

SWE 363: Web Engineering & Development

Module 1 **Internet Basics**



Objectives

- ❑ Learn the basics of the Internet and the web
- ❑ Identify and describe the key elements of the internet and the web

Outline

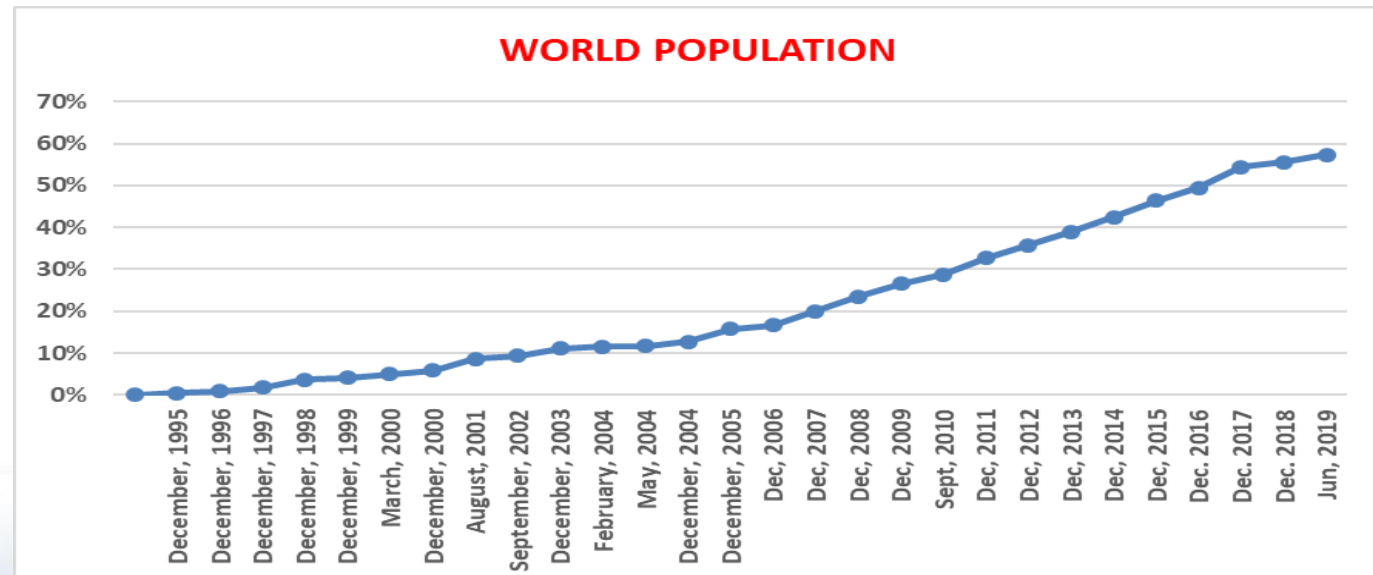
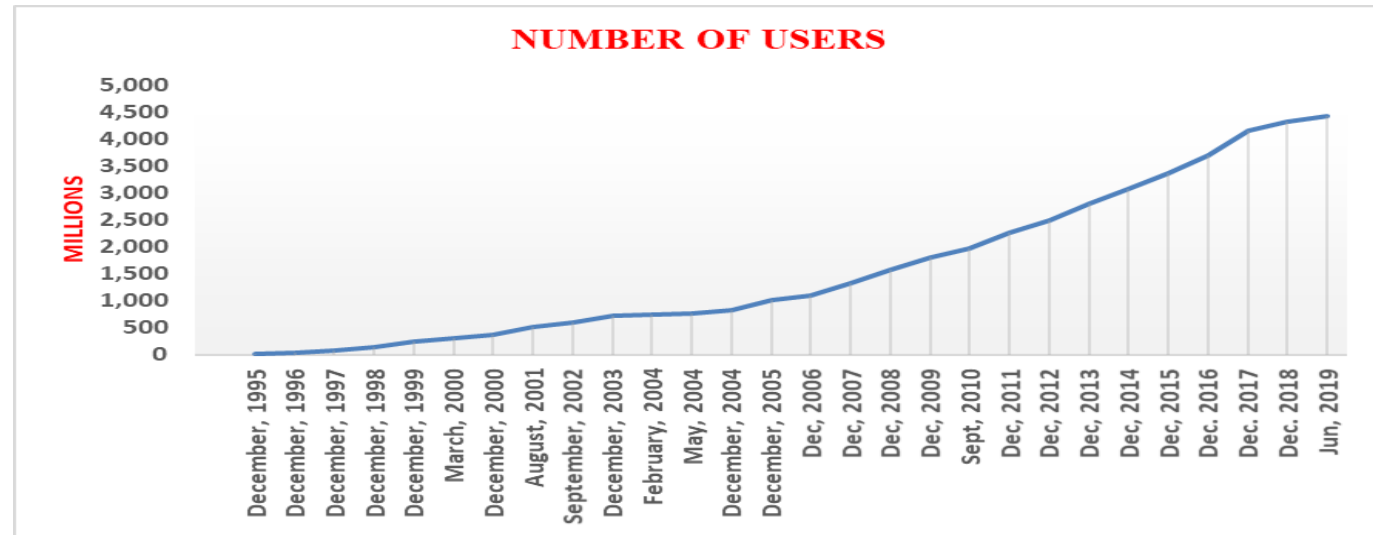
- ❑ Introduction
- ❑ Internet =? World Wide Web
- ❑ Web Browsers
- ❑ Static vs. dynamic websites
- ❑ Web 2.0
- ❑ Internet Protocols
- ❑ Scripting

Used Resources

- ❑ Deitel, Harvey, and Abbey Deitel. Internet and World Wide Web How to Program. Prentice Hall Press, 5th Edition. [**Chapter 1**]
- ❑ Randy Connolly and Ricardo Hoar, “Fundamentals of Web Development”, 2015. [Chapter 1]
- ❑ Other resources:
 - <http://www.internetworldstats.com/>
 - <https://developer.mozilla.org/en-US/docs/Web/HTTP/Caching>

At a glance

- ❑ Growth of the Internet from 1995 till Today



<http://www.internetworldstats.com/>

At a glance...

(Number of websites)

- ❑ Number of websites hits two **BILLION**:
 - A new site is registered every **SECOND**
 - As at **August 2018**, there are *around 2 billion* websites in the world.
 - Asia now accounts for **%50.1** of all Internet users in the world.
- ❑ Every Day”:
 - Over **2 million blog posts** are published on the Internet
 - Over **5.5 billion Google searches**
- ❑ Internet traffic:
 - **51.8%** comes from **bots**,
 - **48.2%** comes from **humans**.
- ❑ The number of **hacked sites** increases year by year !

At a glance...

(E-commerce)

❑ E-Commerce and Conversion Statistics 2017

- The **e-commerce industry** is responsible for about **\$2 trillion** in annual sales.
- *People spend an average of 5 hours per week shopping online.*
- Worldwide B2C e-commerce sales reached **\$1.7 trillion** in 2015, and it is estimated to reach **\$2.35 trillion** by 2018.

❑ For the very first time in history, **global internet advertising** spend is expected to exceed TV advertising spend in 2017.

- ❑ a projected **\$205 billion** Internet ad spend
- ❑ a projected TV ad spend of **\$192 billion**,

Internet live stats

<http://www.internetlivestats.com/>



4,114,502,592

Internet Users in the world



1,945,090,725

Total number of Websites



187,883,020,676

Emails sent [today](#)



4,694,982,142

Google searches [today](#)



4,447,252

Blog posts written [today](#)



545,166,852

Tweets sent [today](#)



5,039,061,951

Videos viewed [today](#)
on YouTube



58,226,094

Photos uploaded [today](#)
on Instagram



96,456,240

Tumblr posts [today](#)

Internet ?= Web

- ❑ The **Internet** is a massive **network of networks**, a networking infrastructure.
 - **Comprises**: network of computers, copper wires, fiber-optic cables & wireless networks
 - **Governed by**: Internet Protocol (**IP**)
 - **Dependency**: This is the base, independent of the World Wide Web
 - **Nature**: Hardware

- ❑ The **World Wide Web**, or simply **Web**, is a way of accessing information over the medium of the Internet.
 - It is an **information-sharing model** that is built on top of the Internet.
 - **Comprises**: files, folders & documents stored in various computers
 - **Governed by**: Hyper Text Transfer Protocol (**HTTP**)
 - **Dependency**: It depends on Internet to work
 - **Nature**: Software



Internet...?

- ❑ A global **heterogeneous network** that uses
 - **transmission media** (copper, fiber, wireless, etc.), **special purpose devices** (routers, switches, etc.), **network operating systems** (NOS) and **applications software** (email, browsers, etc)

- ❑ Goal: **to provide connectivity** between machines/users
 - Share resources
 - Increase reliability and availability
 - Collaborate (email, distributed computing, etc)
 - Access remote information

- ❑ Internet applications can be separated into the following types:
 - online media, online information search, online communications, online communities, online entertainment, e-business, online finance and other applications.



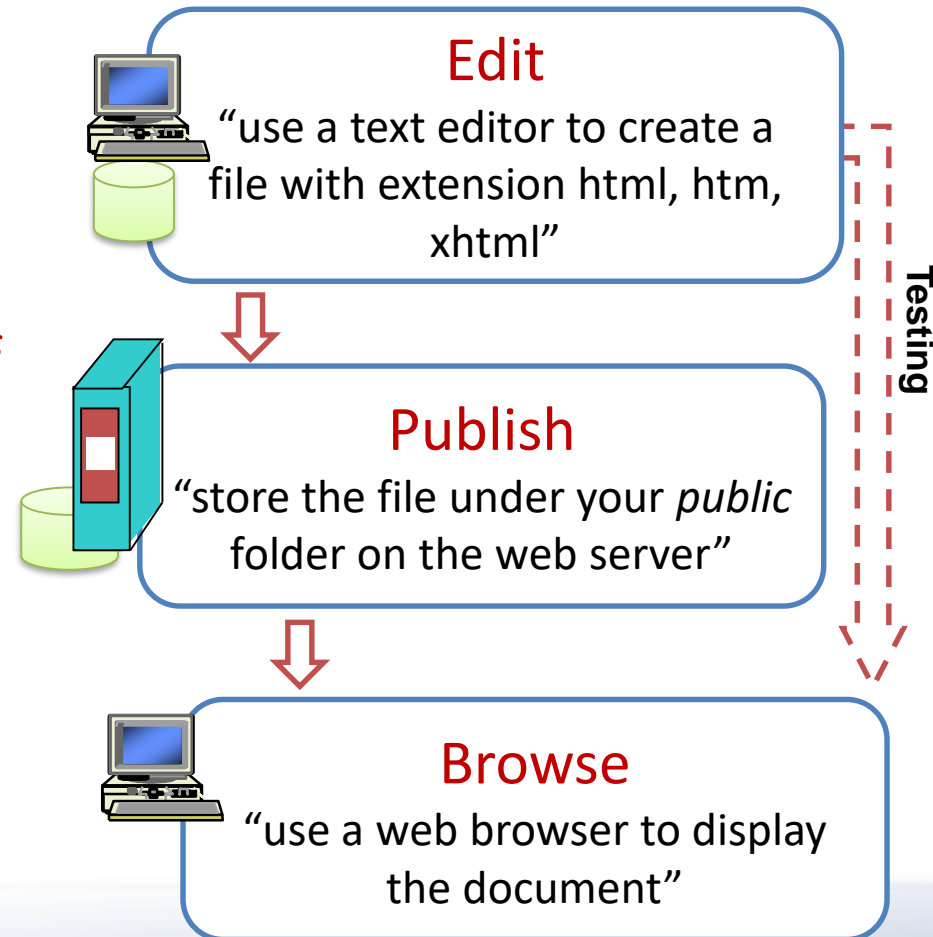
Web..?

