

Web Design and Programming

Week 1

18 April 2024

Instructor: Dr. Peeraya Sripian



Class instructor & TA/SA



Professor

Dr. Peeraya Sripian (Gift)

peeraya@shibaura-it.ac.jp

TA & SA

Faru (PhD)

Kiru (B4)

Aarif (B3)

Ujwal (B2)

Manishk (B2)

Ridhi (B2)

Course objectives

- To understand and be able to use tools for web development
- To learn about fundamental web design and programming technologies (HTML, CSS, JavaScript PHP, MySQL)
- To learn client-side programming for creating interactive webpage and server-side scripting to generate dynamic web content.
- To create and publish well-designed and responsive webpages.

Why did you register for this class?

Let's hear the answer

There are many reasons to learn how to code

- as a developer at a company

- freelancing or create your own business

- a consultant

- create an application to make money

- code as a hobby

To boost your Application development skills

- ML, AI, Robotics and many other domains need a front end to connect with the user.

It is creative and fun!

About this course

Time

- 9AM – 12:30PM

Room

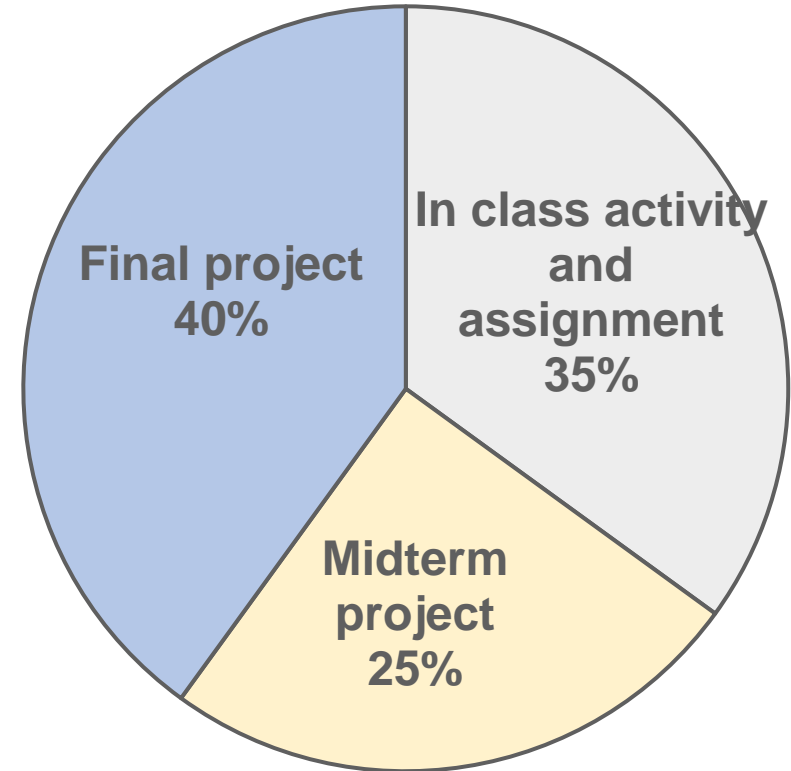
- PC Lecture room 3

Schedule for each class:

- Part 1. Lecture (~ 70-80 mins)
 - Break (~ 5-10 mins)
- Part 2. Exercise (the rest of the class)

Scoring system:

- To get the credit for this class – you need to obtain more than 60%



How to get help

- Slack workspace 

Course schedule

Week	Date	Topic
1	4/18	Intro to WWW, Intro to HTML
2	4/25	CSS Fundamental
	5/2	Holiday (GW)
3	5/9	CSS and Bootstrap
4	5/16	Work on midterm project
5	COIL 22 MAY 18:00-19:30 (counted as 1 class, replacing 23 May)	
6	5/30	Midterm project presentation week
7	6/6	PHP fundamentals + Installation XAMPP
8	6/13	PHP fundamentals 2
9	6/20	mySQL fundamentals
10	6/27	Assessing MySQL using PHP, MVC pattern + Intro of Final project
11	7/4	Cookies, sessions, and authentication + Proposal of final project
12	7/11	Javascript and PHP validation
13	7/18	Final project development
14	7/25	Final project presentation

About submitting homeworks

- We will use **github** to submit the assignment (only for front-end – until week6).
- You need to register for git account using **shibaura-it.ac.jp** email address (no personal email address is allowed)
- We will explain more during practice session of this class.

Learning objectives for this week

- Theory about WWW and internet
 - Internet milestone
 - HTML
 - IP addressing
 - Domain name
 - URL
- Introduction to HTML (1)

It takes a (virtual) village

Website creation roles

Website Creation Roles

- Content
- Design
- Coding: Frontend
- Coding: Backend

Content

Information architecture (IA)

Organizes content logically for ease of findability

Content strategy

Ensures that text **supports** the brand/marketing goals; may include data modeling and updating schedules as well as extending brand voice to social media



Design

User Experience (UX)

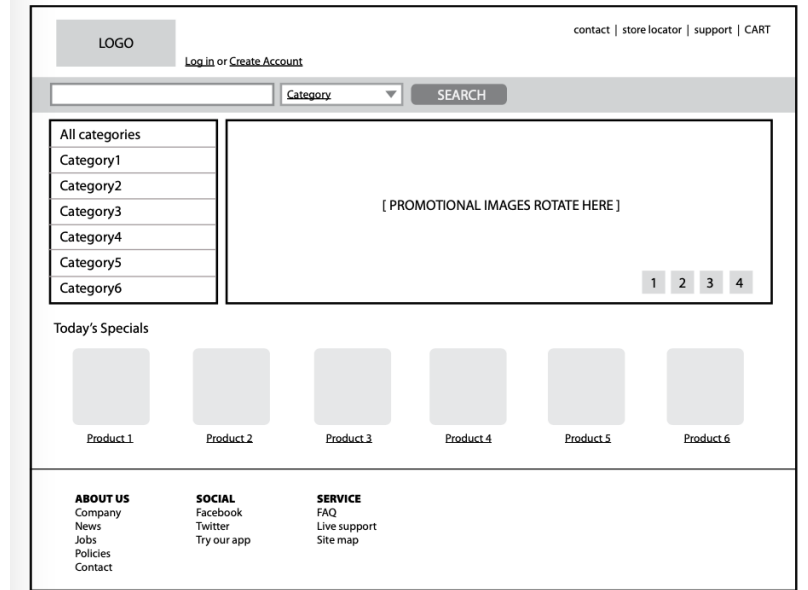
Makes sure whole experience with the site and the product is favorable based on user testing

Interaction Design (IxD)

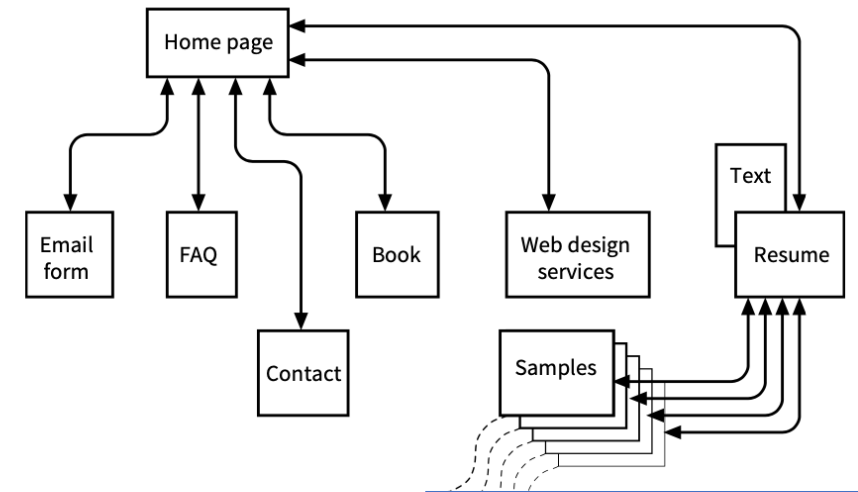
Focuses on how to use the site, including its User Interface (UI)

Visual Design

Creates the “look and feel” of the site



Wireframe diagram



Simple site diagram

Coding: Frontend Development

Authoring/Markup (HTML)

Styling (CSS)

JavaScript and DOM Scripting

Frameworks



Coding: Backend Development

Server software (ex: Apache, MS IIS)

Web application languages (ex: PHP, Ruby, .NET)

Database software (ex: MySQL, Oracle)



Other Web Roles

Product manager

Guides product in a way that meets business goals

Project manager

Coordinates team, schedule, and processes

SEO specialist

Ensures site is highly ranked in web searches

Multimedia producer

Creates sound, video, animation, and interactive media.

