhtLgBXwgE1VmA8PbkcAFa7kqQMScidf_vlfuLeRNMGoRvhVFAYiIBYMEXBI_QUQdaM9x388eQFmeZYt9fb1WuAUBQyA;
expires=Wed, 31-Aug-2016 01:59:02 GMT; path=/; domain=.google.com; HttpOnly
Accept-Ranges: none
Vary: Accept-Encoding
Transfer-Encoding: chunked
```

```
8000
```

```
<!doctype html><html itemscope="" itemtype="http://schema.org/WebPage" lang="en"><head><meta content="Search the world's
information, including webpages, images, videos and more. Google has many special features to help you find exactly what
you're looking for." name="description"><meta content="noodp" name="robots"><meta content="text/html; charset=UTF-8"
http-equiv="Content-Type"><meta content="/logos/doodles/2016/leap-year-2016-5690429188079616-hp.gif"
itemprop="image"><meta content="It's the 29th of February! #leapyear #GoogleDoodle" property="og:description"><meta
content="http://www.google.com/logos/doodles/2016/leap-year-2016-5690429188079616.4-thp.png" property="og:image"><meta
content="550" property="og:image:width"><meta content="220" property="og:image:height"><title>Google</title><script>
```

HTTP: HyperText Transfer Protocol

```
$curl -v www.google.com//logos/doodles/2016/leap-year-2016-5690429188079616-hp.gif > doodle.gif
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload  Total      Spent    Left     Speed
  0     0    0     0     0     0     0     0  ---:---:--  ---:---:--  ---:---:--    0*   Trying 173.194.123.16...
* Connected to www.google.com (173.194.123.16) port 80 (#0)
> GET //logos/doodles/2016/leap-year-2016-5690429188079616-hp.gif HTTP/1.1
> Host: www.google.com
> User-Agent: curl/7.43.0
> Accept: */*
>
< HTTP/1.1 200 OK
< Content-Type: image/gif
< Date: Tue, 01 Mar 2016 02:09:50 GMT
< Expires: Wed, 01 Mar 2017 02:09:50 GMT
< Cache-Control: public, max-age=31536000
< Last-Modified: Thu, 18 Feb 2016 23:56:07 GMT
< X-Content-Type-Options: nosniff
< Server: sffe
< Content-Length: 304891
< X-XSS-Protection: 1; mode=block
<
{ [2524 bytes data]
100 297k 100 297k    0     0 1732k    0  ---:---:--  ---:---:--  ---:---:-- 1741k
* Connection #0 to host www.google.com left intact
```




Not always that complicated ... or cute