- ❑ WWW also utilizes **browsers**, such as **Internet Explorer** or **Firefox**, to access Web documents called **Web pages** that are linked to each other via **hyperlinks**.
- ❑ **Web site** is a collection of **linked Web pages** that has a common theme or focus.
 - A website is **hosted on one or more servers** and can be accessed by visiting its **homepage**.
 - The site's **home page** is the main page that all of the pages on a particular web site are organized around and link back to it.
- ❑ Each web page on the Internet is associated with a unique **URL** (Uniform Resource Locator) .
 - **URL** specifies the address (i.e., location) of the web page displayed in the browser window.
 - URLs usually begin with **http://** for example.

Web Documents

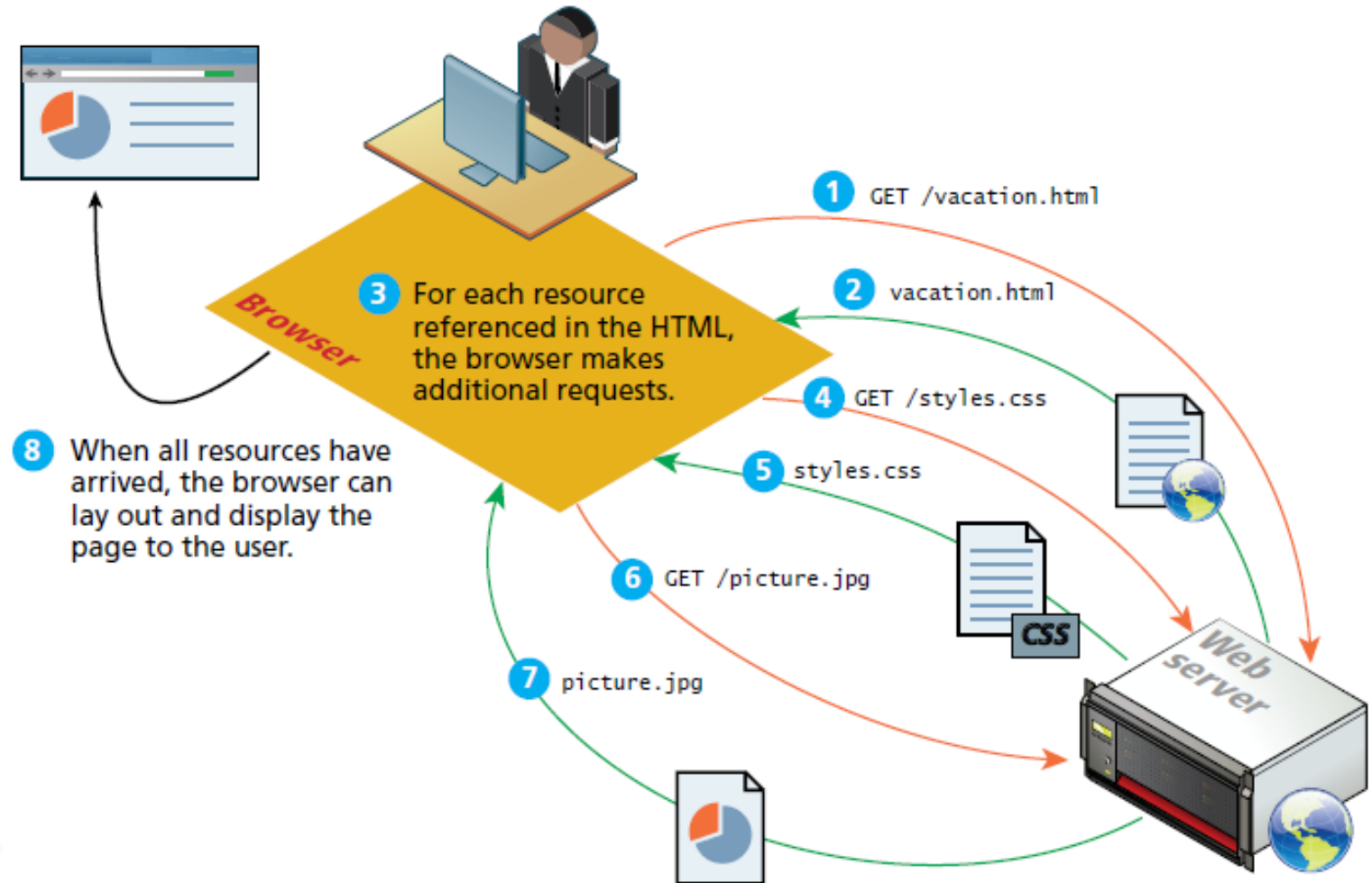
- ❑ A *web document* (page) is a **text file** that contains **text** marked up with **elements** that are delimited by **tags**; other media objects and applications are embedded
- ❑ Web pages are stored on **Web Servers** and are requested and displayed by **Web Browsers**
- ❑ When a page is viewed by a web browser, it shows text, images, lists, headings, tables, forms, hyperlinks to other online documents, etc.

Development steps



Fetching a Web Page

- ❑ Browser parsing HTML and making subsequent requests



Web Browsers

- ❑ A **Web Browser** is a software program that enables you to view Web pages on your computer (is called *rendering the webpage*)
- ❑ **Browsers** are available in **many versions** and on **many different platforms** (Microsoft Windows, Apple Macintosh, Linux, UNIX, etc.)
 - Adding features to each new version that sometimes result in **cross-platform incompatibility** issues.
- ❑ Ensuring a **consistent look and feel** on **client-side browsers** is one of the great challenges of developing web-based applications.
 - No standard exist to which software vendors must adhere when creating web browsers.
- <http://caniuse.com/> - to list of features covered by each browser.
- <http://html5test.com/> - to score each browser based on its support for latest features

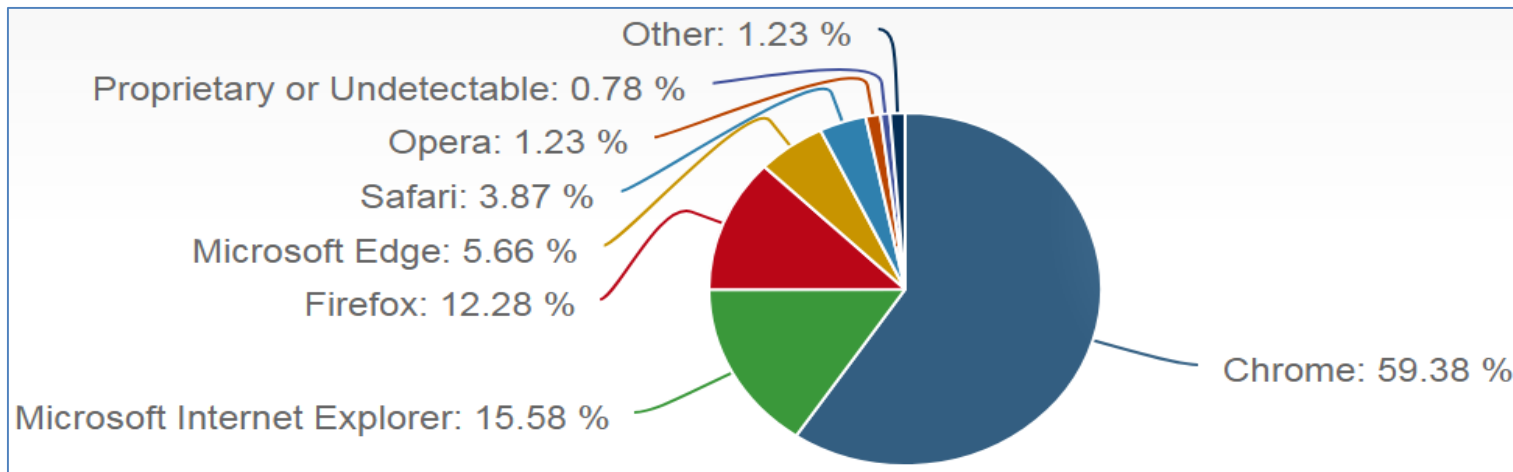
Web Browsers

Cross-Browser Compatibility

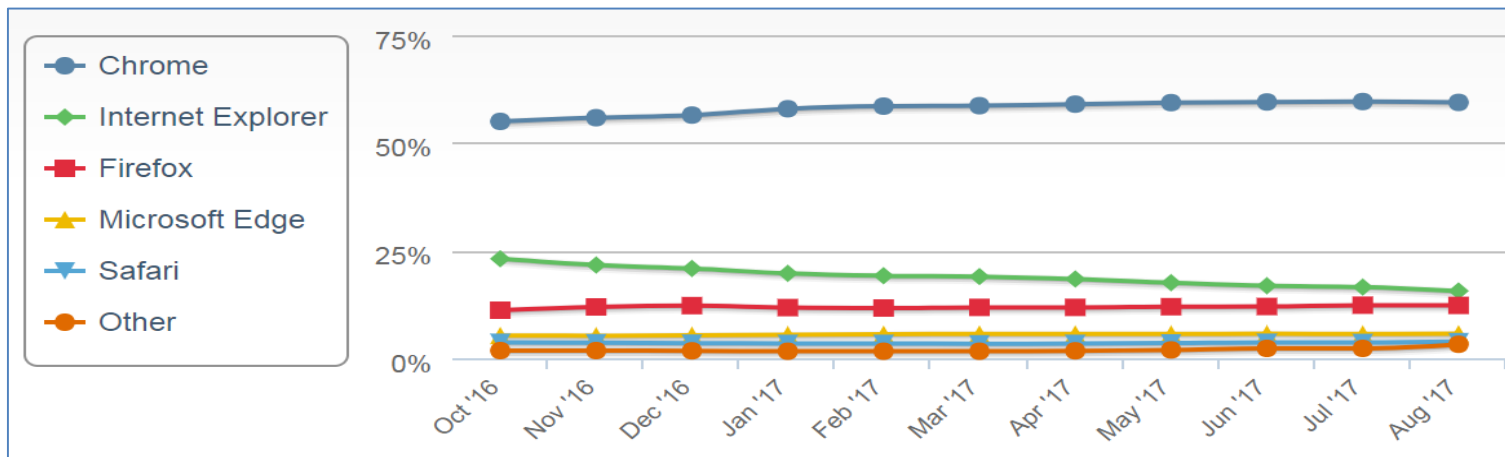
- ❑ Great diversity of web browsers
 - Differ in **functionality**, **performance**, and **supported features** for HTML, JavaScript, CSS, etc..
- ❑ **Cross-browser** refers to the ability of a **website**, **web application**, **HTML construct** or **client-side script** to:
 - function in environments that provide its required features
 - withdraw gracefully when features are absent or lacking
- ❑ Cross-Browser Challenges:
 - Many different browsers and Many different versions
- ❑ A **cross-browser compatible** web page will look more or less the same in all of the existing Web browsers
 - Cross-browser compatibility **is difficult to achieve**
 - **W3C** (www consortium) is working toward the goal of a **universal client-side platform**.