Typical Equipment

- Solid up-to-date computer
- Large monitor (or several)
- Second computer for testing
- Mobile devices for testing
- Scanner and/or camera (for artwork)

Software Typically Used

- Coding tools (Atom, etc..)
- User Interface/layout tools (Mockflow, Sketch, etc..)
- Web graphic creation tools (Adobe PS, AI, GIMP, etc..)
- Variety of browsers (Chrome, FF, MS Edge, Safari, IE)
- File management/transfer tools (Filezilla, WinSCP)



Theory of internet and the WWW

What is the internet

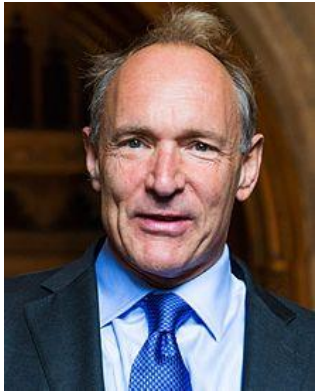
- The **largest network** in the world that connects hundreds of thousands of individual networks all over the world.
- The popular term for the Internet is the “**information highway**”.
- Rather than moving through geographical space, it moves your ideas and information through **cyberspace** – the space of electronic movement of ideas and information.



Internet

- No one owns it
- It has no formal management organization.
- As it was originally developed by the Department of defense, this lack of centralization made it less vulnerable to wartime or terrorist attacks.
- To access the Internet, an existing network need to pay a small registration fee and agree to certain standards based on the TCP/IP (Transmission Control Protocol/Internet Protocol)
- Information transferred using email (POP3/IMAP/SMTP), file transfer (FTP), secure shell (SSH), etc.. → Protocols .

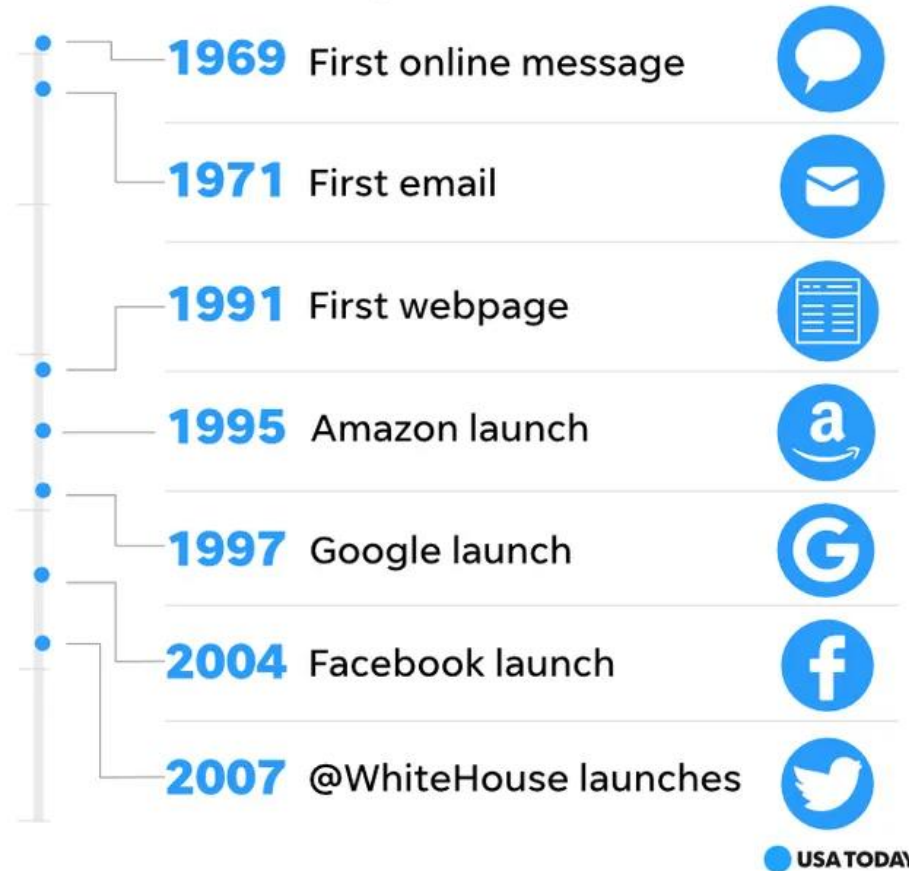
Internet milestones



Tim Berners-Lee

Provided a system for sharing documents via hyperlinks

Internet history milestones



→ Online msg “lo” by Apranet R&D Project by UCLA. It took an hour to send the whole word

→ First email was sent by MIT researcher, the first time ‘@’ was used to designate a specific recipient of a message

→ Invented by Tim Berners-Lee in 1989, published in 1991

→ As a virtual bookstore

→ Google search

→ Followed by iTunes in 2003, Facebook in 2004, Youtube in 2005, Twitter in 2006, and Instagram in 2010

What is Web?

- The **Web (World Wide Web)** consists of information organized into Web pages containing text and graphic images.
- It contains hypertext links, or highlighted keywords and images that lead to related information.
- A collection of linked Web pages that has a common theme or focus is called a **Web site**.
- The main page that all of the pages on a particular Web site are organized around and link back to is called the site's **home page**.
- The web use a protocol called HTTP

Client/Server Structure of the Web

- Web is a collection of files that reside on computers, called **Web servers**, that are located all over the world and are connected to each other through the Internet.
- When you use your Internet connection to become part of the Web, your computer becomes a **Web client** in a worldwide client/server network.
- A **Web browser** is the software that you run on your computer to make it work as a web client.

Hypertext Markup Language (HTML)

- The public files on the web servers are ordinary text files, much like the files used by word-processing software.
- To allow Web browser software to read them, the text must be formatted according to a generally accepted standard.
- The standard used on the web is Hypertext markup language (**HTML**).

Hypertext Markup Language (HTML)

- HTML uses codes, or tags, to tell the Web browser software how to display the text contained in the document.
- For example, a Web browser reading the following line of text:

```
<B> A Review of the Book <I>Wind Instruments of the 18th Century</I></B>
```

recognizes the `` and `` tags as instructions to display the entire line of text in **bold** and the `<I>` and `</I>` tags as instructions to display the text enclosed by those tags in *italics*.

Addresses on the Web: IP Addressing

- **IP** (Internet protocol) is a simple protocol for attempting to send data between two computers.
- The IP addressing system currently in use on the Internet is a **32-bit IP address**, written as four 8-bit numbers (0-255).
- Each part of the address is a number ranging from 0 to 255, and each part is separated from the previous part by period,
- For example, 106.29.242.17

IP Addressing

- The combination of the four IP address parts provides 4.2 billion possible addresses ($256 \times 256 \times 256 \times 256$).
- This number seemed adequate until **1998**.
- The most recent version of IP is **IPv6**, initiated by Internet Engineer Taskforce in early 1994
- IPv6 is a **128-Bit IP address**

Tips: you can find your own IPv4 and IPv6 address easily using <https://whatismyipaddress.com/>

IPv4

VS

IPv6

Example: 127.255.255.255

Features of IPv4

- Connectionless Protocol
- Allow creating a simple virtual communication layer over diversified devices
- It requires less memory, and ease of remembering addresses
- Already supported protocol by millions of devices
- Offers video libraries and conferences

Example:

2001:0db8:85a3:0000:0000:8a2e:0370:7334

Features of IPv6

- Hierarchical addressing and routing infrastructure
- Stateful and Stateless configuration
- Support for quality of service (QoS)
- An ideal protocol for neighboring node interaction

Transmission Control Protocol (TCP)

- Adds multiplexing, guaranteed message delivery on top of IP
- **multiplexing**: multiple programs using the same IP address
 - **port**: a number given to each program or service
 - port 80: web browser (port 443 for secure browsing)
 - port 25: email
 - port 22: ssh
- some programs (games, streaming media programs) use simpler UDP protocol instead of TCP