```
$telnet localhost 8080
Trying ::1...
telnet: connect to address ::1: Connection refused
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
GET /cos/333
```

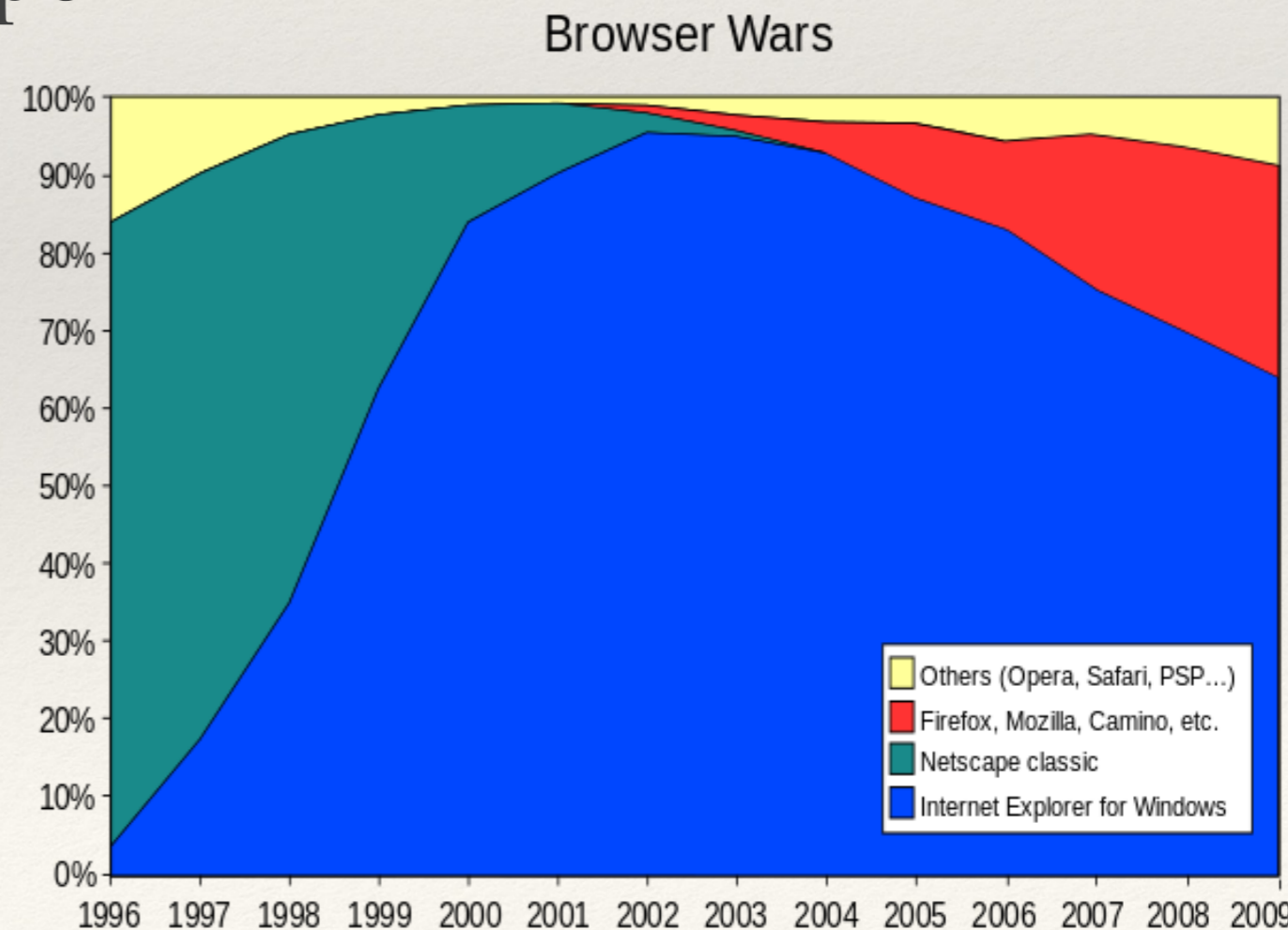
COS 333 TTh 11:00–12:20 Advanced Programming Techniques Christopher
M. Moretti Friend Center 101

HTML

- ❖ plain text description of content and markup for a page's structure
- ❖ interpreted by a browser
 - ❖ browsers can still differ in how they interpret HTML, but slowly standardizing
- ❖ tags with attributes bracket content (very incomplete set):
 - ❖ `<html><title>...</title><body>...</body></html>`
 - ❖ `<h1>...</h1> <h2>...</h2> <p> emphasis </p>`
 - ❖ `......`
 - ❖ `link to Google`
 - ❖ ``
 - ❖ `<table ... > ... </table>`
 - ❖ `<script> alert("hello"); </script>`

A Brief History of HTML

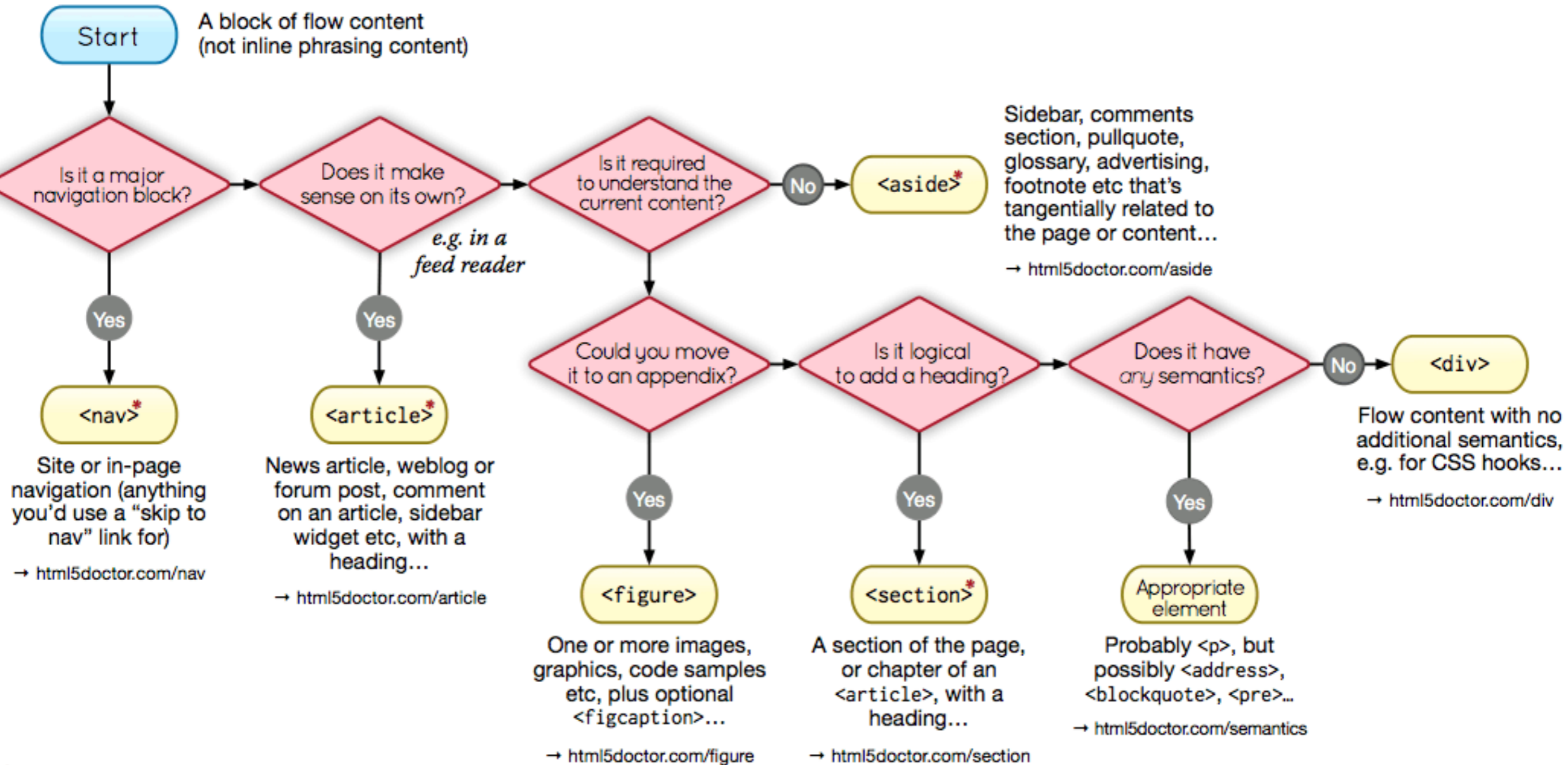
- ❖ Originally strongly influenced by SGML (1986 standard)
- ❖ 1993: NCSA Mosaic browser released
- ❖ 1994: HTML2 spec; Netscape
- ❖ 1995: W3C, HTML3 spec; Internet Explorer
 - ❖ Thus began the “browser wars”
- ❖ 1999 / 2001: IE5 / IE6



The Good-then-Bad Old Days

- ❖ In the beginning* the Web was purely semantic. As early as 1995, there was pushback about style-focused HTML mixing content and design, losing structure.