Web Browsers

Desktop Browser Market Share

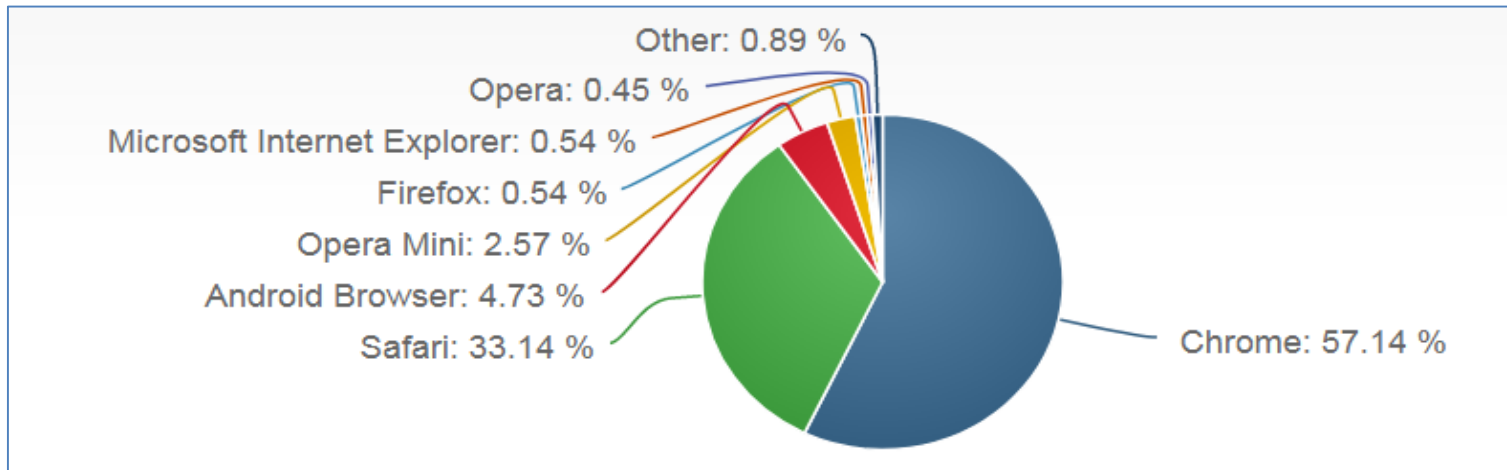


July, 2019
Chrome: 71.05%
Firefox: 9.52%
Safari: 5.41%

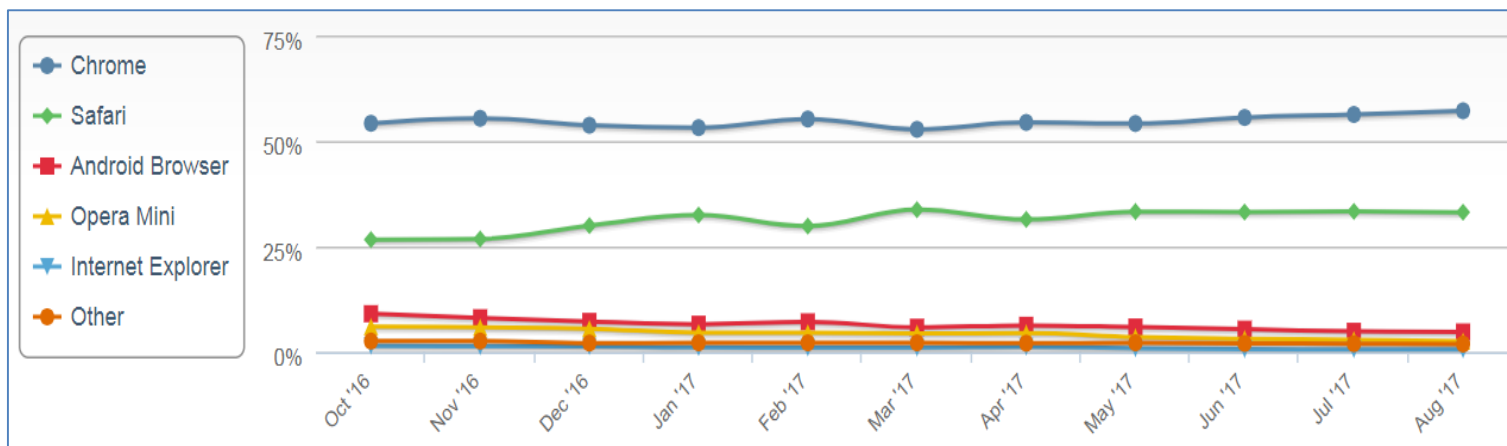


Web Browsers

Mobile/Tablet Browser Market Share



July, 2019
Chrome: 62.21%
Safari: 27.99%



Source: <https://www.netmarketshare.com/>

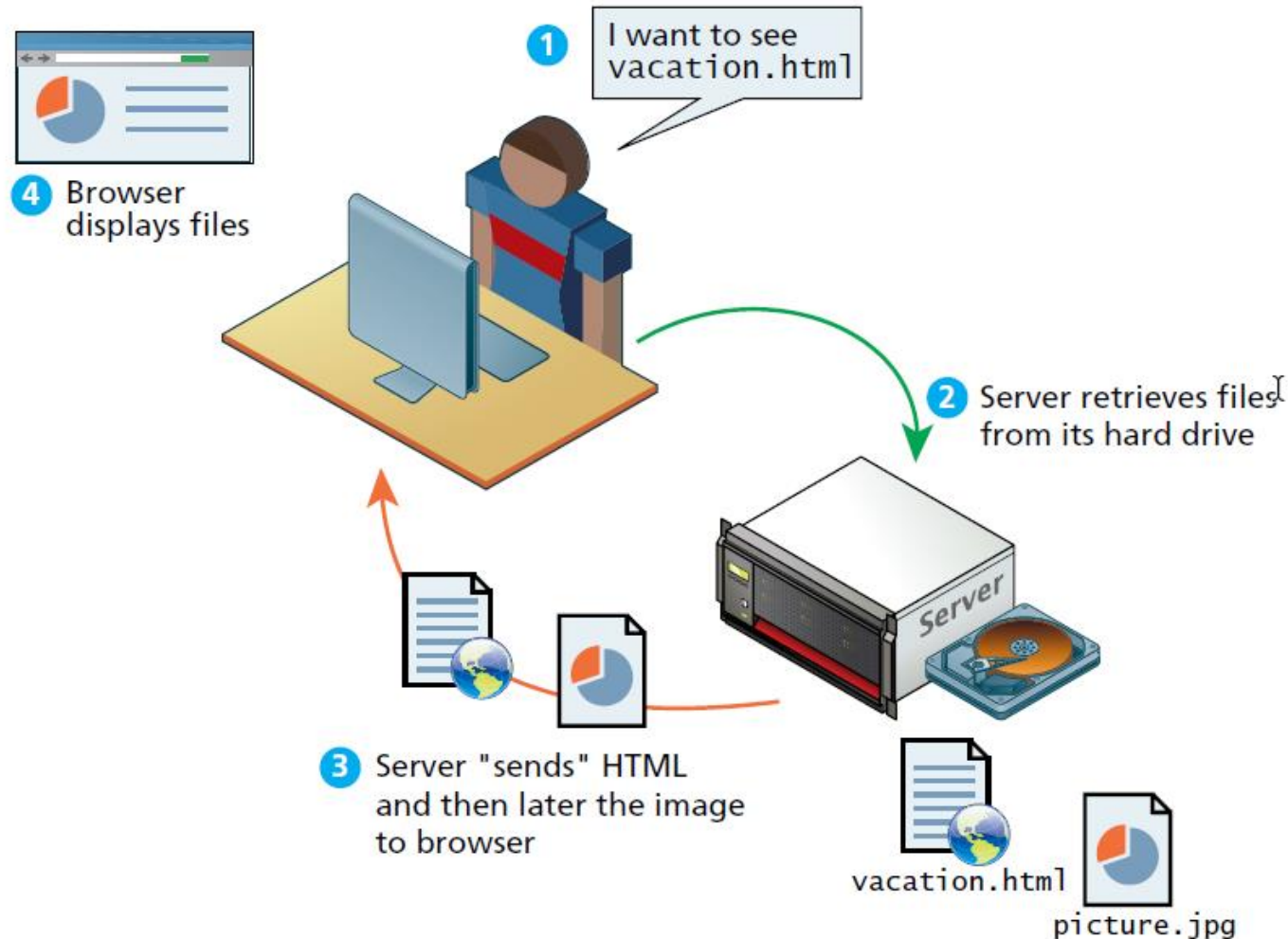
Static and Dynamic Websites

- ❑ A website can be of two types: **Static** and **Dynamic** Website

- ❑ **Static website** is the basic type of website that is easy to create.
 - It is usually written in plain HTML and what is in the code of the page is what is displayed to the user.
 - No web programming and database design are needed
 - The codes are **fixed** for each page so the information contained in the page does not change and it looks like a printed page.