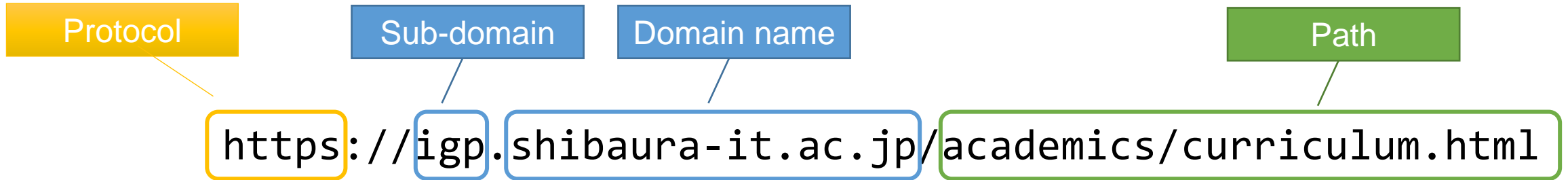
Domain Name Addressing

- A set of servers that map written names to IP addresses.
 - Example:
www.google.com → 216.58.196.228
www.shibaura-it.ac.jp → 153.120.129.227
- A **domain name** is a **unique name** associated with a specific IP address by a program that runs on an Internet host computer.
- This program, which coordinates the IP addresses and domain names for all computers attached to it, is called **DNS (Domain Name System) software**.
- The host computer that runs this software is called a **domain name server**.

Uniform Resource Locators

- The IP address and the domain name each identify a particular computer on the Internet.
- However, they do not indicate where a Web page's HTML document resides on that computer.
- To identify a Web pages **exact location**, Web browsers rely on Uniform Resource Locator (URL).
- URL is a four-part addressing scheme that tells the Web browser:
 - What transfer protocol to use for transporting the file
 - The domain name of the computer on which the file resides
 - The pathname of the folder or directory on the computer on which the file resides
 - The name of the file

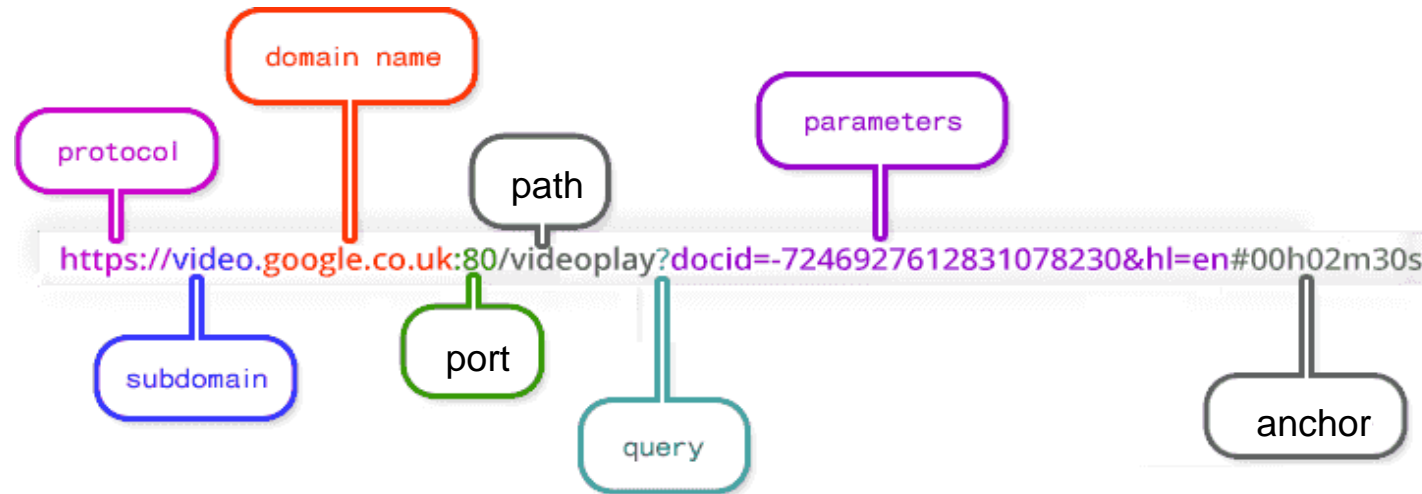
Structure of a Uniform Resource Locators



upon entering this URL into the browser, it would

- ask the DNS server for the IP address of *shibaura-it.ac.jp*
- connect to that IP address at port 80
- ask the server to GET */academics/curriculum.html*
- display the resulting page on the screen

More advanced URLs



- Anchor – jumps to a given section of a webpage
- Port – for web servers on ports other than the default 80 (eg: 8080)
- Query string – a set of parameters passed to a web program

HTTP

- The transfer protocol is the **set of commands** that the computers use to move files from one computer to another on the Internet.
- The most common transfer protocol used on the Internet is the Hypertext Transfer Protocol (**HTTP**).
- Two other protocols that you can use on the Internet are the **File Transfer Protocol (FTP)** and the Telnet Protocol
- Browser send these commands internally
 - GET filename: download
 - POST filename: send a web form response
 - PUT filename: upload

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
301-303	Page has moved (permanently or temporarily)
403	You are forbidden to access this page
404	Page not found
500	Internal server error

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
- [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

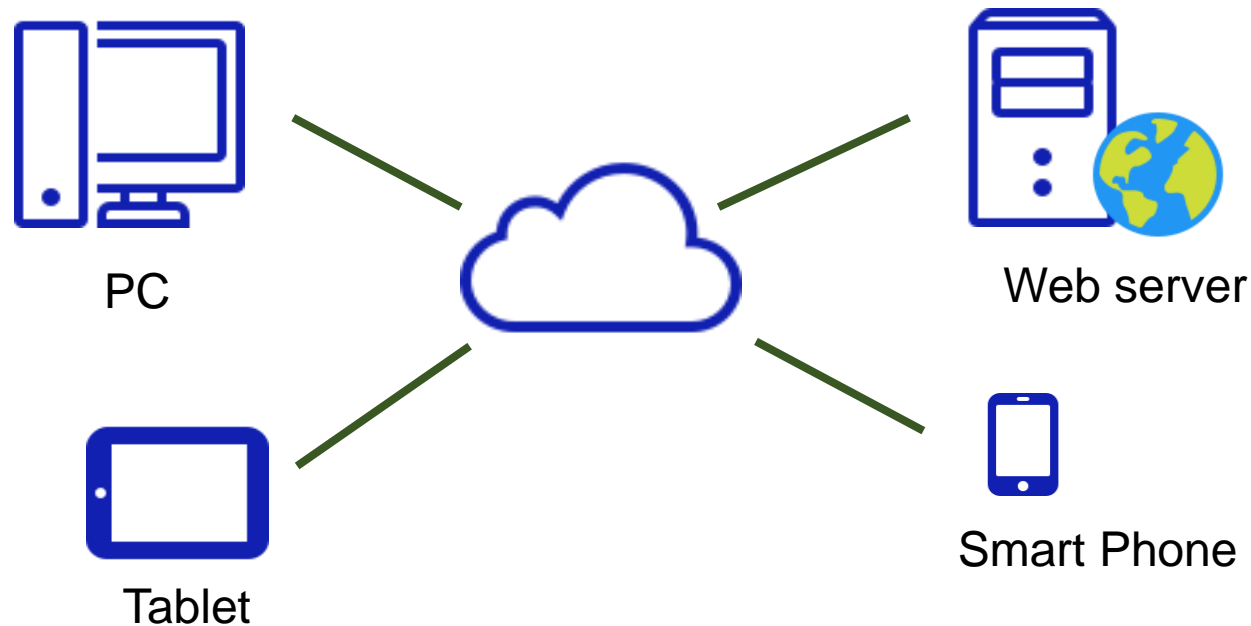
Internet vs. WWW in summary

- Internet is a worldwide set of computer networks for personal, commercial, academic, and government use.
- WWW is a subset of the internet containing a set of web pages and sites.
- IP is used for addressing of internet communication, each computer on the internet has IP address.
- Web emails file transfer are layered on top of TCP
- URL map to IP address through a service called DNS
- HTTP allows a browser to request a web document
- Web programming involves many languages: HTML, CSS, PHP, Javascript, XML, and SQL