- ❖ Feature expansion, rampant incompatibilities, WYSIWYG editors exacerbated this
 - ❖ Every page is a table!
- ❖ SemanticHTML: HTML4.0, 4.1 Strict, 5 deprecate most “presentational markup” in favor of CSS for design and presentation, HTML for structure and content.
(So no ``, `<u>`, `<i>`, ``, etc.)

* 1992: <http://bit.ly/1QiGeJM>



* Sectioning content element
 These four elements (and their headings) are used by HTML5's outlining algorithm to make the document's outline
 → html5doctor.com/outline

So where DOES presentation go?

- ❖ Cascading Style Sheets (CSS): a language for describing the presentation of a markup language document
- ❖ Control color, size, alignment, position, padding, border
- ❖ Defined in a separate .css file (best), a `style` tag in an HTML document, or a `style` attribute in a tag (worst)
 - ❖ `<link rel="stylesheet" href="style.css" />`
 - ❖ `<style type="text/css" media="all">`
 `body {background: #fff; color:#000; }`
 `</style>`
 - ❖ `<p style="color:red;">`

CSS Syntax

- ❖ `optional-selector {prop:val; prop:val; ...}`
- ❖ selectors:
 - ❖ tag `h1 {color: red;}`
 - ❖ .class `.big {font-size: 200%;}`
 - ❖ #id `#first {padding-top: 10px;}`
 - ❖ :pseudo-class `a:hover {color: DeepPink;}`
`q:lang(fr) {quotes: "«" "»";}`