Static and Dynamic Websites

Static website



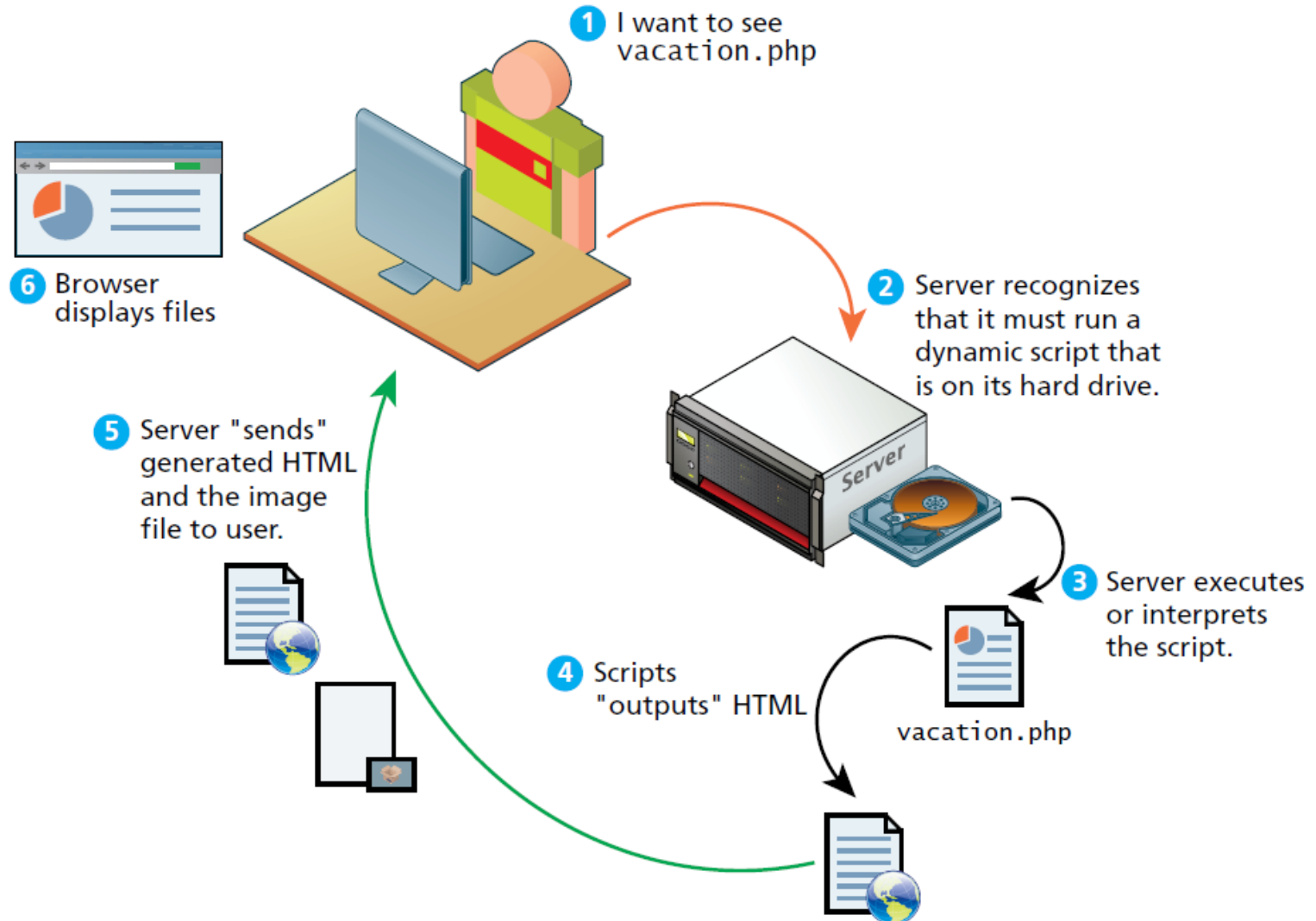
Static and Dynamic Websites..

Dynamic website

- ❑ **Dynamic website** is a collection of dynamic web pages whose **content changes dynamically**.
 - is written using a server-side scripting language such as PHP, ASP, etc.
 - The **content is called** in by the scripting language from other files or from a database depending on actions taken by the user.
 - Therefore, when you alter or update the content of the database, the content of the website is also altered or updated.
- ❑ Dynamic website uses **client-side scripting** or **server-side scripting**, or both to generate dynamic content.

Static and Dynamic Websites..

Dynamic website



Interactive website

- ❑ An **interactive website** means a website that allows it's users to communicate and interact with it.
- ❑ An interactive website is a **two-way process**.
 - A user does something then the website responds.
 - A user asks a question - the website answers. A user enters some information - the website saves it.
- ❑ An **interactive website** is an internet page that uses different kinds of software to create a rich, **interactive experience** for the user
 - **Engagement** is the Key
- ❑ Some of the most common types of interactive websites include Blogs, Forums , Wikis, and Social networks.
 - >> allow users to change the way the website displays, play games, interact with friends online, and perform a host of tasks.

Web 2.0

It refers to World Wide Web websites that

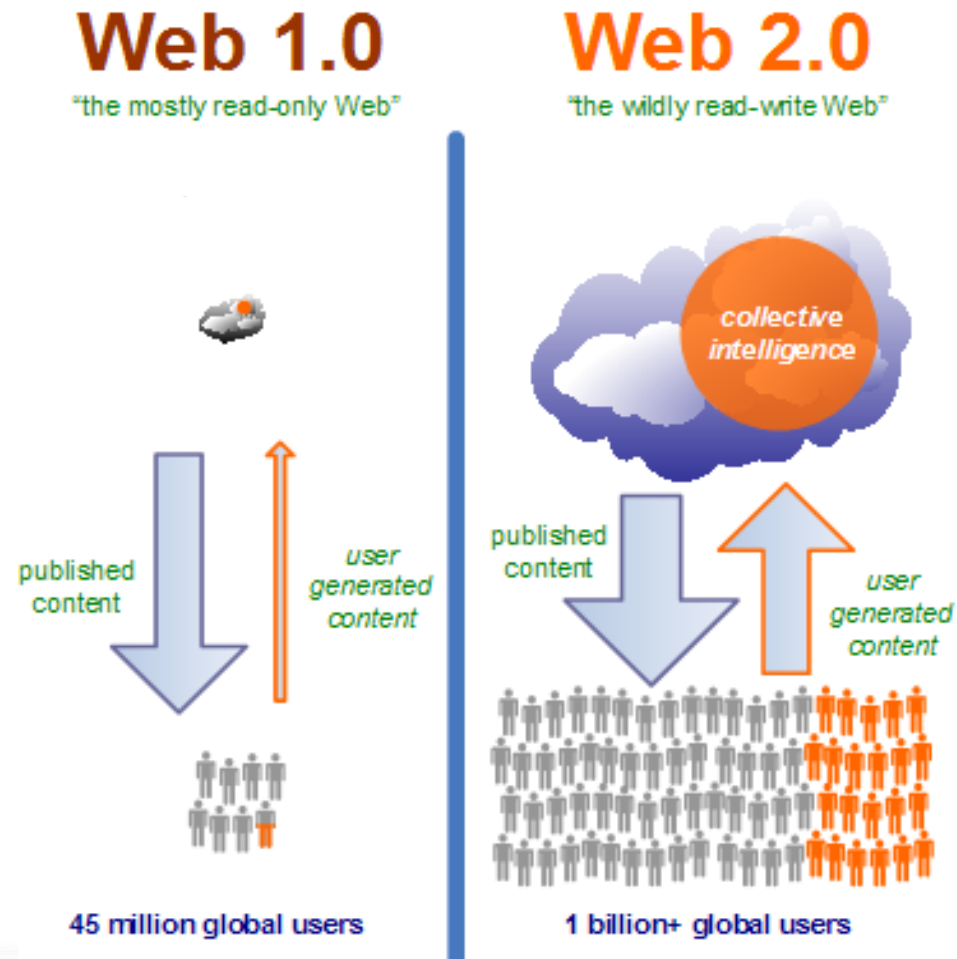
- emphasize user-generated content, usability (ease of use, even by non-experts), and
- The ability of a website can work well with other products, systems, and devices for end users.

What is Web 2.0

- ❑ Web 1.0 (the state of the web through the 1990s and early 2000s) was focused on a relatively small number of companies and advertisers producing content for users to access (some people called it the “brochure web”).
- ❑ Web 2.0 involves the users
 - not only do they can create content, but they help organize it, share it, remix it, update it, etc.
- ❑ For simplicity you can:
 - look at Web 1.0 as a lecture, a small number of professors informing a large audience of students
 - In comparison, Web 2.0 is a conversation, with everyone having the opportunity to speak and share views

Web 1.0 vs. Web 2.0

- ❑ In just a few short years, we have gone from a Web that was primarily "*read only*" to one where
 - Creating **content** is almost as easy as consuming it.
 - Writing and publishing in the forms of **blogs**, **wikis** and **podcasts** and many other such tools is available to everyone.
 - Connecting not just to **content** but to **people**, **ideas** and **conversations** as well.



What is Web 2.0

- ❑ **Web 2.0.** This term had two meanings, one for users and one for developers.
- ❑ For the **users**, Web 2.0 lets people **collaborate** and **share** information online
 - the web becomes as a platform to create **collaborative, community-based** sites
- ❑ For **software developers**, Web 2.0 also referred to a change in the paradigm of how dynamic websites are created.
 - Programming logic, which previously existed only on the server, began to migrate to the browser.
- ❑ Web 2.0 offers **many interactive software choices**, many of which have become household names. some examples of Web 2.0:
 - online banking
 - social networking sites, blogs, wikis
 - online radio, video hosting, digital photo processing, photos sharing
 - geomatics and mapping services
 - Etc.

Web 2.0

Architecture of Participations

- ❑ Web 2.0 embraces an **architecture of participation**
 - a design that encourages **user interaction** and **community contributions**.

- ❑ Using **collective intelligence**
 - a large diverse group of people will create smart ideas
 - communities collaborate to develop software that many people believe is **better** and **more robust** than proprietary software.

- ❑ **Rich Internet Applications** (RIAs) are being developed using technologies (such as Ajax) that have the look and feel of desktop software, enhancing a user's overall experience.

Web 2.0

Search Engines and Social Media

- ❑ Optimized way to find information on the web sites
 - people are tagging (i.e., labeling) web content by **subject** or **keyword** in a way that helps anyone locate information **more effectively**.

- ❑ Semantic Web:
 - In the future, computers will learn to understand the **meaning** of the data on the web (**Web 3.0**)

- ❑ Social Media

- ❑ **Ajax** is one of the premier Web 2.0 software technologies
 - It helps Internet-based applications perform like desktop applications
 - A difficult task, given that such applications suffer transmission delays as data is shuttled back and forth between your computer and servers on the Internet.

Web 2.0 applications

Blogs

- ❑ A frequently updated website, automatically archived.
 - Easily updated using any web browser connected to the Internet.
 - An online journal or newsletter, consisting of personal thoughts & commentary
 - Blogs include links to articles and websites.
 - Blogs can be authored by an individual or by multiple authors.
- ❑ Most blogs are interactive, allowing visitors to leave comments and even message each other via widgets on the blogs
 - Such interactivity that distinguishes them from other static websites
- ❑ Blogs types:
 - Personal blogs
 - Cooperate and organizational blogs
 - etc.

Web 2.0 applications

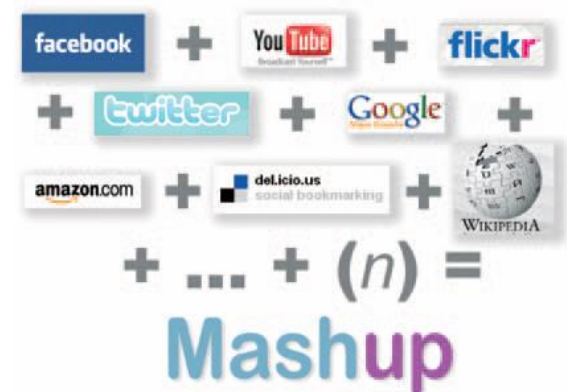
Wikis

- ❑ A **wiki** is a web application that allows **users** to **add/modify content from web browser**.
 - A wiki enables communities of editors and contributors to write documents collaboratively.
- ❑ A wiki is run using **wiki software**, known as a wiki engine.
 - A wiki engine is a type of content management system, but it differs from most other such systems, including **blog software**, in that
 - the content is created without any defined owner or leader,
 - wikis have little implicit structure, allowing structure to emerge according to the needs of the users.
- ❑ Example:
 - <https://www.wikipedia.org/>

Web 2.0

Mashups

- ❑ A **mashup** is a **Web page** or **application** created by combining information or capabilities from more than one existing source to deliver new functions & insights.
 - **Example:** a huge range of tools to combine Googlemaps data with other source.
- ❑ The main characteristics of a mashup are **combination**, **visualization**, and **aggregation**.
 - It is important to make existing data more useful, for personal and professional use.
- ❑ **Example:** <https://www.trendsmap.com/>



What mashups do?

- ❑ Rapid creation

- To foster innovation by unlocking and remixing information in ways not originally planned for.