How webpages are processed

How web applications work

- The components of a web application

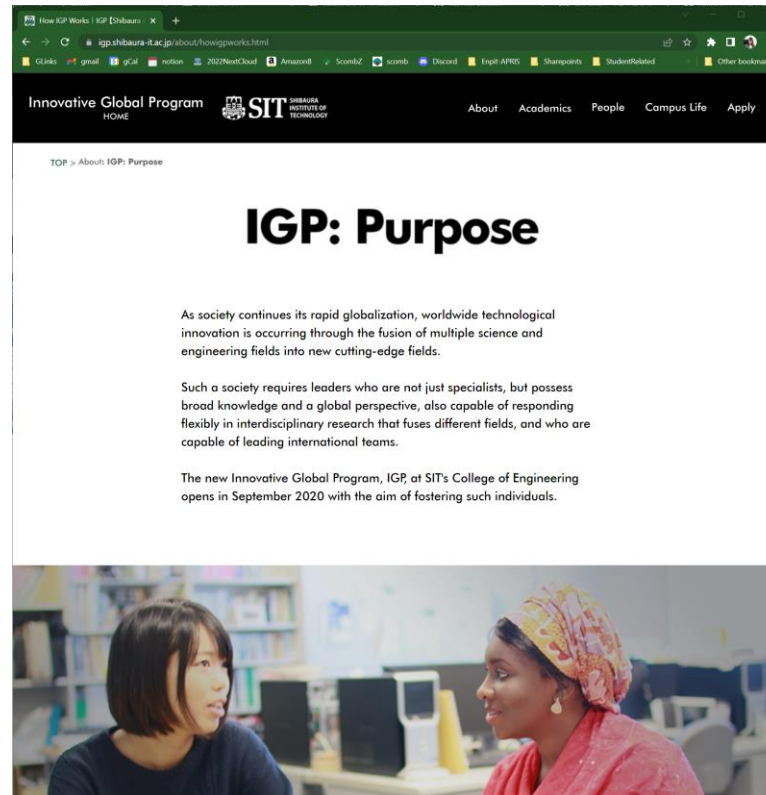


- Clients are programs –web browsers) to request web pages from the web server.
- Web server then returns the pages that are requested to the browser.

