Web Design
CSIS-180

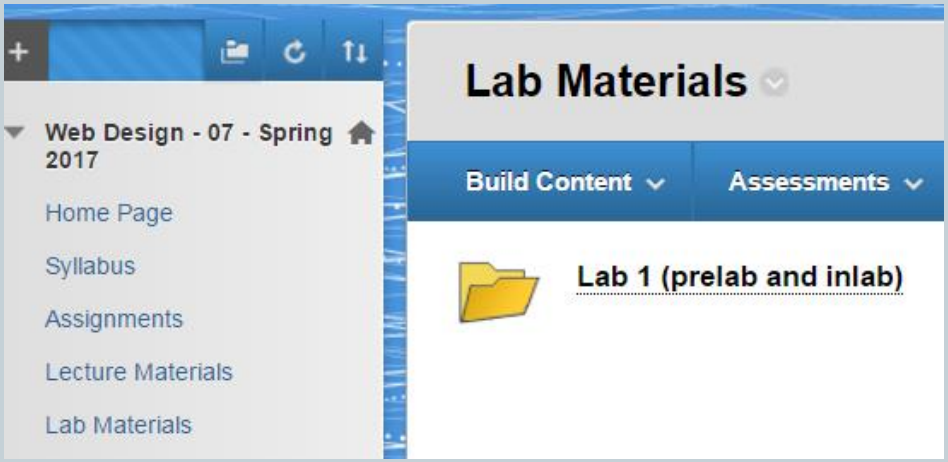
SIENA COLLEGE
DEPARTMENT OF COMPUTER SCIENCE

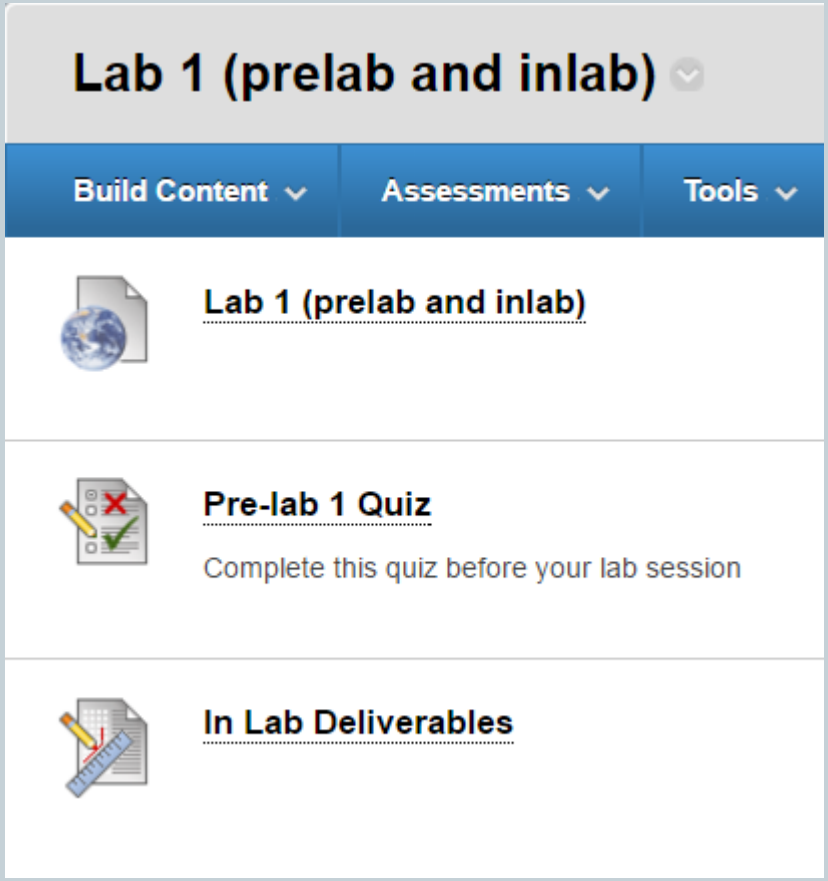
DR. JAMI COTLER
SPRING 2017

Today's Class

- Reset
- Group Syllabus Quiz
- Usability Scavenger Hunt
- Lab Discussion
- Chapters one and two

Labs


Web Design - 07 - Spring 2017
Home Page
Syllabus
Assignments
Lecture Materials
Lab Materials
Lab Materials
Build Content
Assessments
Lab 1 (prelab and inlab)


Lab 1 (prelab and inlab)
Build Content
Assessments
Tools
Lab 1 (prelab and inlab)
Pre-lab 1 Quiz
Complete this quiz before your lab session
In Lab Deliverables

Chapter One
Where do I Start?

CSIS-180
WEB DESIGN

What is Web Design

Encompasses several different areas:

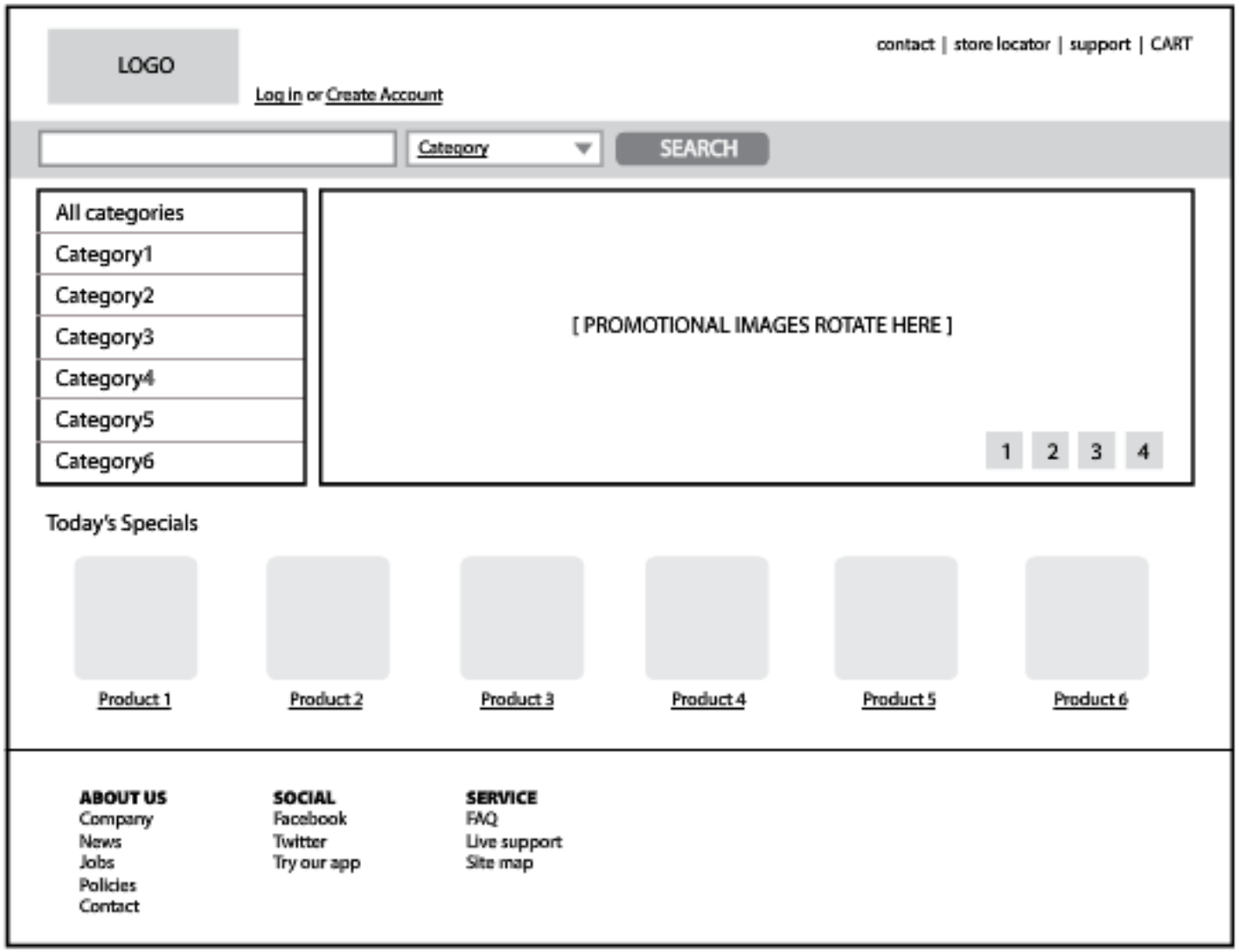
- Graphic Design
- User Interface Design
- Document Design
  - Once called Desktop Publishing
- Content Design
  - Content Strategy
  - Content Management
- Multimedia
- Application Development
  - Programming & Scripting

Designing for Users

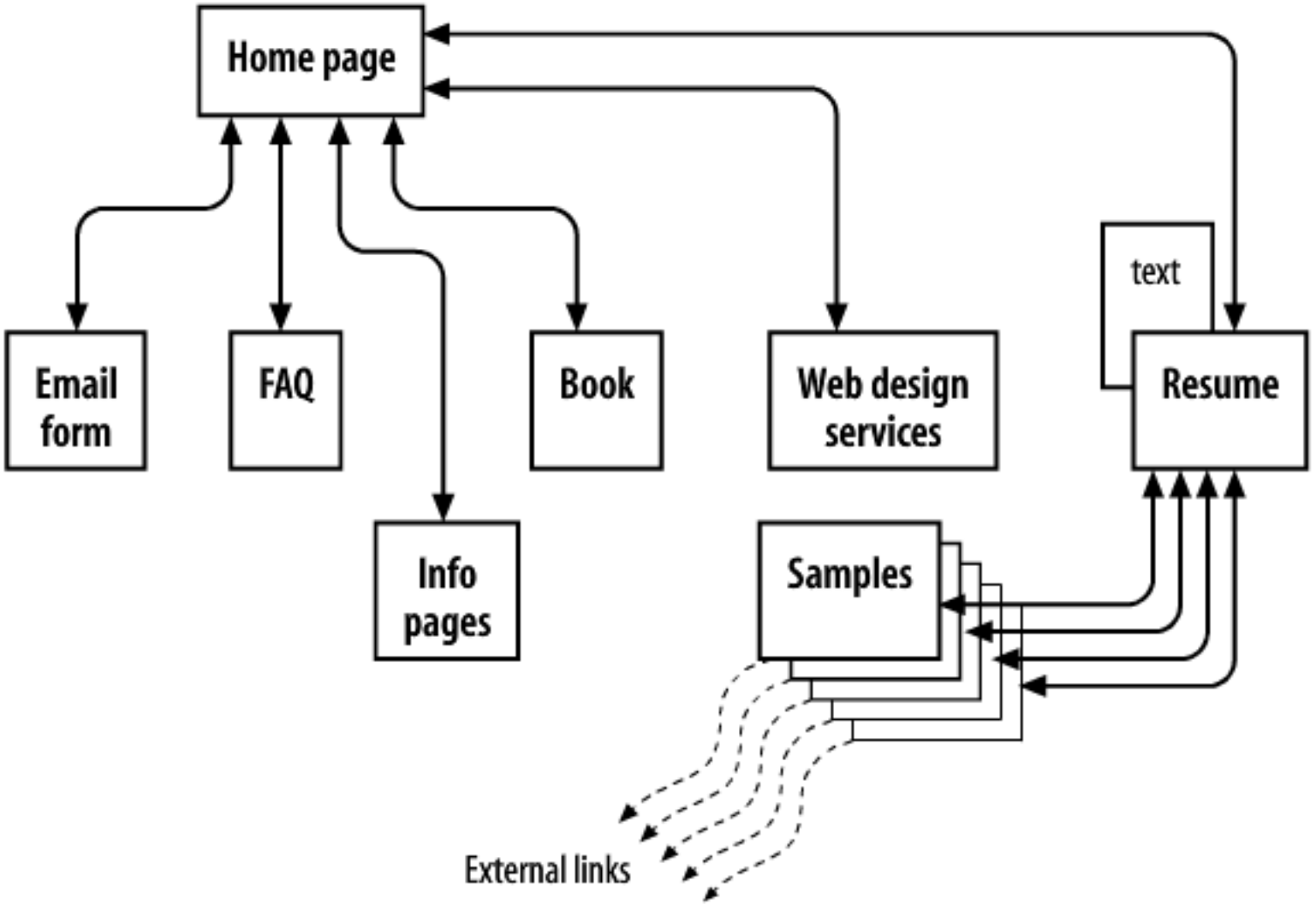
- Not just how it **looks**, but
  - how it works and
  - how it “feels” for users