- ❑ Reusing exiting capabilities, but delivering new functions + insights

- ❑ To increase agility by supporting dynamic assembly and configuration of applications

- ❑ To reduce development costs through integration and reuse

- ❑ Requires less technical skills

Internet Protocols

Internet Protocols

- ❑ A **protocol** is a set of rules that partners (in communication) use when they communicate.
- ❑ The internet protocols have been implemented in every operating system, and make fast web development possible.
- ❑ TCP/IP (**Transmission Control Protocol / Internet Protocol**) is one of the essential internet protocols.

Internet Protocols...

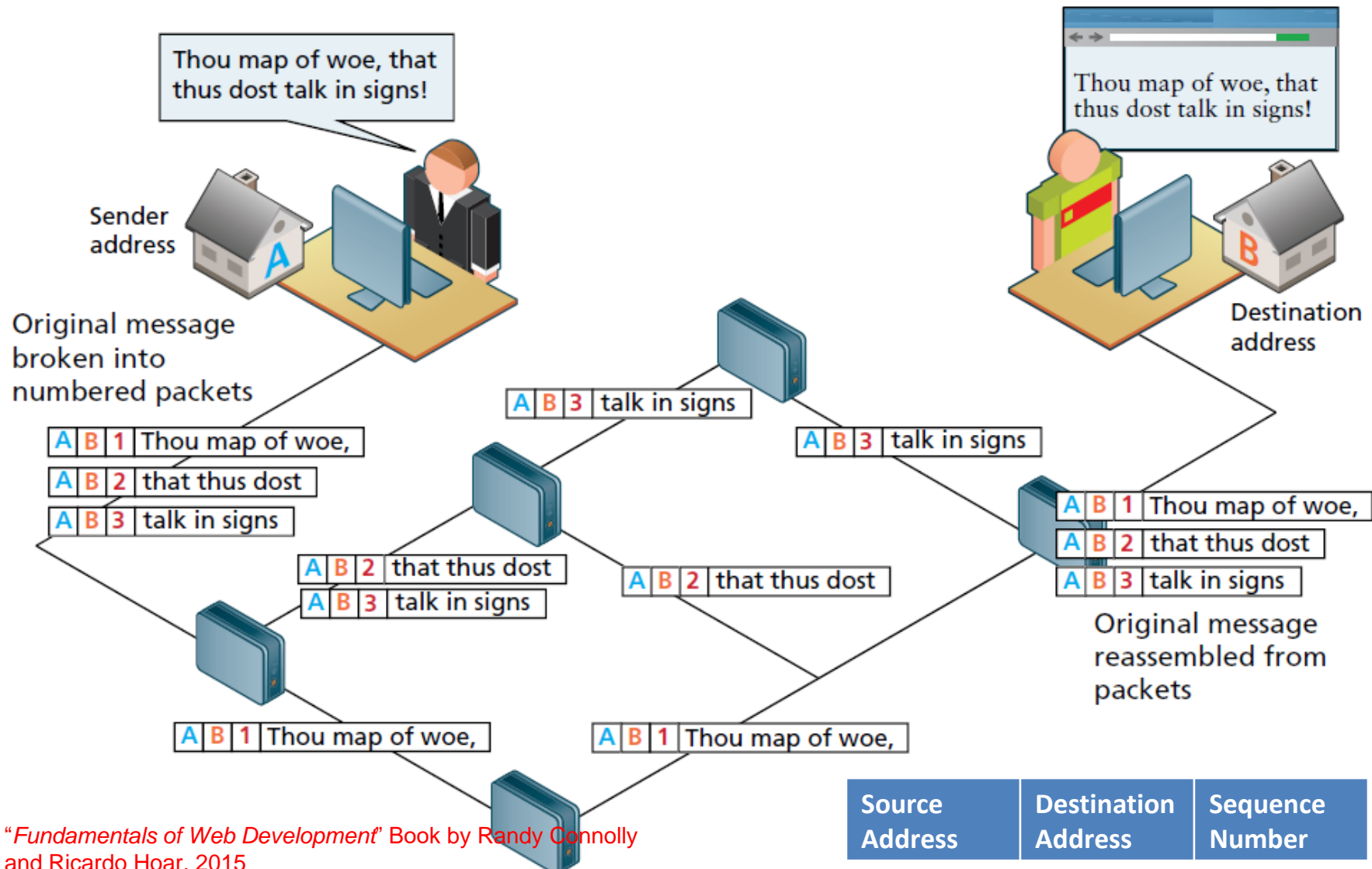
- ❑ **TCP/IP** is a set of rules (protocols) governing **communications** among all computers on the Internet
- ❑ TCP/IP dictates how **information should be packaged** (turned into bundles of information called **packets**), **sent**, and **received**, as well as how to get to its destination. >> **Packet Switching**
- ❑ **TCP** ensured that messages were properly routed from sender to receiver and that they arrived in correct order.
- ❑ **IP** dictates the logistics of packets sent out over networks; it tells packets where to go and how to get there.
- ❑ Each computer on the Internet has a unique **IP address**.

Packet switching

- ❑ How to allow multiple users to send and receive information simultaneously over the same communications paths?
- ❑ The network operated with a technique called **packet switching**
- ❑ **Packet Switching** refers to protocols in which the **data are divided into packets before they are sent.**
 - Each packet is then transmitted individually and can even follow different routes to its destination.
 - Once all the packets forming a message arrive at the destination, they are recompiled into the original message.
 - The packets contained **address, error-control** and **sequencing information.**
- ❑ No centralized control- If a portion of the network failed, the remaining working portions would still route packets from senders to receivers over **alternative paths** for reliability.

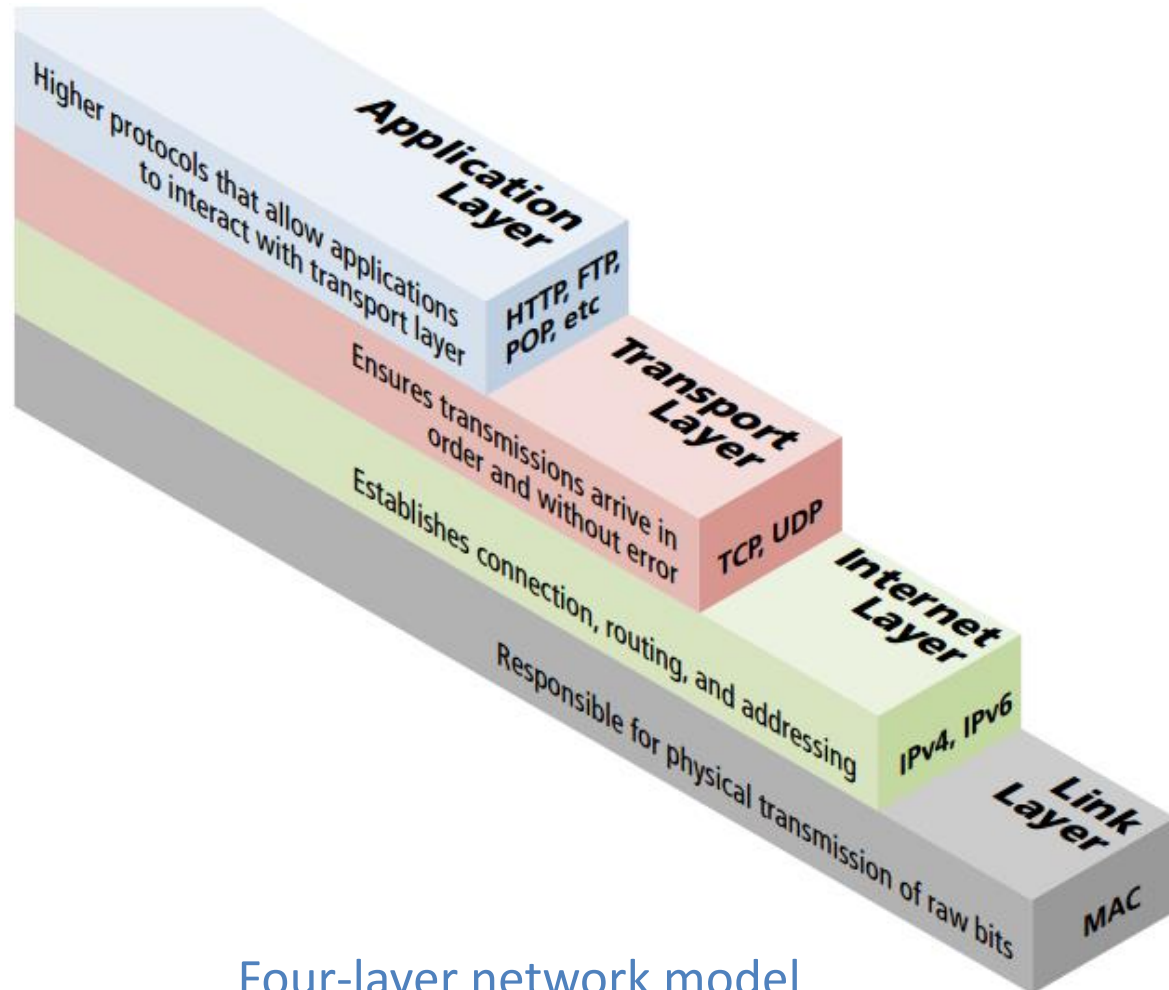
Packet switching..

An example



TCP/IP

- ❑ The TCP/IP Internet protocols were originally abstracted as a **four-layer network model**.
- ❑ Layers communicate information up or down one level
- ❑ Lower layers handle the more fundamental aspects of transmitting signals through networks,
- ❑ The higher layers concern how a client and server interact.