Two types of webpages

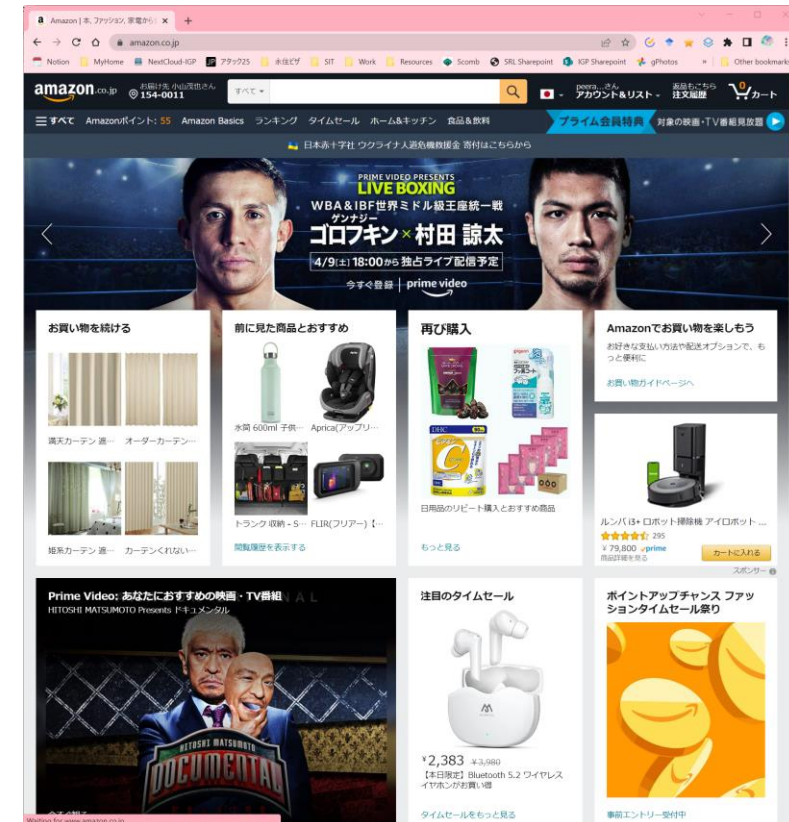
Static

<https://igp.shibaura-it.ac.jp/about/howigpworks.html>

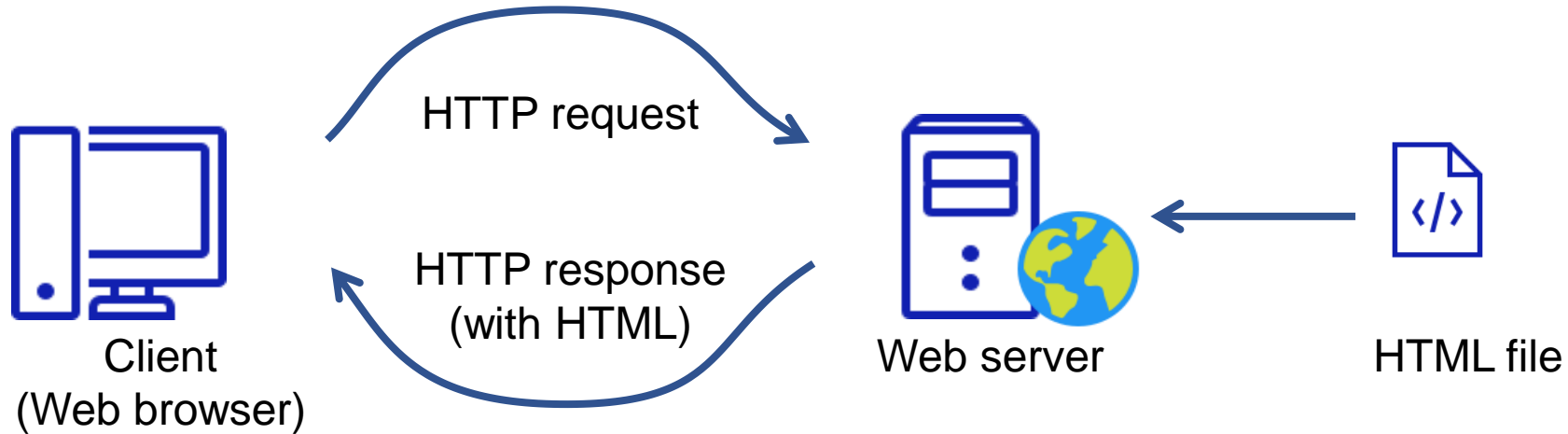


Dynamic

<https://www.amazon.com/>

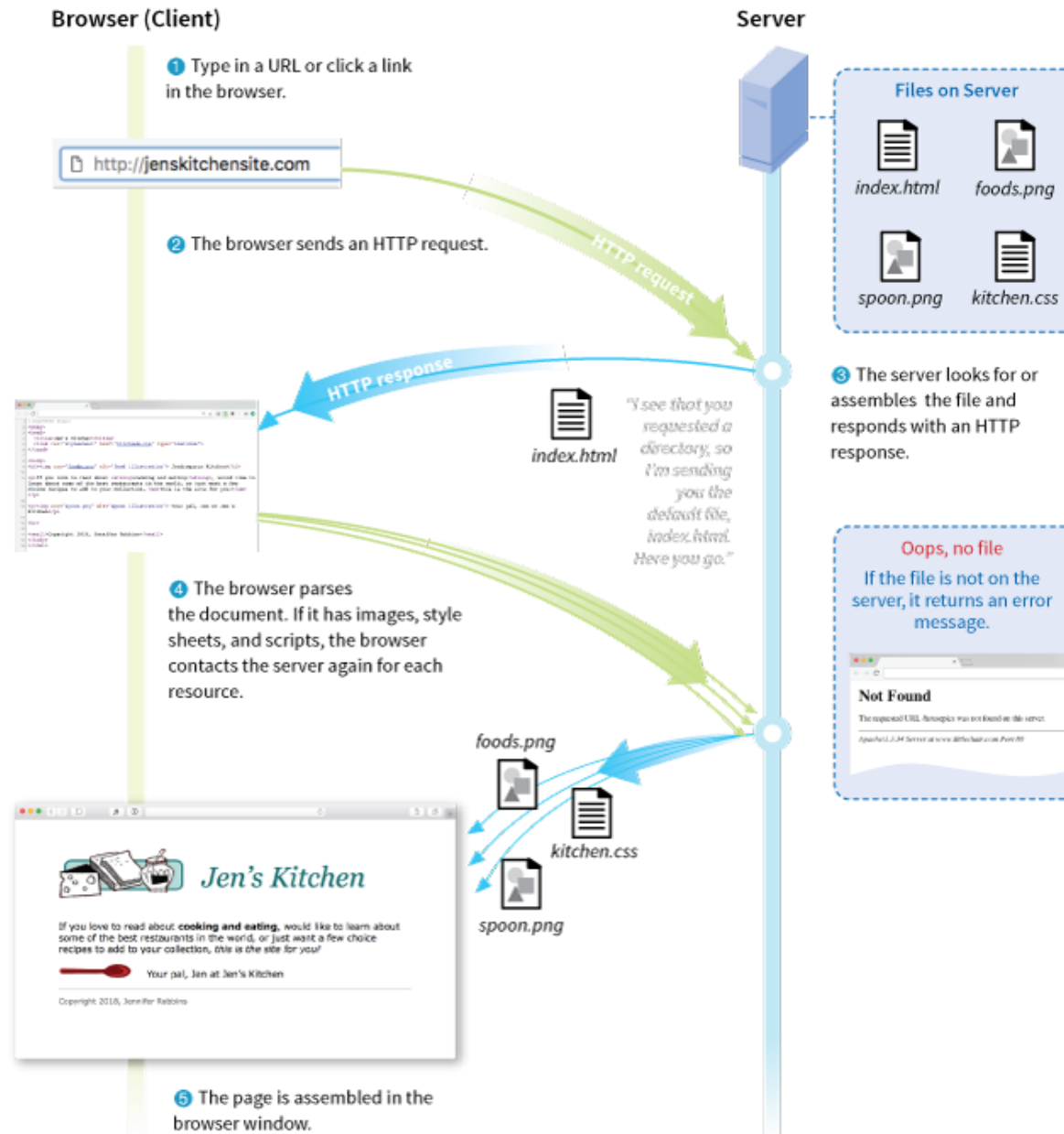


How static web pages are processed

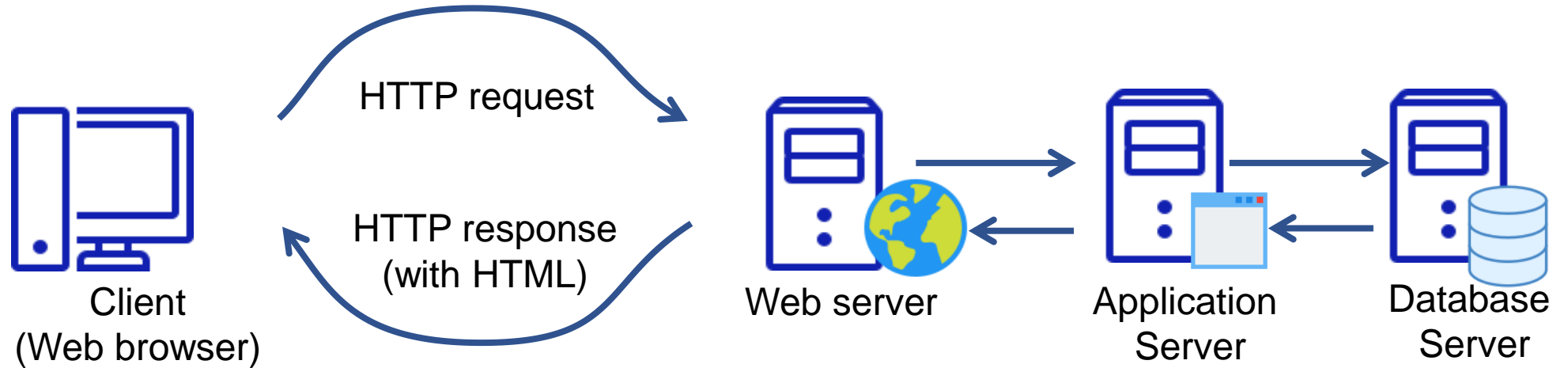


- A static webpage is an HTML document stored on a web server and does not change
- Usually with filenames .htm .html
- When user request a static webpage, the browser send HTTP request include the filename
- Web server retrieve HTML from webpage and send it back to browser as part of HTTP response
- Browser then renders the HTML into a webpage and display

Example of how static web pages are processed



How dynamic web pages are processed



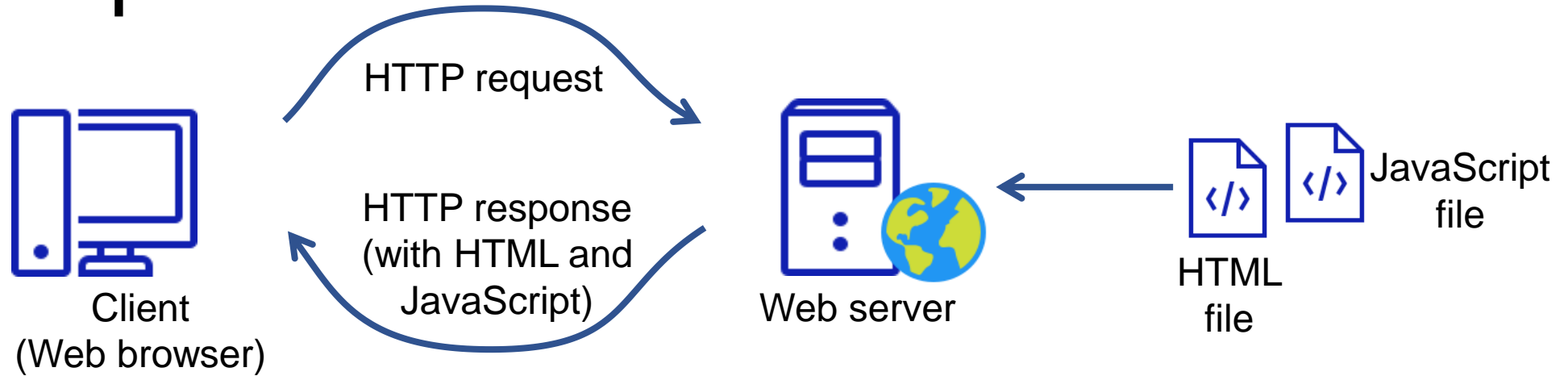
- A dynamic webpage is a webpage that is generated by a server-side program or script.
- A web server would look up the **extension** of the requested file to find out which **application server** should process the request
- Application server runs the specified script. Often uses the data get from web browser to get the appropriate data from a **database server**.
- When application server finishes processing the data, it generates the **HTML** for a web page and returns to web server, then return to web browser as part of HTTP response.

Server-side scripting languages

- To develop dynamic web pages, we need a server-side scripting language.
- Scripting languages run on specific web servers such as IIS and Apache.

Language	Description	Used by
Node.js	Quickest growing. JavaScript code on server-side. No need to learn a new language for back-end development. Good for real time application	Paypal, Uber, LinkedIn, Netflix
PHP	The most used (80% of websites now). Good for content-based web pages. Embed in HTML code. Easy to write.	Facebook, Wikipedia, Wordpress
Java	Good for enterprise, performance.	Google, Amazon, eBay
Ruby	Developing applications fast. Adopted by a lot of startups.	Twitter (early days), Github, Airbnb
Python	Clear and easy to read syntax. Popular for statistical analysis.	Youtube, Instagram, Dropbox, Quora

How client-side JavaScript fits into web development



- JavaScript – a scripting language for client-side processing.
- Common uses – Data validation, Image swaps and rollovers, Accordions
- Since it does not run on server, it does not require a trip back to the server. Therefore, the application will run more efficiently.