- Key Areas:
  - Interaction Design (IxD)
  - User Interface (UI) design, and
  - User Experience (UX) design.

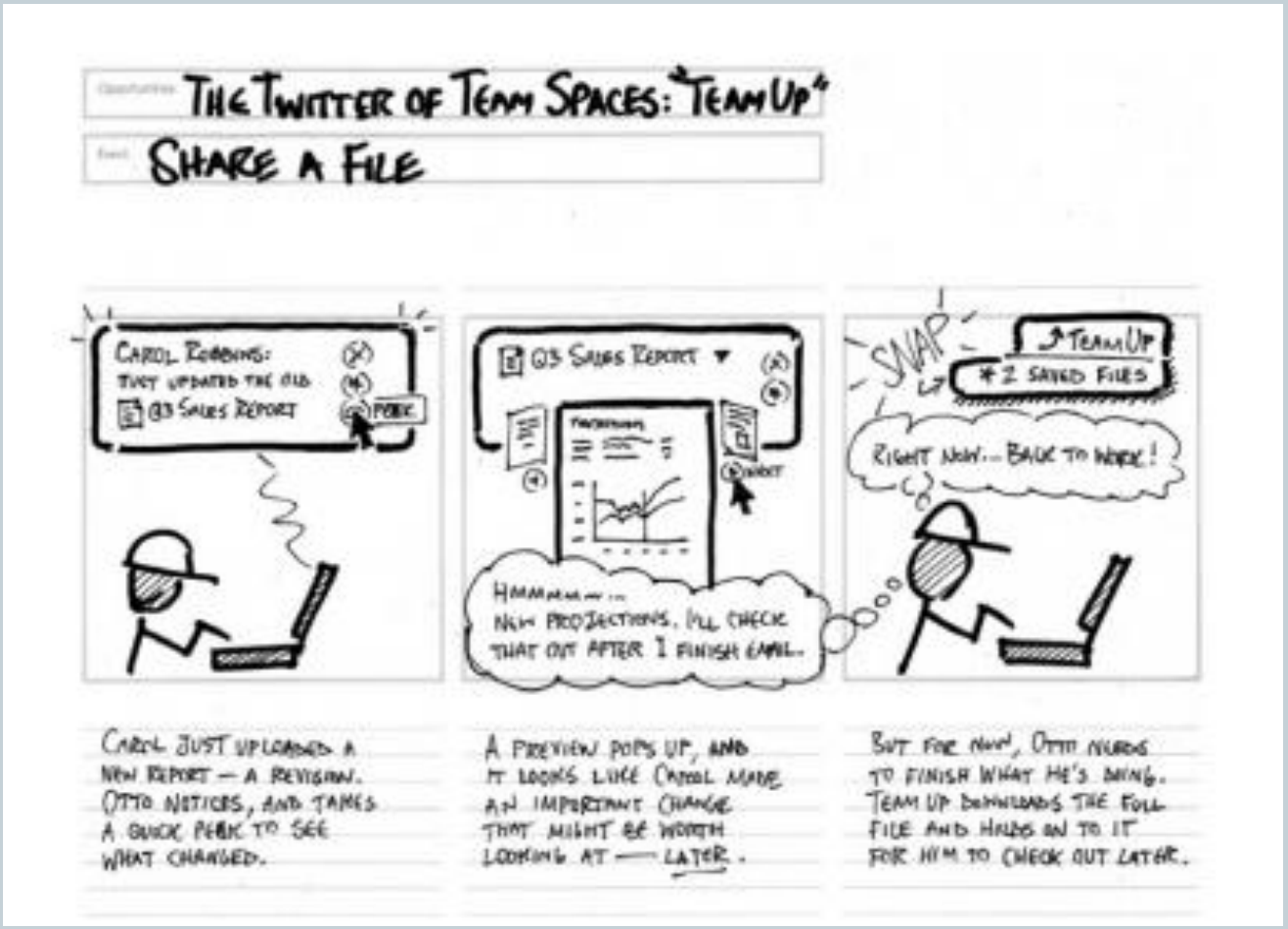
Design Examples: Wireframes



Design Examples: Site Diagram



Storyboards and user flow charts



Visual/ Graphic Design for the Web

- Traditional Areas:
  - logos, graphics, type, colors, layout, etc.
  - Ensure consistency with the brand/message of the organization
- Web Areas
  - Layout and position on a web page
  - Creating graphics optimized for web delivery
- Samples:
  - Resume
  - Visual Images on the Web (Penn State)

Visual/ Graphic Design for the Web

Key Technologies:

- Image capture
  - Digital photography
  - Image scanning
- Graphic formats
  - GIF, SVG, JPG, PNG

Visual Web Language:

- CSS: Cascading Stylesheet

Key Software:

- Photo Editing
  - GIMP
  - Adobe Photoshop
- Digital Image Creation:
  - Abobe Illustrator
  - Adobe Fireworks

Web Content Management & Strategy

Organizing page content

- Layout strategy
- Page template design
- Semantic markup
  - Meta Data
  - Meaningful Document Structure

Organizing website content

- Logical site organization
  - Hierarchy of pages
- Site template design
- Navigation design

Core Web Languages

What languages do I need to learn?

1. Content Structure (**HTML**)
   {markup language}
2. Content Styling (**CSS**)
   {presentation, screen reader}
3. Content Interaction (**JavaScript**)

4. Server-side programming and database management

   PHP, Python, Ruby, JavaScript & Java, ASP.NET

What do I need?

Equipment

For a comfortable web development environment, I recommend the following equipment:

A solid, up-to-date computer. Macintosh, Windows, or Linux is fine. Creative departments in professional web development companies tend to be Mac-based. Although it is nice to have a super-fast machine, the files that make up web pages are very small and tend not to be too taxing on computers. Unless you're getting into sound and video editing, don't worry if your current setup is not the very latest and greatest.

Extra memory. Because you'll tend to bounce between a number of applications, it's a good idea to have enough RAM installed on your computer to allow you to leave several memory-intensive programs running at the same time.

A large monitor. Although not a requirement, a large monitor makes life easier, particularly for a visual designer. (I've seen code-based developers get by just fine on an 11" MacBook Air.) The more monitor real estate you have, the more windows and control panels you can have open at the same time. You can also see more of your page to make design decisions.

If you're using a large monitor, just make sure you design for users with smaller monitors and devices in mind.

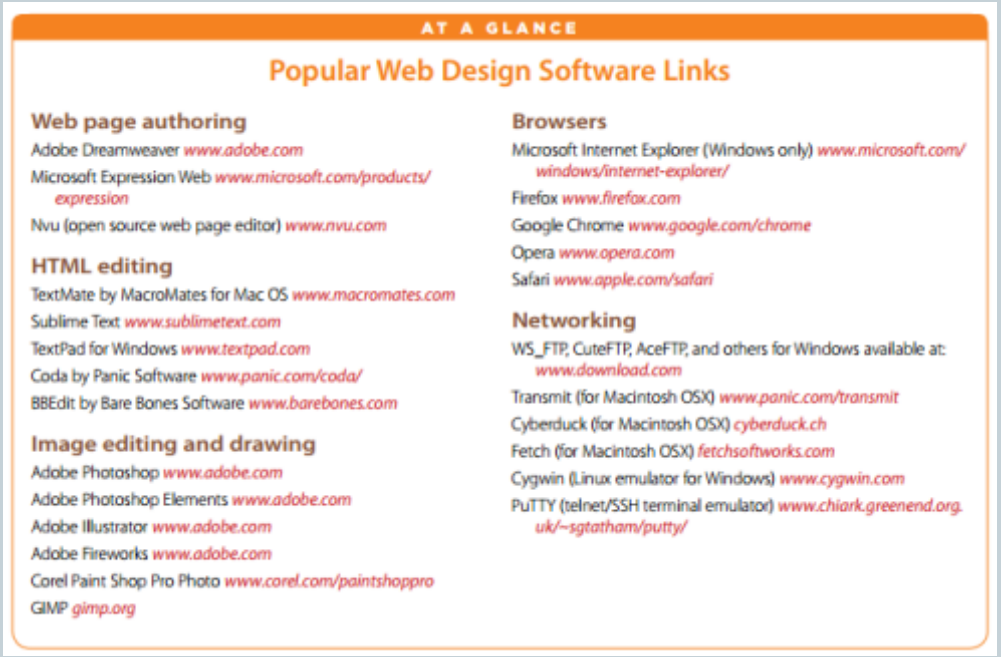
A scanner and/or digital camera. If you anticipate making your own images and textures, you'll need some tools for creating them. I know a designer who has two scanners: one is the "good" scanner, and the other he uses to scan things like dead fish and rusty pans.

A second computer. Many web designers find it useful to have a test computer running a different platform than the computer they use for development (i.e., if you design on a Mac, test on a PC). Because browsers work differently on Macs than on Windows machines, it's critical to test your pages in as many environments as possible, and particularly on the current Windows operating system. If you are a hobbyist web designer working at home, check your pages on a friend's machine. Mac users should check out the "Run Windows on Your Mac" sidebar.