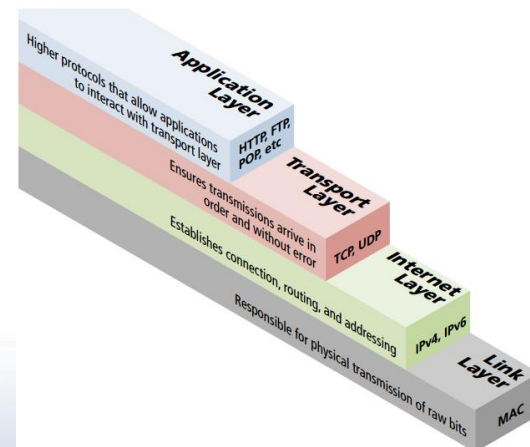


Four-layer network model

TCP/IP

Link Layer

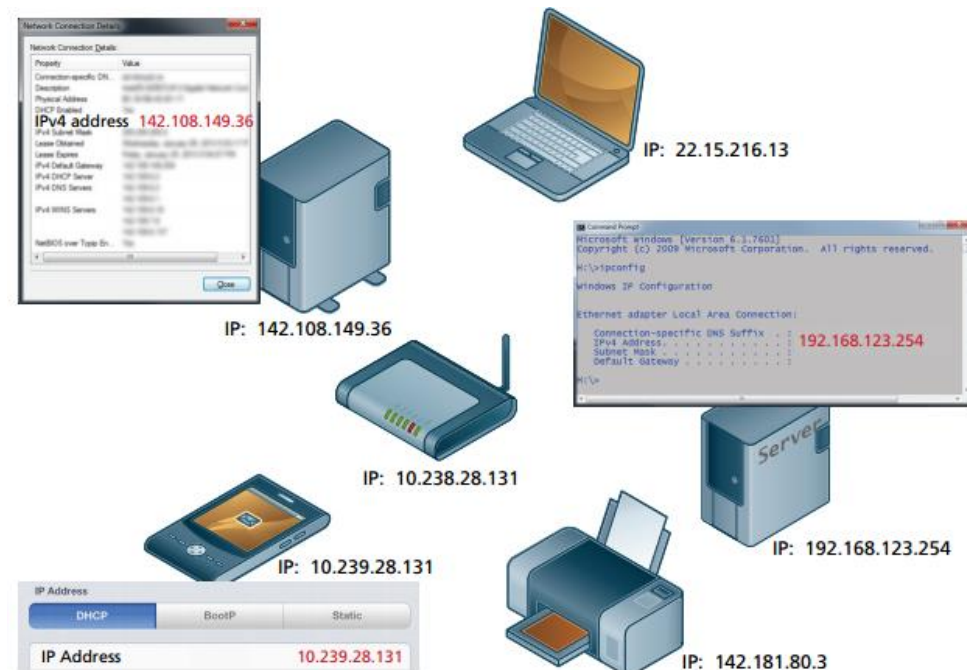
- ❑ The **link layer** is the lowest layer, responsible for both the physical transmission across media (wires, wireless) and establishing logical links.
- ❑ It handles issues like packet creation, transmission, reception, error detection, collisions, line sharing, and more.
- ❑ The one term here that is sometimes used in the Internet context is that of **MAC (media access control)** addresses.
 - These are unique 48- or 64-bit identifiers assigned to network hardware and which are used at the physical networking level.



TCP/IP

Internet Layer

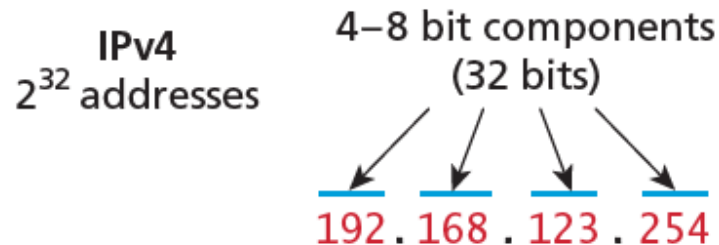
- ❑ The Internet uses the **Internet Protocol (IP)** addresses to identify destinations on the Internet.
- ❑ The **Internet layer** (called the IP Layer) routes packets between communication partners across networks.
- ❑ Every device connected to the Internet has (**unique identification**) an **IP address**, which is a numeric code that is meant to uniquely identify it.
 - For example, **216.58.216.164** is one of the **IP address** for Google
- ❑ There are two types of IP addresses: **IPv4** and **IPv6**.



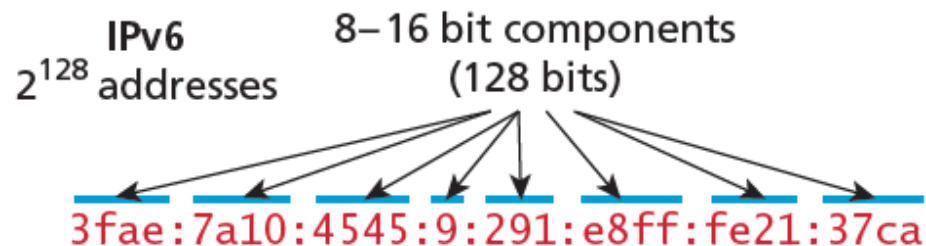
TCP/IP

Internet Layer

- ❑ IPv4 addresses are the IP addresses from the original TCP/IP protocol.
- ❑ In IPv4, 12 numbers are used (implemented as **four 8-bit integers**), Each part of the address is a number ranging from 0 to 255, and each part is separated from the previous part by period



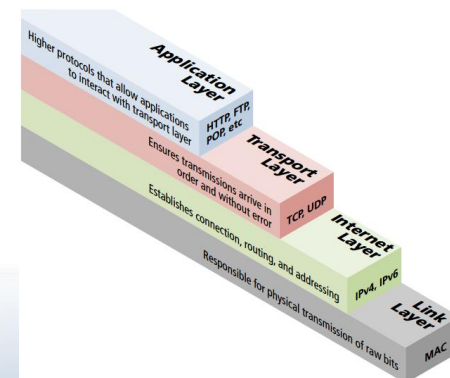
- ❑ IPv6 uses eight 16-bit integers for 2^{128} → the 16-bit integers are normally written in hexadecimal, due to their longer length.



TCP/IP

Transport Layer

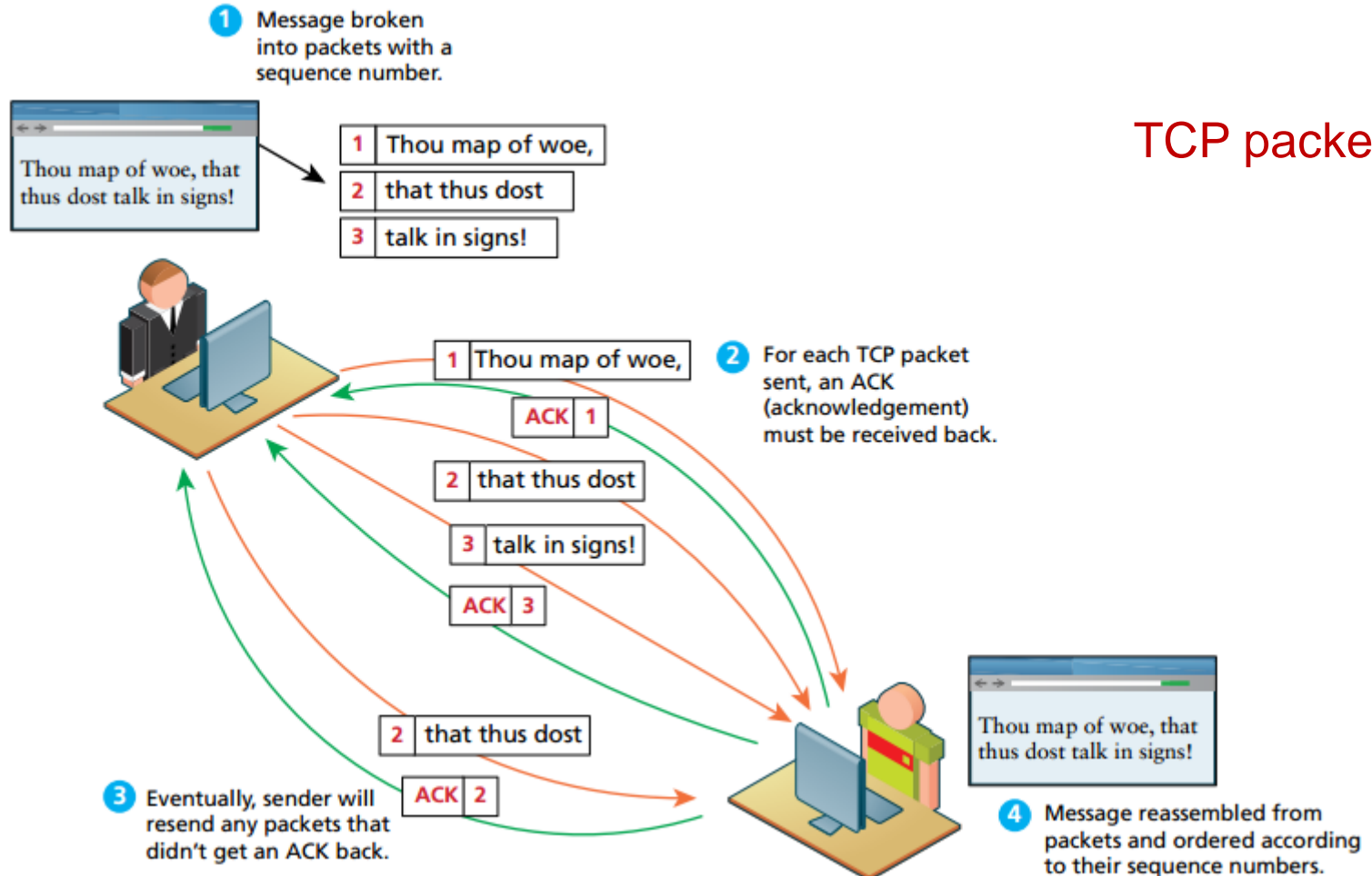
- ❑ The **transport layer** ensures transmissions arrive in order and without error.
- ❑ To have a guarantee that messages sent will arrive and in order, a few mechanisms are used .
 - First, the **data is broken into packets** formatted according to the **Transmission Control Protocol** (TCP). Each data packet has a header that includes a sequence number, so the receiver can put the original message back in order, no matter when they arrive.
 - Secondly, **each packet is acknowledged** back to the sender so in the event of a lost packet, the transmitter will realize a packet has been lost since no **ACK** arrived for that packet. That packet is retransmitted, and although out of order, is reordered at the destination.



TCP/IP

Transport Layer

TCP packets



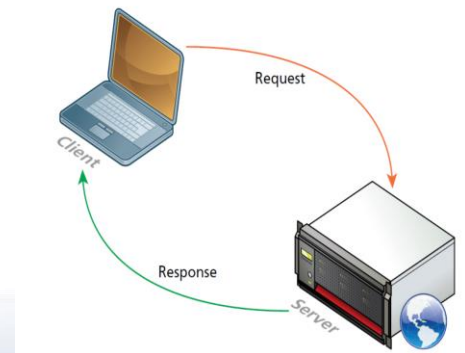
TCP/IP

Application Layer

- ❑ The **application layer** includes protocols familiar to most web developers.
- ❑ **Application layer** protocols implement process-to-process communication.

- ❑ There are many application layer protocols. A few that are useful to web developers include:
 - **HTTP**. The Hypertext Transfer Protocol is used for web communication.
 - **SSH**. The Secure Shell Protocol allows remote command-line connections to servers.
 - **FTP**. The File Transfer Protocol is used for transferring files between computers.
 - **POP/IMAP/SMTP**. Email-related protocols for transferring and storing email.
 - **DNS**. The Domain Name System protocol used for resolving domain names to IP addresses.

- ❑ Hyper Text Transfer Protocol is the main access protocol of the World Wide Web
- ❑ It is designed to enable communications between clients and servers (to send information over the web) for non-secure data transmissions.
- ❑ It works as a request-response protocol between a client (a web browser) and server (an application on a computer that hosts a web site).
 - A web client (i.e. Internet browser on a computer) sends a request to a web server to view a web page.
 - The web server receives that request and sends the web page information back to the web client.



- ❑ Two commonly used methods for a **request-response** between a client and server.
- ❑ A **get** request typically **gets (or retrieves)** information from a **specified resource (server)**,
 - such as an HTML document, an image or search results based on a user-submitted search term.
- ❑ A **post** request typically **posts (or sends)** data to be processed to a **specified resource**
 - Common use: is to send form data or documents to a server.