Anatomy of a Web Page

The page you see in the browser window is nearly always made up of multiple files, including:

- An **HTML document** (gives the content structure)
- **Style sheets** (describes how it should look)
- **Images and other media** (embedded on the page on the fly)
- **Scripts** (add behaviors and functionality)

A Web Page and Its Components



index.html

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Jen's Kitchen</title>
  <link rel="stylesheet" href="kitchen.css" type="text/css">
</head>

<body>
<h1> Jen's Kitchen</h1>

<p>If you love to read about <strong>cooking and eating</strong>, would like to learn 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> Your pal, Jen at Jen's Kitchen</p>

<hr>
<small>Copyright 2018, Jennifer Robbins</small>
</body>
</html>
```

kitchen.css

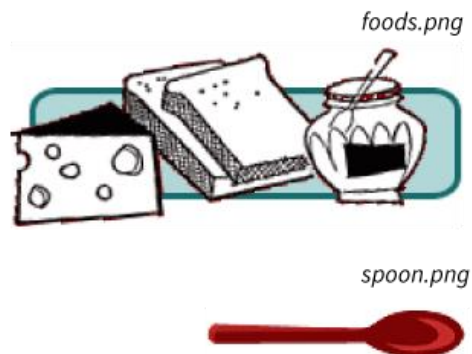
```
body { font: normal 1em Verdana; width: 80%; margin: 1em auto; }

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; }
```



What Style Sheets Do



Browser's default rendering



Simple style sheet applied

Some big concepts you need to
know

A Multitude of Devices

- Your web pages will be viewed on all manner of devices, large and small, fast and slow, visual and non-visual.
- One of the challenges of being a web designer is creating a good experience regardless of the browsing device.



Brad Frost's depiction of the web viewing environment.

Web Standards

- The World Wide Web Consortium (W3C) writes the specifications for web technologies:
[w3.org/standards](https://www.w3.org/standards)
- Sticking with web standards ensures **consistency** across browsers and **forward-compatibility**.

Progressive Enhancement

Progressive enhancement is a strategy for coping with unknown browser capabilities.

- Start with baseline experience that provides content and basic functionality even on minimal browsers and assistive devices
- Layer on styles, scripts, and advanced features for browsers that can handle them
- Finish with “nice to have” effects (like animation) that aren’t critical to the brand or functionality

Progressive Enhancement (cont'd)

HTML strategy

Write in a logical order, with elements marked up in a meaningful way

Style strategy

Use universally supported properties as the baseline and add cutting-edge styles as embellishment

Scripting strategy

Make sure basic functionality (like content display, linking, and forms) are possible when JavaScript is turned off. Enhance the experience when JavaScript is available

Responsive Web Design

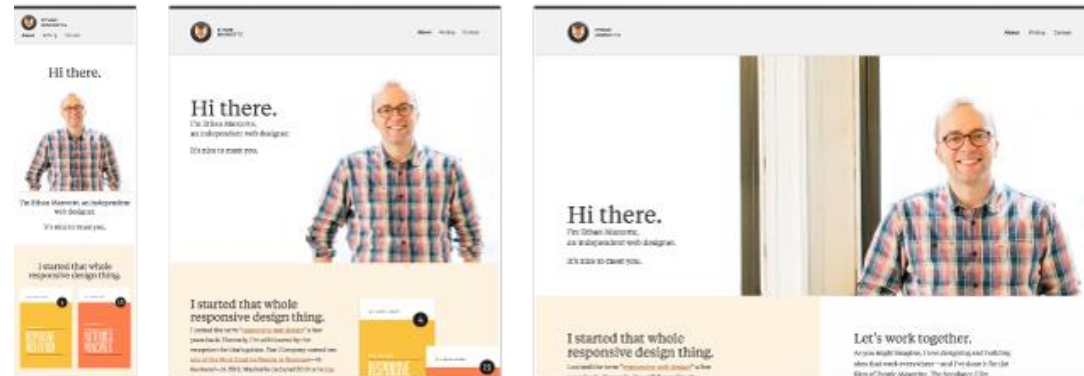
Responsive web design is a strategy for dealing with unknown screen size:

- The heart of the method is using one HTML source for all devices and swapping out the styles based on the size of the browser window (viewport)
- It is preferred to building separate sites just for mobile devices (“m.dot” sites)
- It may not be the solution for all sites, but making sites that adapt to screen size is now common practice

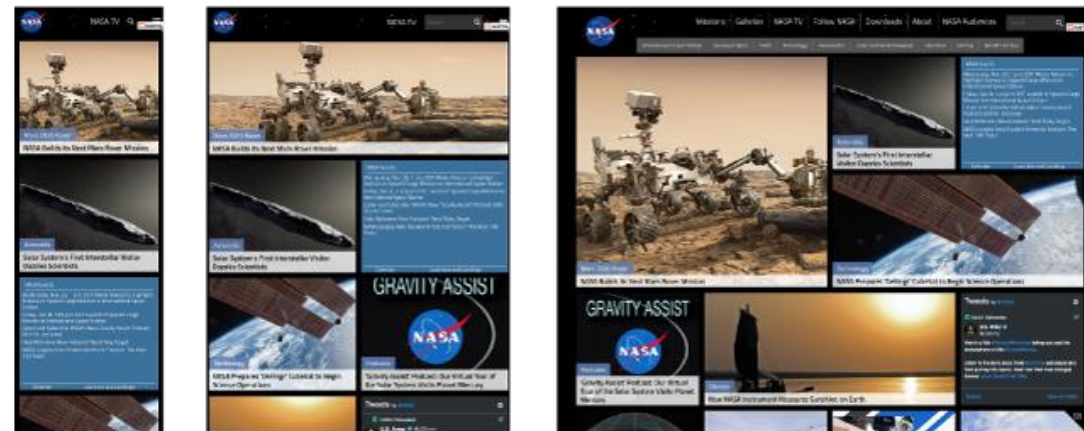
Responsive Web Design (cont'd)

Page layout changes based on the width of the screen.

Ethan Marcotte personal site
ethanmarcotte.com



NASA
nasa.gov



Accessibility

- **Users access web content in many ways:** Keyboard, mouse, voice commands, screen readers, Braille output, magnifiers, joysticks, foot pedals, and so on
- **Four broad categories of disabilities** affect how people interact with computers:
 - Vision impairment
 - Mobility impairment
 - Auditory impairment
 - Cognitive impairment

Accessibility (cont'd)

- There are measures you can take to improve the accessibility of your web pages.
- The Web Accessibility Initiative (WAI) is the group responsible for making web technologies accessible: www.w3.org/WAI.
- The WAI-ARIA (Accessible Rich Internet Applications) specification documents accessibility features.

Site Performance

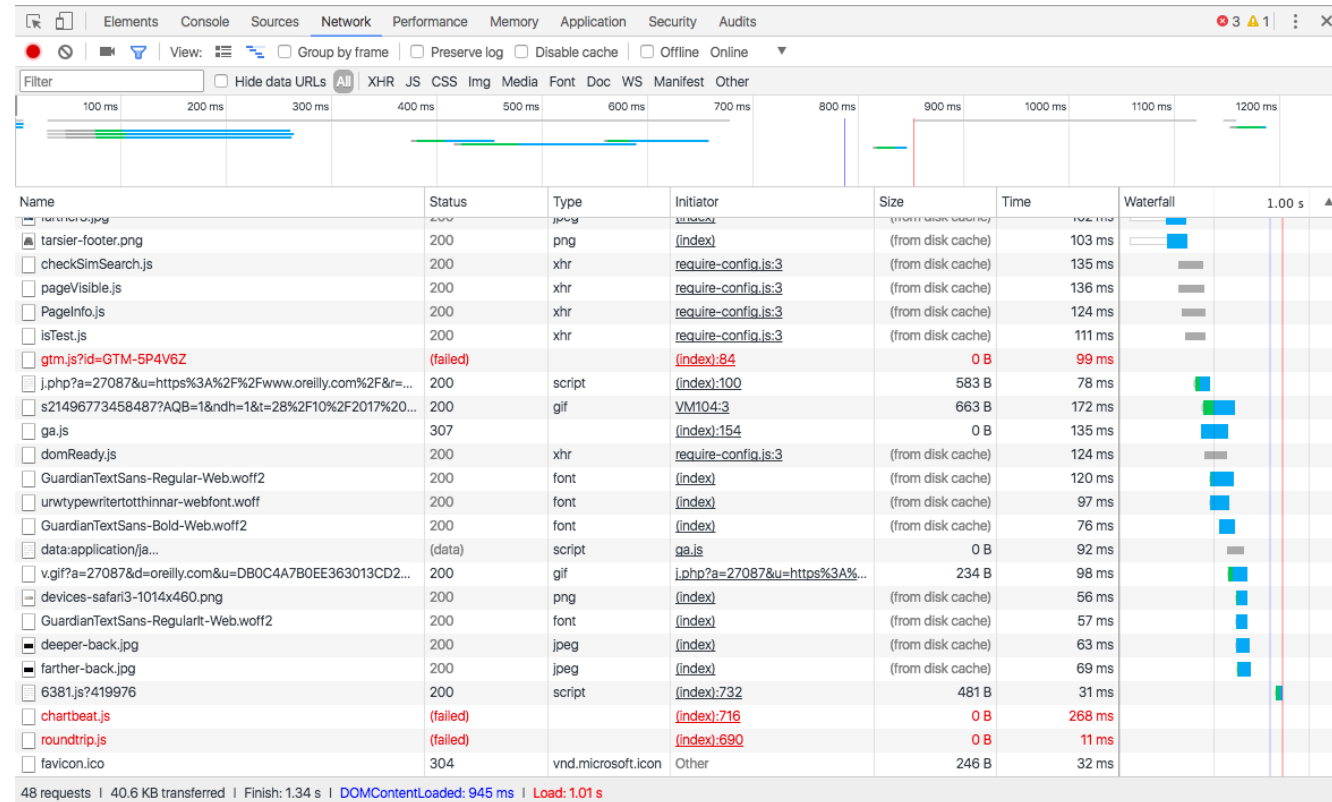
- It is critical that web pages display as **quickly** as possible.
- Users on mobile devices generally leave a page if it does not display in **3 seconds**.
- Even milliseconds can affect the bottom line on retail sites.

Site Performance Tips

- Make image files as small as possible.
- Streamline HTML markup.
- Keep JavaScript to a minimum.
- Add scripts in a way that they don't block page rendering.
- Don't load unnecessary assets.
- Reduce the number of times the browser makes requests of the server.

Site Performance Tools

- Use a **waterfall chart** to see what assets are downloading for your page and how many milliseconds they take.
- This tool is built into the Chrome browser (Developer > Developer Tools).
- Also available in other browser



HTML Overview

Hypertext Markup Language (HTML)



- Describes the *content* and structure of information on a web page
- Not the same as the presentation (appearance on screen)
- Surrounds text content with opening and closing tags
- Each tag's name is called an element
 - syntax: `<element>` content `</element>`
 - example: `<p>`This is a paragraph`</p>`

Content Without Markup

Without HTML markup to describe content structure, text runs together; line breaks are ignored:

Black Goose Bistro

The Restaurant

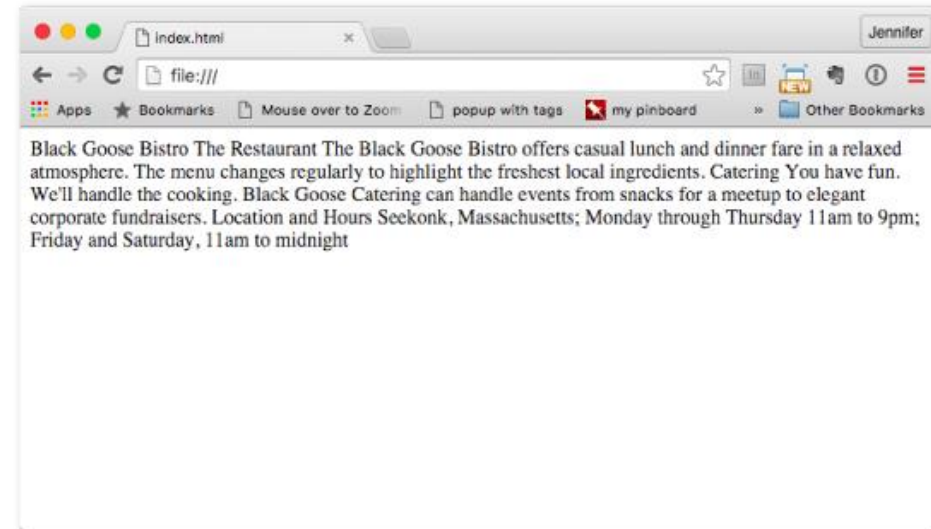
The Black Goose Bistro offers casual lunch and dinner fare in a relaxed atmosphere. The menu changes regularly to highlight the freshest local ingredients.

Catering

You have fun. We'll handle the cooking. Black Goose Catering can handle events from snacks for a meetup to elegant corporate fundraisers.

Location and Hours

Seekonk, Massachusetts;
Monday through Thursday 11am to 9pm; Friday
and Saturday, 11am to midnight



What Browsers Ignore

- Multiple character spaces (white space)
- Line breaks (carriage returns)
- Tabs
- Unrecognized markup

Structure of a minimal HTML page