Mobile devices. The Web has gone mobile! That means it is absolutely critical that you test the appearance and performance of your site on a mobile browser on a smartphone or tablet device. You may already have a smartphone yourself. If you don't have a budget for devices with multiple platforms, ask your friends if you can spend a few minutes looking at your site on theirs. I have one web developer friend who checks out his designs on the phones at his local mobile carrier store (although you might quickly wear out your welcome).

What do I need (software)?

- Web page authoring
  - Adobe Dreamweaver
- HTML editors – NotePad ++
- Image-editing and drawing software
- Internet Tools
  - Web Browser
- File-transfer protocol (FTP) program



AT A GLANCE

Popular Web Design Software Links

Web page authoring

Adobe Dreamweaver *www.adobe.com*

Microsoft Expression Web *www.microsoft.com/products/expression*

Nvu (open source web page editor) *www.nvu.com*

HTML editing

TextMate by MacroMates for Mac OS *www.macromates.com*

Sublime Text *www.sublimetext.com*

TextPad for Windows *www.textpad.com*

Coda by Panic Software *www.panic.com/coda/*

BBEdit by Bare Bones Software *www.barebones.com*

Image editing and drawing

Adobe Photoshop *www.adobe.com*

Adobe Photoshop Elements *www.adobe.com*

Adobe Illustrator *www.adobe.com*

Adobe Fireworks *www.adobe.com*

Corel Paint Shop Pro Photo *www.corel.com/paintshoppro*

GIMP *gimp.org*

Browsers

Microsoft Internet Explorer (Windows only) *www.microsoft.com/windows/internet-explorer/*

Firefox *www.firefox.com*

Google Chrome *www.google.com/chrome*

Opera *www.opera.com*

Safari *www.apple.com/safari*

Networking

WS_FTP, CuteFTP, AceFTP, and others for Windows available at: *www.download.com*

Transmit (for Macintosh OSX) *www.panic.com/transmit*

Cyberduck (for Macintosh OSX) *cyberduck.ch*

Fetch (for Macintosh OSX) *fetchsoftworks.com*

Cygwin (Linux emulator for Windows) *www.cygwin.com*

PuTTY (telnet/SSH terminal emulator) *www.chiark.greenend.org.uk/~sgtatham/putty/*

Chapter Two
How the Web Works

CSIS-180
WEB DESIGN

Browser vs. Server

Web Browser

- Current market leaders:
  - Google's Chrome
    Preferred by Web Developers
  - Firefox
    The legacy of Netscape
  - Apple's Safari
    Leading in mobile browsing
  - Microsoft's Internet Explorer
    Still around

Web Server

- Current market leaders:
  - Apache
    http://httpd.apache.org/
  - Microsoft's IIS
    Internet Information Services
  - Nginx
    http://wiki.nginx.org/
  - GWS
    Google's Web Server

Browser vs. Server

Web Browser

- Makes http requests
  - Asks for web pages
- Renders web pages
  - Converts HTML, CSS and JavaScript into displayed document.
- Remembers browsing history, preferences and cookies

Web Server

- Responds to http requests
  - Sends web pages
- Processes requests
  - Create dynamic pages
  - Run web applications
  - Fetch data from Databases
  - Store session data

Development: Backend vs. Frontend

Frontend:

What you see on the Web Browser

- Graphic design
- Image production
- Interface design
- User experience
- **HTML** markup
- **CSS** styles
- Sometimes **JavaScript**

Backend:

What happens on the Web Server

- Processing Data
- Database Programming
- Content Management Systems
- Server-side Scripting
  - PHP, ASP, Ruby, JSP
- Sometimes **JavaScript**

Web Page Addresses (URLs)

The parts of a URL

A complete URL is generally made up of three components: the protocol, the site name, and the absolute path to the document or resource, as shown in Figure 2-1.

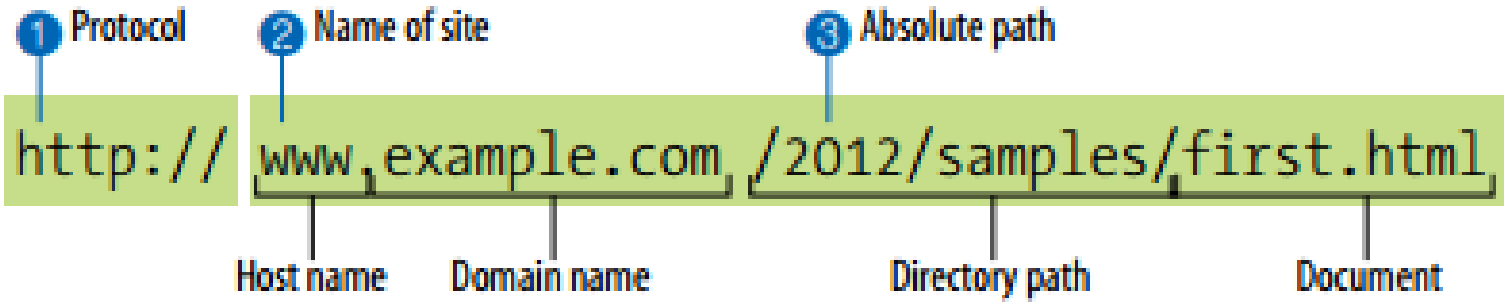


Figure 2-1. The parts of a URL.