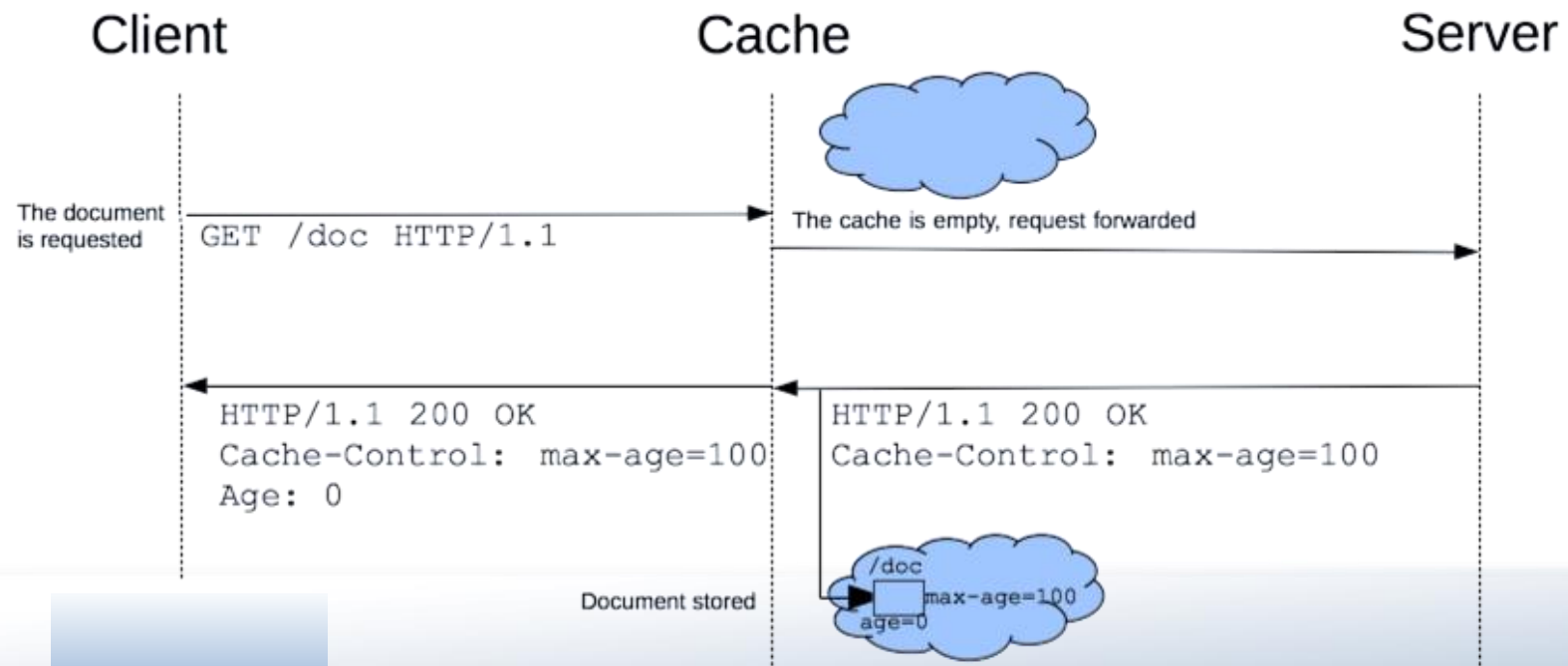
HTTP

HTTP get and post Requests

- ❑ A **get** request may be initiated by
 - submitting an HTML form whose method attribute is set to "get",
`<form action="/action_page.php" method="get">`
 - typing the URL (**possibly containing a query string**) directly into the browser's address bar.
www.google.com/search?q=KFUPM
- ❑ The server uses data passed in a query string to **retrieve an appropriate resource** from the server and **sends a response to the client**.
- ❑ A **get** request typically **limits the query string** (i.e., everything to the right of the ?) to a specific number of characters, so it's often necessary to send **large amounts of information** using the **post** method.
- ❑ The post method is also sometimes **preferred** because **it hides the submitted data** from the user by embedding it in an HTTP message.
 - A **post request** sends **form data** as **part of the HTTP message**, not as part of the URL.

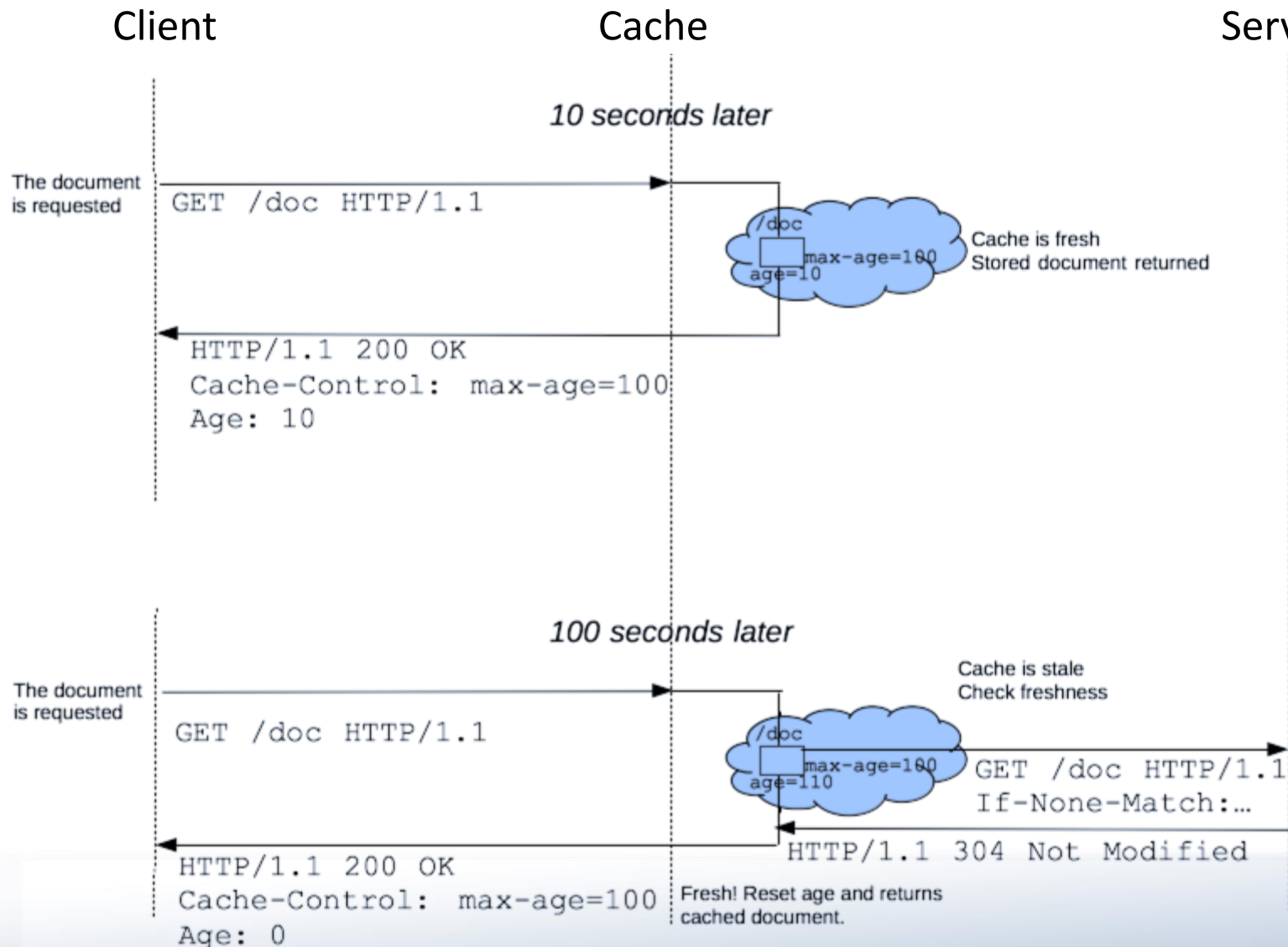
HTTP Caching

- HTTP caching techniques are always **associates with the client side**.
 - Browsers often **cache** (save on disk) **recently viewed** web pages for quick reloading.
 - If there are **no changes** between the version stored in the cache and the current version on the web, this speeds up the user's browsing experience.



- ❑ There are **two** major cache controlling **mechanisms**.
 - **Time** based caching
 - **Content** based caching
- ❑ An HTTP response can indicate the **length of time** for which the content remains “fresh.”
 - If this amount of time **has not been reached**, the browser can avoid another request to the server.
 - Otherwise, the browser requests the document from the server.
- ❑ Similarly, there’s also the “not modified” HTTP response, indicating that the file content **has not changed since it was last requested**

HTTP Caching...

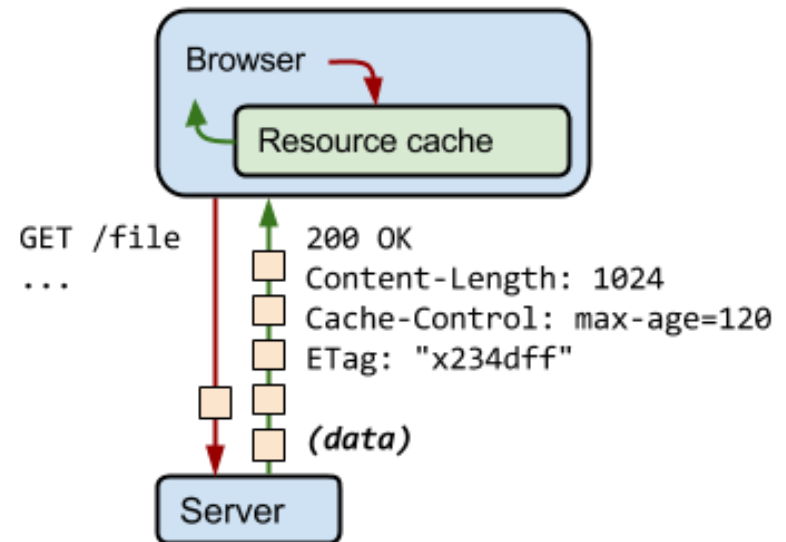


HTTP

Caching...

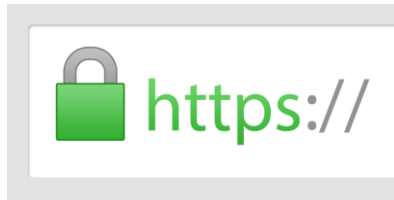
- ❑ Every browser ships with an implementation of an **HTTP cache**
- ❑ When the server returns a response, it also emits a collection of HTTP headers, describing its content-type, length, caching directives, validation token, and more.

>> e.g. the server returns a **1024-byte** response, instructs the client to cache it for up to **120** seconds, and provides a validation token ("**x234dff**") that can be used after the response has expired to check if the resource has been modified.



Authentication a Problem?