```
<!DOCTYPE html>  
<html>
```

```
<head>  
  information about the page  
</head>
```

Header

```
<body>  
  page contents  
</body>
```

Body

```
</html>
```

- the **header** describes the page
- the **body** contains the page's contents
- an **HTML** page is saved into a file ending with extension `.html`
- **DOCTYPE** tag tells browser to interpret our page's code as HTML5, the

Markup Basics

Text must be marked up meaningfully and accurately (**semantically**) with HTML tags:

- Browsers need markup to display content correctly.
- Markup makes content elements available to scripts and style rules.
- Markup aids search engines.

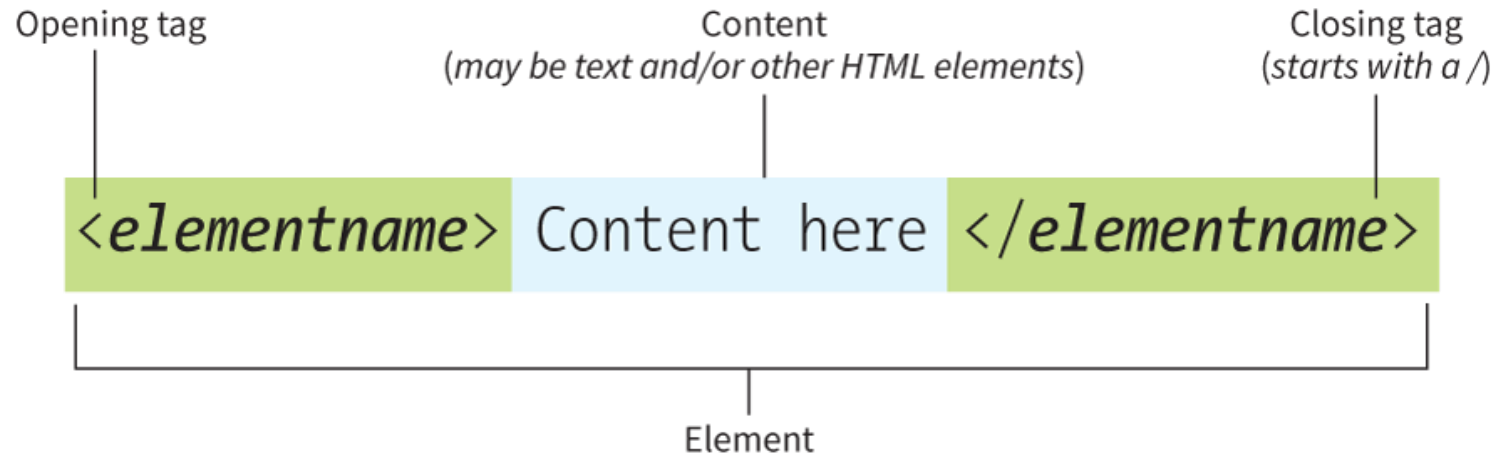
Anatomy of an HTML Element

tag

The element name in angle brackets

element

The content and its markup (start and end tags)



Example:

`<h1>Black Goose Bistro</h1>`

Some Elements Are Empty

Some elements have no content and provide a simple directive. They are called **empty elements**:

```
<element-name>
```

Example: The **br** element inserts a line break.