Default files

Default files

Obviously, not every URL you see is so lengthy. Many addresses do not include a filename, but simply point to a directory, like these:

```
http://www.oreilly.com
http://www.jendesign.com/resume/
```

When a server receives a request for a directory name rather than a specific file, it looks in that directory for a default document, typically named *index.html*. So when someone types the above URLs into his browser, what he'll actually see is this:

```
http://www.oreilly.com/index.html
http://www.jendesign.com/resume/index.html
```

Quick Introduction to HTML markup language

four separate files: an HTML text document, a style sheet and two images. Tags in the HTML source document give the browser instructions for how the text is structured and where the images should be placed.

index.html

```
<!DOCTYPE html>
<html>
<head>
<title>Jen's Kitchen</title>
<link rel="stylesheet" href="kitchen.css" type="text/css" >
</head>

<body>
<h1><img src="foods.gif" alt="food illustration"> Jen&rsquo;s Kitchen</h1>

<p>If you love to read about <strong>cooking and eating</strong>, would like to find out about
some of the best restaurants in the world, or just want a few choice recipes to add to your
collection, <em>this is the site for you!</em></p>

<p><img src="spoon.gif" alt="spoon illustration"> Your pal, Jen at Jen's Kitchen</p>
<hr>
<p><small>Copyright 2011, Jennifer Robbins</small></p>
</body>
</html>
```

Quick Introduction to HTML markup language

Jen's Kitchen

If you love to read about **cooking and eating**, would like to find out about some of the best restaurants in the world, or just want a few choice recipes to add to your collection, *this is the site for you!*

Your pal, Jen at Jen's Kitchen

four separate files: an HTML text document, a style sheet and two images. Tags in the HTML source document give the browser instructions for how the text is structured and where the images should be placed.

kitchen.css

```
body { font: normal 1em Verdana; margin: 1em 10%;}
h1 { font: italic 3em Georgia; color: rgb(23, 109, 109); margin: 1em 0 1em;}
img { margin: 0 20px 0 0; }
h1 img { margin-bottom: -20px; }
small { color: #666666; }
```

foods.gif

spoon.gif

How browsers display web pages

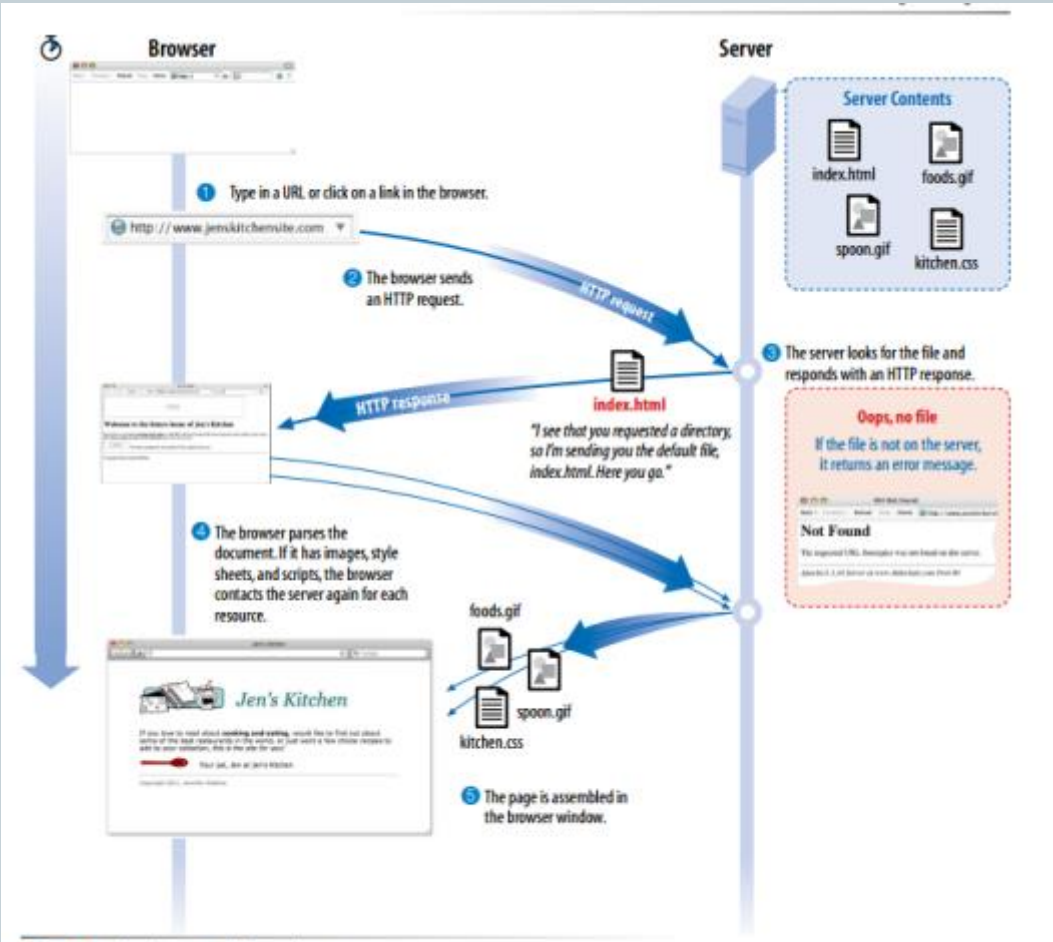


Figure 2-5. How browsers display web pages.