```
<html>
<style>
  body {background-color: #bbbbbb }
  p { text-align: left; color: #000088; background-color: #aaaaaa }
  h1, h2 {text-align: center; color: #0000aa; background-color: #888888 }
  ul { color: #008800; background-color: #bbbbbb }
  .first { padding-top: 20px}
  .last { padding-bottom: 20px }
  #special {color: #880000 }
</style>
<body>
  <h1>Primary Header</h1>
  <p class="first">This is the first paragraph.</p>
  <p>This is just another paragr
  <h2>Subheader</h2>
  <p id="special">This is a spec
  <ul>
    <li class="first">First item
    <li>Another item in the list
    <li>Yet another item in the
    <li class="last">Last item i
  </ul>
  <p>This is just another paragr
  <p class="last">This is the la
</body>
</html>
```

Primary Header

This is the first paragraph.

This is just another paragraph.

Subheader

This is a special paragraph.

- First item in the list.
- Another item in the list.
- Yet another item in the list.
- Last item in the list

This is just another paragraph.

This is the last paragraph.

Zen Garden

The Beauty of CSS Design

A demonstration of what can be accomplished visually through CSS-based design. Select any style sheet from the list to load it into this page.

Download the sample [html file](#) and [css file](#)

of the existing designs in the list. Clicking on any one will load the style sheet into this very page. The code remains the same, the only thing that has changed is the external .css file. Yes, really.

CSS allows complete and total control over the style of a hypertext document. The only way this can be illustrated in a way that gets people excited is by demonstrating what it can truly be, once the reins are placed in the hands of those able to create beauty from structure. To date, most examples of neat tricks and hacks have been demonstrated by structurists and coders. Designers have yet to make their mark. This needs to change.

The Road to Enlightenment

Littering a dark and dreary road lay the past relics of browser-specific tags, incompatible DOMs, and broken CSS support.

Today, we must clear the mind of past practices. Web enlightenment has been achieved thanks to the tireless efforts of folk like the W3C, WaSP and the major browser creators.

The css Zen Garden invites you to relax and meditate on the important lessons of the masters. Begin to see with clarity. Learn to use the (yet to be) time-honored techniques in new and invigorating fashion. Become one with the web.

So What is This About?

There is clearly a need for CSS to be taken seriously by graphic artists. The Zen Garden aims to excite, inspire, and encourage participation. To begin, view some



select a de

Under the S
Stoltz

Make 'em Pr
Michael McA
Scotty Reifs

Orchid Beau
Kevin Addis

Oceanscape
Gray

CSS Co., Ltd
Benjamin Kl

Sakura by T
Uchida

Kyoto Fores
Politowski

A Walk in th
by Simon Va
Hauwermeir

archives:

next design

[View All Des](#)

Dynamic Stylesheets



The dynamic stylesheet language.

LESS extends CSS with dynamic behavior such as variables, mixins, operations and functions.

LESS runs on both the server-side (with Node.js and Rhino) or client-side (modern browsers only).



Download [less.js](#)

[version 1.3.3](#)
[changelog](#)

Write some LESS:

```
@base: #f938ab;

.box-shadow(@style, @c) when (iscolor(@c)) {
  box-shadow: @style @c;
  -webkit-box-shadow: @style @c;
  -moz-box-shadow: @style @c;
}

.box-shadow(@style, @alpha: 50%) when (isnumber(@alpha)) {
  .box-shadow(@style, rgba(0, 0, 0, @alpha));
}

.box {
  color: saturate(@base, 5%);
  border-color: lighten(@base, 30%);
  div { .box-shadow(0 0 5px, 30%) }
}
```

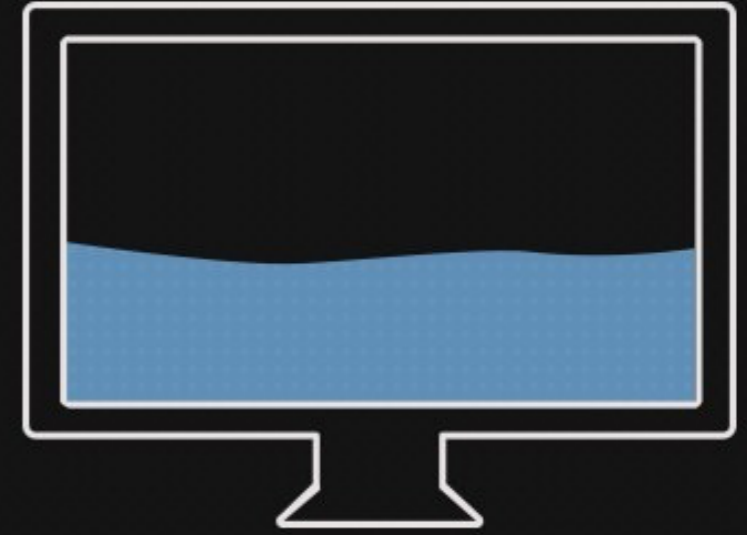
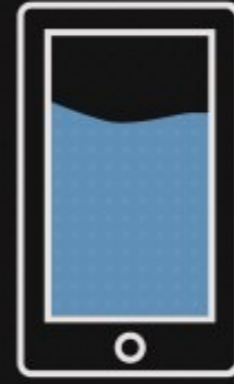
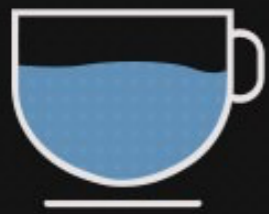
Compile to CSS:

```
npm install -g less
lessc styles.less styles.css
```

Responsive Design

- ❖ Fluid grid with proportional / relative units
- ❖ “Mobile First”
- ❖ “Progressive Enhancement”
 - ❖ Instead of “Graceful Degradation”
- ❖ Testing:
 - ❖ Ff Responsive Design Mode <http://mzl.la/1XXzYtT>
 - ❖ Enterprise: Browserstack, Adobe Edge Inspect, Ghostlab

CONTENT IS LIKE WATER



“ You put water into a cup it becomes the cup.
You put water into a bottle it becomes the bottle.
You put it in a teapot, it becomes the teapot. ”

Josh Clark (originally Bruce Lee) - Seven deadly mobile myths

Illustration by Stéphanie Walter

Introducing Bootstrap.

Need reasons to love Bootstrap? Look no further.



By nerds, for nerds.

Built at Twitter by [@mdo](#) and [@fat](#), Bootstrap utilizes [LESS CSS](#), is compiled via [Node](#), and is managed through [GitHub](#) to help nerds do awesome stuff on the web.

Made for everyone.

Bootstrap was made to not only look and behave great in the latest desktop browsers (as well as IE7!), but in tablet and smartphone browsers via [responsive CSS](#) as well.

Packed with features.

A 12-column responsive [grid](#), dozens of components, [JavaScript plugins](#), typography, form controls, and even a [web-based Customizer](#) to make Bootstrap your own.

Dynamic Content

- ❖ How can we make web pages more dynamic, interactive, responsive, low-latency, etc.?
 - ❖ Javascript
 - ❖ AJAX
 - ❖ Plugins (e.g. Flash, Quicktime, Silverlight, Applets)

Javascript

- ❖ Client-side scripting language (Brendan Eich, Netscape)
- ❖ Weakly typed: double, string, array, object types
- ❖ Object-Oriented (but not class-based)
- ❖ Loaded into HTML content with `<script>` tags
 - ❖ Also event-driven, e.g. `onClick` attributes.
- ❖ Dynamic interaction with DOM; can change page without a full redraw

* Technically Javascript is just one implementation of the ECMAScript standard

Language Basics

- ❖ Constants:
 - ❖ double literals, true, false, null, 'string literals'
- ❖ Variables
 - ❖ declaration is optional. types are unspecified
 - ❖ only 2 scopes: local to this function or global.
- ❖ Operators are like C, except ...
 - ❖ /regexp/ pattern matching
 - ❖ === and !==

Language Basics (cont'd)

- ❖ `if else switch`
- ❖ `while do-while break`
- ❖ `for (; ;) for (var in object)`
- ❖ `try { } catch () { } finally { }`
- ❖ `function name(arg1, arg2) { ... }`
 - ❖ anonymous functions:
`function (arg1, arg2 ...) { ... }`
 - ❖ functions are objects so `this` doesn't do what you think

Javascript Objects and Arrays

- ❖ Object is a compound type with named components
 - ❖ More like associative array / C struct than a Java class
 - ❖ Accessed as `obj.name` or `obj['name']`
 - ❖ mutable:

```
var point = {x:0, y:0, name: "origin"};
point.x = 1; point["y"] = 2; point.name = "(1,2)"
```
- ❖ Arrays are basically objects with numbered components

```
var mixedarr = ["what", 15, {t:"", h:"", e:""}, point ]
```

That notation looks familiar ...

- ❖ JSON - Javascript Object Notation - Douglas Crockford
 - ❖ Two structures: Arrays, Objects as in previous slide
 - ❖ Makes a lightweight data interchange format
 - ❖ Simpler, cleaner than XML, but also not as structured
 - ❖ Widely adopted into other languages
 - ❖ Python json module in A4

JavaScript: the bad and the ugly

- ❖ “In JavaScript, there is a beautiful, elegant, highly expressive language that is buried under a steaming pile of good intentions and blunders.”
- ❖ `==` vs `===` , `!=` vs `!==`
- ❖ optional semicolons, declarations
- ❖ global variables everywhere
- ❖ oddball scope rules
- ❖ + overloading



Thursday: Server Side



Node.js is a platform built on **Chrome's JavaScript runtime** for easily building fast, scalable network applications. Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient, perfect for data-intensive real-time applications that run across distributed devices.

<http://www.w3schools.com/js/default.asp>

<http://www.cs.princeton.edu/courses/archive/spr11/cos333/reading/javascriptsummary.pdf>