```
<p>1005 Gravenstein Highway North<br>Sebastopol, CA 95472</p>
```

Attributes

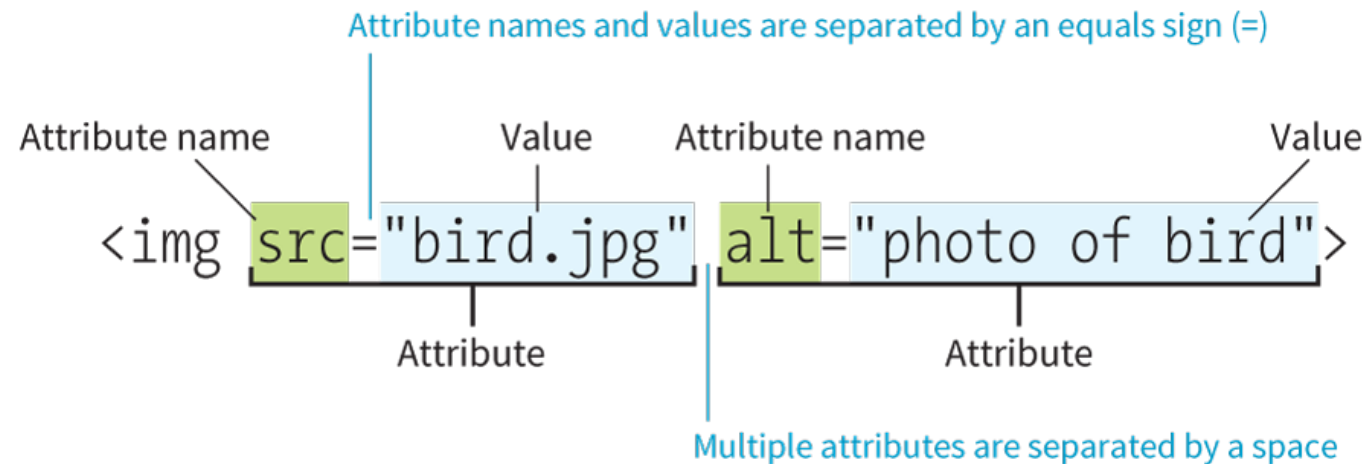
Attributes are instructions that clarify or modify an element. They appear in the opening tag after the element name:

```
<element attribute="value">Content</element>
```

```
<a href="http://oreilly.com">O'Reilly site</a>
```

Attributes (cont'd)

There can be more than one attribute in a tag:



They are separated by spaces and can go in any order.

Attributes (cont'd)

- Most attributes take values, which follow an = sign; some are single descriptive words.
- A value might be a number, word, string of text, URL, or measurement.
- Quotation marks aren't strictly required but are recommended for consistency.
- Single or double quotation marks are okay.
- Attribute names and values are defined in the HTML specification.
- Some attributes are required.

Nesting Elements

Putting elements inside other elements is called **nesting**.
Make sure closing tags don't overlap:

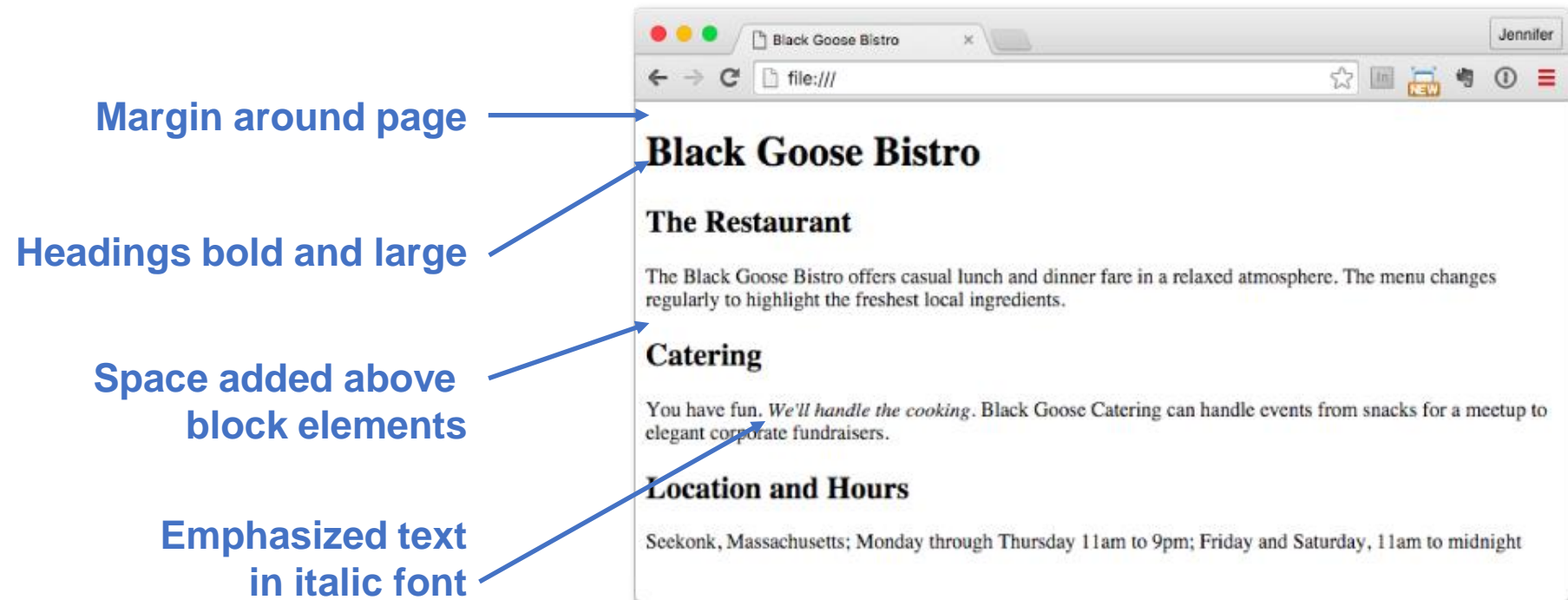
```
<div>  
  <h1>Headline</h1>  
  <p>This is <em>emphasized</em> text.</p>  
</div>
```

Style Sheets

- HTML only describes structure, not presentation
- Presentation is controlled by style sheets (CSS)
- Browsers have their own style sheets (user agent style sheets) with default styles for HTML elements
- You can write your own styles to override the default styles

Style Sheets (cont'd)

The browser parsed the markup and used its built-in style sheet to format the text elements in the example:



Troubleshooting HTML

Small mistakes and missing characters can cause HTML documents to “break.”

Common mishaps:

- Missing closing tag (or / in closing tag)
- Missing closing bracket in a tag
- Missing quotation mark around an attribute value
- Not saving your document before viewing changes in the browser

Troubleshooting (cont'd)

When a slash is omitted, the browser doesn't know when the element ends:

```
<h2>Catering</h2>
<p>You have fun. <em>We'll handle the cooking.<em> Black Goose
Catering can handle events from snacks for a meetup to elegant
corporate fundraisers.</p>
```



g.

Catering

You have fun. *We'll handle the cooking. Black Goose Catering can handle events from snacks for a meetup to elegant corporate fundraisers.*

Location and Hours

Seekonk, Massachusetts;
Monday through Thursday 11am to 9pm;
Friday and Saturday, 11am to midnight

Troubleshooting (cont'd)

A missing end bracket makes the browser interpret all the following characters as part of the tag:

`<h2The Restaurant</h2>`

`<p>The Black Goose Bistro offers casual lunch and dinner fare in a relaxed atmosphere. The menu changes regularly to highlight the freshest local ingredients.</p>`

`<h2The`

Missing subhead

Without the bracket, all the following characters are interpreted as part of the tag, and “The Restaurant” disappears from the page.

BLACK GOOSE

The Black Goose Bistro offers casual lunch and dinner fare in a relaxed atmosphere. The menu changes regularly to highlight the freshest local ingredients.

Catering

You have fun. *We'll handle the cooking.* Black Goose offers a meetup to elegant corporate fundraisers.

Validating Your Documents

Validate a document to make sure that you have abided by the HTML rules and that there are no errors:

- Include the DOCTYPE declaration
- Indicate the character encoding
- Include required attributes
- Don't use non-standard elements
- Don't mismatch tags
- Nest elements correctly (no overlaps)
- Check for typos and other minor errors



LET'S HAVE
SOME
BREAK

In class exercise

1. Join Slack channel
2. Join Github classroom
3. Do the exercise for today's class

Join slack



Use only your @shibaura-it.ac.jp email address

**Create Github account
And join Github classroom**

DEMO by SA

Use only your @shibaura-it.ac.jp email address

END OF CLASS 1