Browser Wars

Early Business Model:

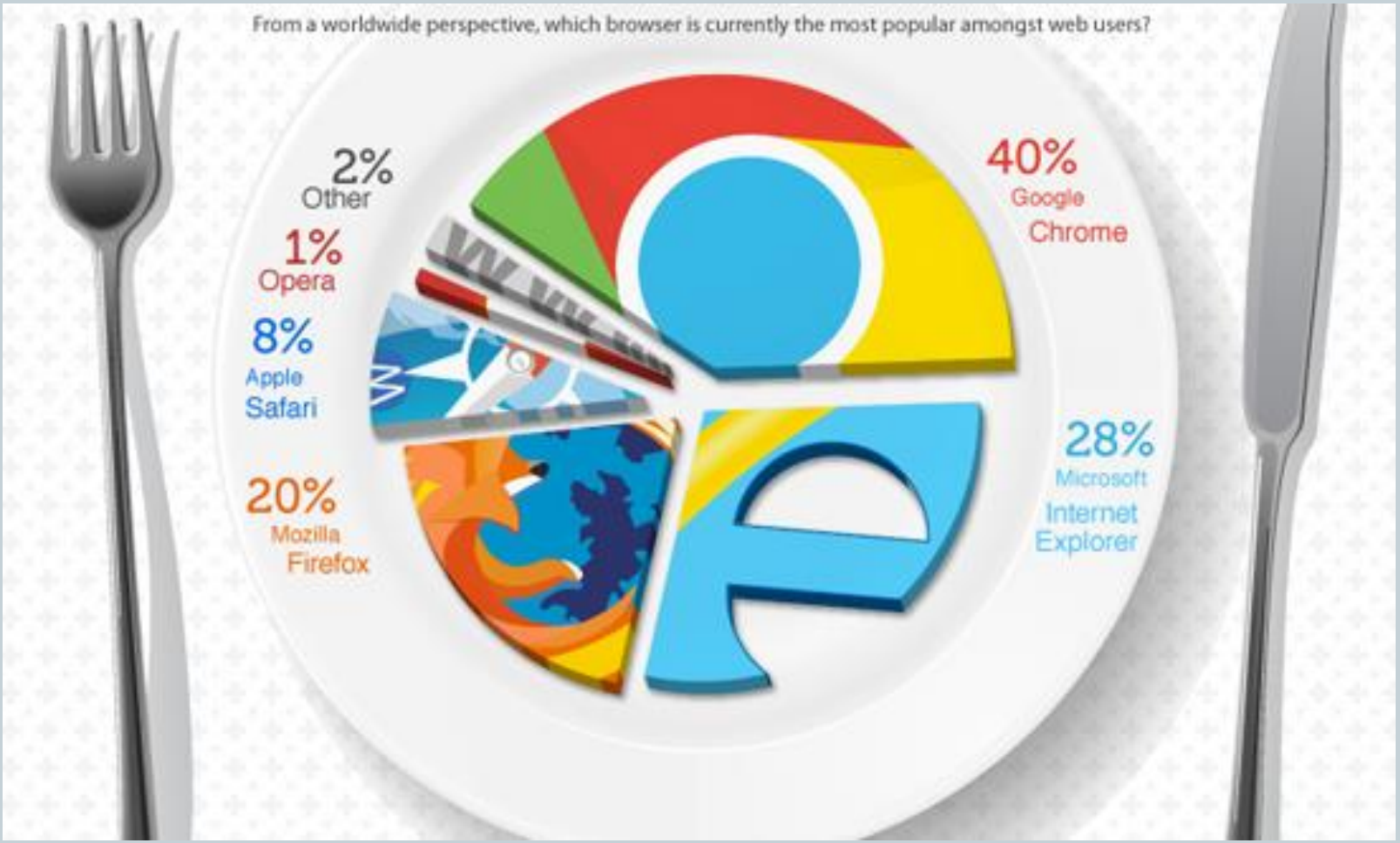
- Give your web browser to users for free
- Sell your web server to companies $$$
- Web pages will work best if your server and browser are both used.
- If more people use your browser, more companies will want to buy your server.

Browser Wars

Rough Timeline:

- In 1995-1998 Netscape dominated early market 90%
- In 2001 Microsoft wiped Netscape out of existence. 92% market share
- In 2008, Firefox was the only browser seriously challenging Internet Explorer (IE)
- In 2012, IE was finally overtaken by Google Chrome
- Today, Apple's Safari is the leader in the mobile browser market. Chrome is not far behind.

Web Browser Wars



Chapter Three
Big Concepts

CSIS-180
WEB DESIGN

Multimedia

Unlike other documents, web pages have many layers capable of combining almost any form of media

- **Text & Images**
  - obviously
- **Audio**
  - Embedded players and files (mp3)
- **Video**
  - Embedded players and fields (mp4)
- **Animation**
  - Flash-driven, JavaScript, jQuery, and CSS-based
- **Interaction**
  - Embedded Programs & Application
  - "Interaction" is a form of media. Very different than passively viewed video.

Internet vs. WWW

Internet

- **Physical Hardware Layer**
  - WiFi Routers
  - Ethernet Switches
  - Cable Modems
- **Key Technologies**
  - TCP/IP Protocol
  - Packet Switching

World Wide Web

- **Widely-available Content Layer of Internet**
  - Web servers
  - Email
  - File sharing (FTP)
- **Key Technologies**
  - HTTP Protocol
  - URLs

Key Internet Concepts

- **TCP/IP:**
  Transmission Control / Internet Protocol
  - Network of Networks concept
- **Packet Switching:**
  Data is broken into small packets that can be independently routed

- Think of the **Internet** as earth's network of highways and ports (sea and air)
- **TCP/IP** establishes standards for roads and ports so people can get everywhere seamlessly.
- **Packet Switching** means cargo can travel in small chunks to easily move through bottlenecks via different paths.

Key WWW Concepts

- **HTTP:**
  Hypertext Transfer Protocol
  - Rules for making requests and responding to requests.
- **URI:**
  Uniform Resource Identifier
  - Unique identifier for finding stuff on the WWW; Includes: **URL** (Location) and **URN** (Unique Name at Location)

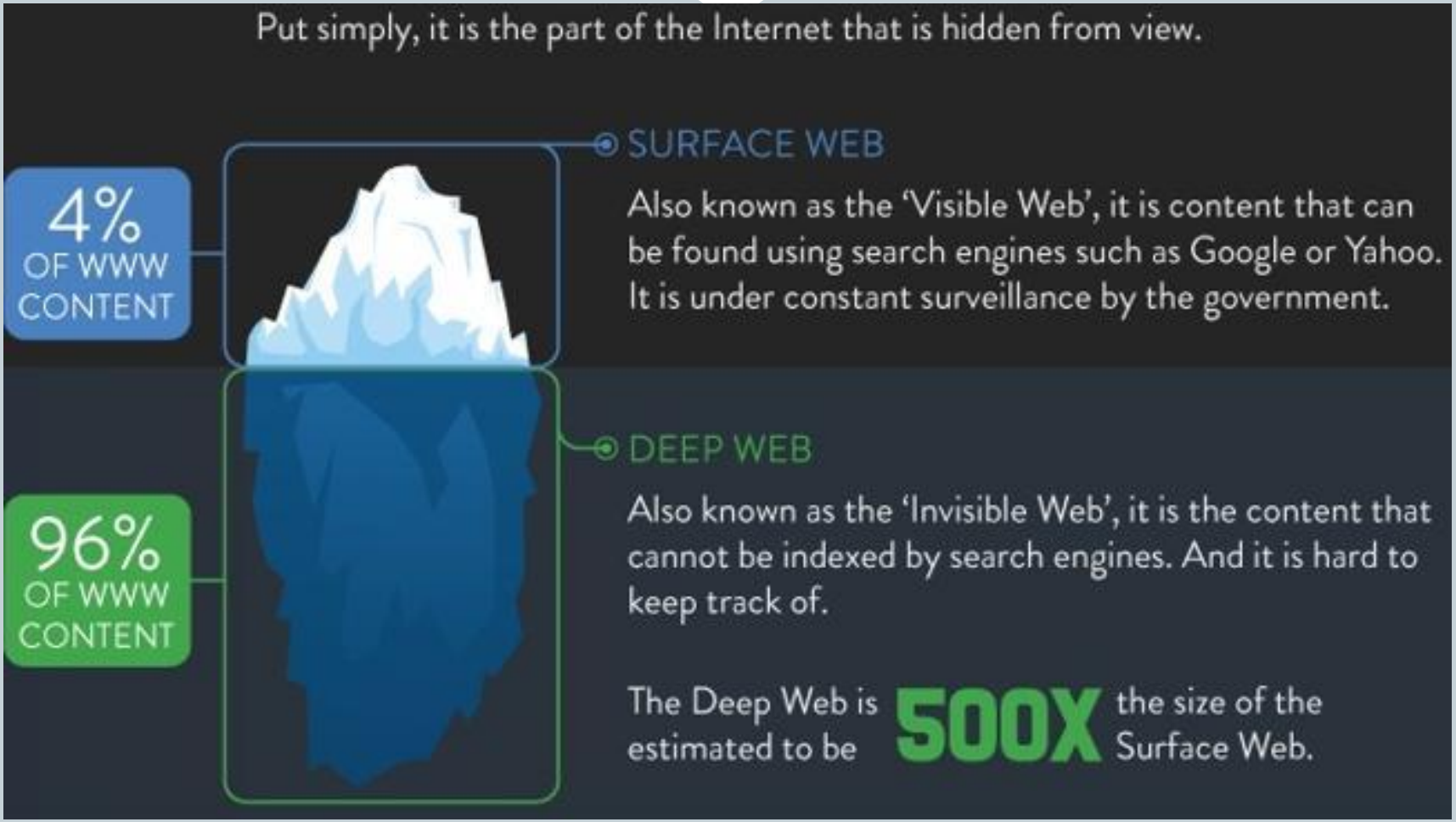
- Think of the WWW as everything that can be publicly accessed from the world's highways, airport and sea ports.
- Think of **HTTP** as the standard language used to ask for directions.
- Think of **URLs** as street addresses and **URNs** as IDs for objects at particular addresses.

Internet is bigger than WWW

- The Internet's protocol (TCP/IP) can support many sub-protocols, some that are proprietary (private/secret).

- Examples:
  - Many peer-to-peer files sharing systems
  - Specialized client-server systems (early banking systems)
  - Content so deeply embedded in systems that it's very hard to find (Deep Web)
  - Content requiring access via secret non-standard browsers (Darknet)

Quick detour about Deep Web



But, the WWW matters more

- The HTTP protocol standardizes requests so any web browser can access any web server.
- URLs (WWW layer) makes finding and remember servers much easier than numeric IP addresses (Internet layer)
- The WWW is all about accessibility via open, widely adopted standards.
  - It's the largest, most expandable information system ever built.