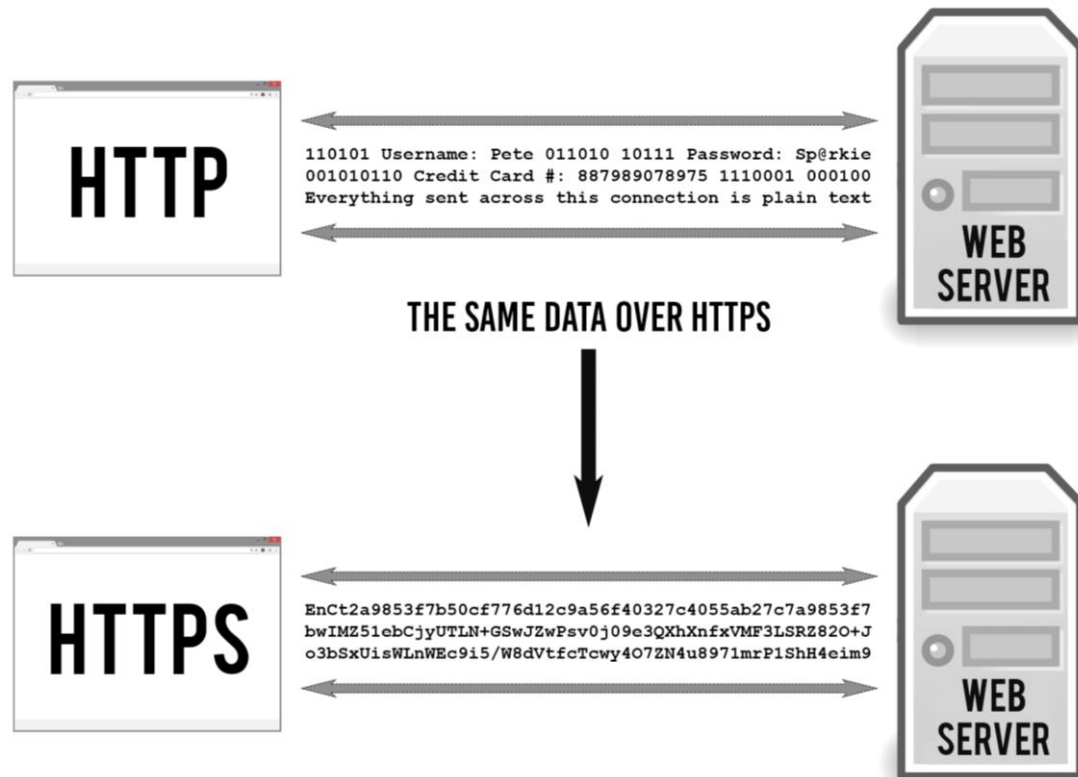
- ❑ Using different ways you, or your browser, can be tricked into thinking you are connected to a familiar website, when in reality you are connected to an attacker trying to steal your credentials.
- ❑ For example, you are visiting Facebook.
 - If your **DNS** is manipulated by an attacker, then your browser will connect to that attacker's IP address thinking that it is Facebook.
 - If there is an attacker in **control of a router** on the path between you and Facebook's servers, the attacker may intercept your traffic, and respond to it themselves pretending to be Facebook. Since a real reply from Facebook would likely come through their router anyway, it can be very difficult to know **you've received a forgery**.
 - If you see a link to Facebook and click it without realizing that it actually says, for example, "faceboook.com" (with an extra "o"), you may end up connected to an attacker who registered that **slight-typo domain** on purpose for malicious activities.
- ❑ These are all forms of a **"phishing" attack**, and they happen every day.



Hypertext Transfer Protocol Secure

- ❑ This lock is very important. It indicates that your browser has used HTTPS to properly **secure** and **authenticate** your connection with a website.
- ❑ HTTPS creates a **secure channel** over an **insecure network**.
 - This ensures **reasonable protection** from **eavesdroppers** and **man-in-the-middle** attacks, provided that the server certificate is verified and trusted.
- ❑ HTTPS has three main goals:
 - **Privacy**: Encrypting data such that anything in-between your browser and the website cannot read your traffic.
 - **Integrity**: Ensuring that the data received on either end has not been altered unknowingly along the way.
 - **Authentication**: Proving that the website your browser is talking to is who they say they are.

- ❑ HTTPS keeps your stuff secret by encrypting it as it moves between your browser and the website's server. This ensures that anyone listening in on the conversation can't read anything.



- ❑ HTTPS is the standard for transferring encrypted data on the web.
 - It combines HTTP with the Secure Sockets Layer (SSL) and the more recent Transport Layer Security (TLS) cryptographic schemes for securing communications and identification information over the web.

- ❑ Essentially, you need three things to encrypt data:
 1. The data you want to encrypt
 2. A unique encryption key (just a long string of random text)
 3. An encryption algorithm (a math function that “garbles” the data)

- ❑ There are two basic techniques for encrypting information:
 - symmetric encryption (also called secret key encryption) and
 - asymmetric encryption (also called public key encryption.)

To read more:

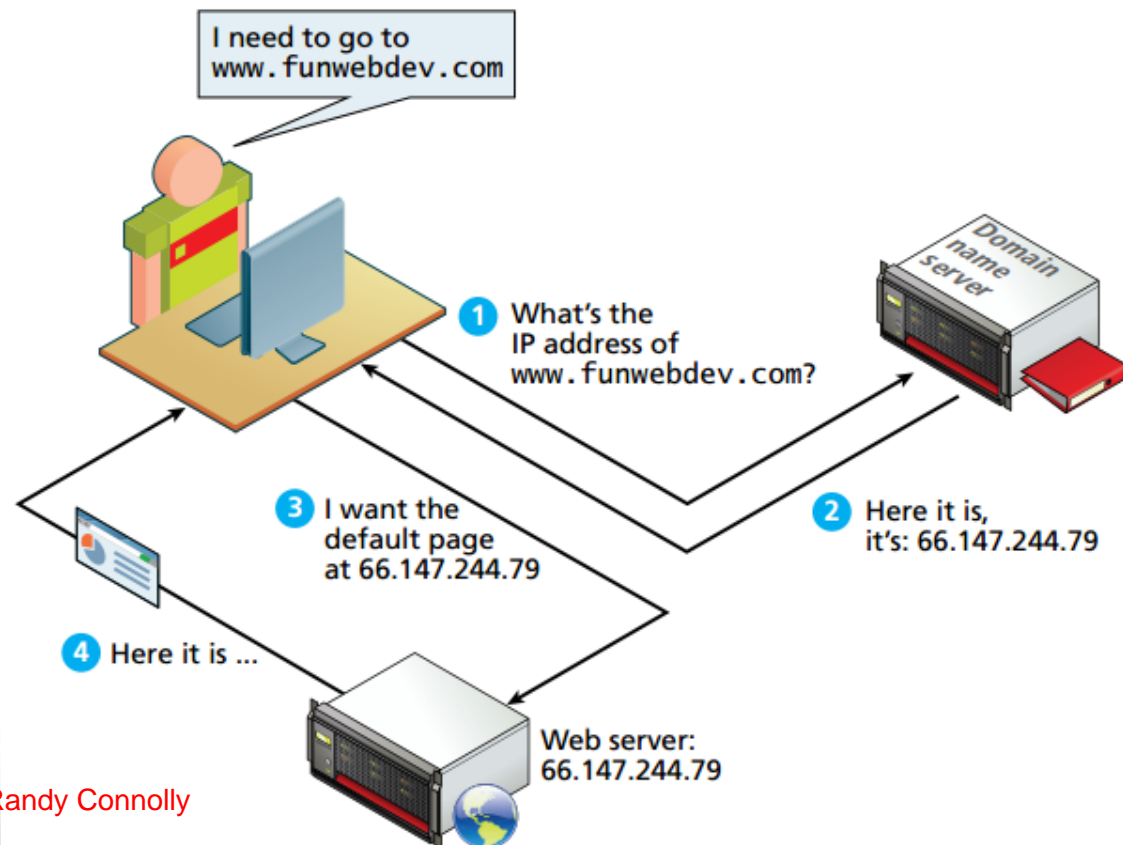
<https://tiptopsecurity.com/how-does-https-work-rsa-encryption-explained/>

<https://www.ssl2buy.com/wiki/symmetric-vs-asymmetric-encryption-what-are-differences>

DNS

Domain Name System

- ❑ To access google search engine, you'd have to type **216.58.216.164** instead of using **google.com**
- ❑ **DNS** is one of the core systems that make an easy-to-use Internet possible.

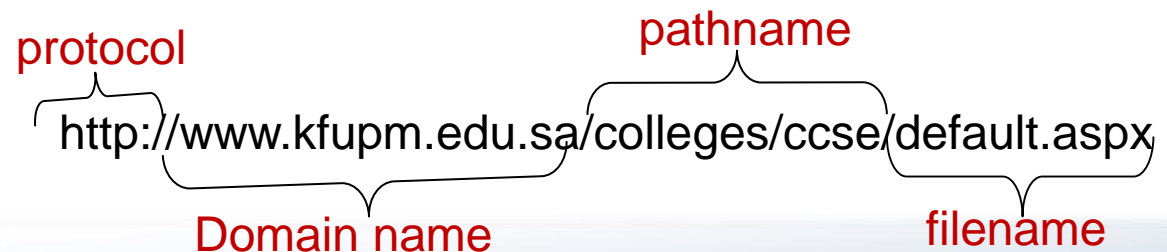


- ❑ DNS protocol is used for resolving domain names to IP addresses.
- ❑ Domain names follow **hierarchical model** that you can follow from top to bottom if you read the name from the right to the left.
 - For example, the domain name **kfupm.edu.sa** is the computer connected to the Internet at the King Fahd University (**kfupm**), which is an educational institution (**edu**), in Saudi Arabia (**sa**)
 - No other computer on the Internet has the same domain name.

URL

Uniform Resource Locator

- ❑ 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 **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**



URL Components

Web Operation

- ❑ A user starts a browser on his computer
- ❑ Request a **web document** by specifying its **URL** e.g.,
<http://www.kfupm.edu.sa/>
 - Browser resolves the URL to get the **server IP address** using the **DNS** server.
 - Then, the browser sends a message to the server requesting the required document
- ❑ Based on the type of the requested document, the web server
 - **finds** the document in its file system and **sends** it back to the browser
 - **creates** a response document and sends it back
- ❑ The browser interprets the content of the document and displays (renders) it for the user
- ❑ If the document contains images,
 - Images will be on separate files and only their URLs will be embedded in the base document
 - The browser will send a different request for each image file

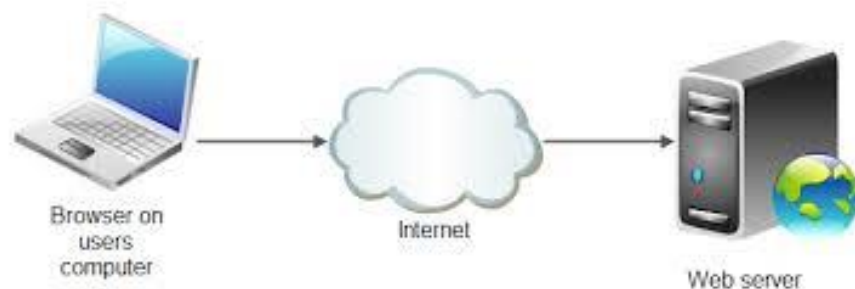
Web programming technologies

❑ Client-side programming technologies

- To build **web pages** and applications that are **run on the client** (i.e., in the browser on the user's device)

❑ Server-side programming

- To build the **applications** that **respond to requests** from client-side web browsers,
 - such as searching the Internet, checking your bank-account balance, ordering a book from Amazon, bidding on an eBay auction and ordering concert tickets.

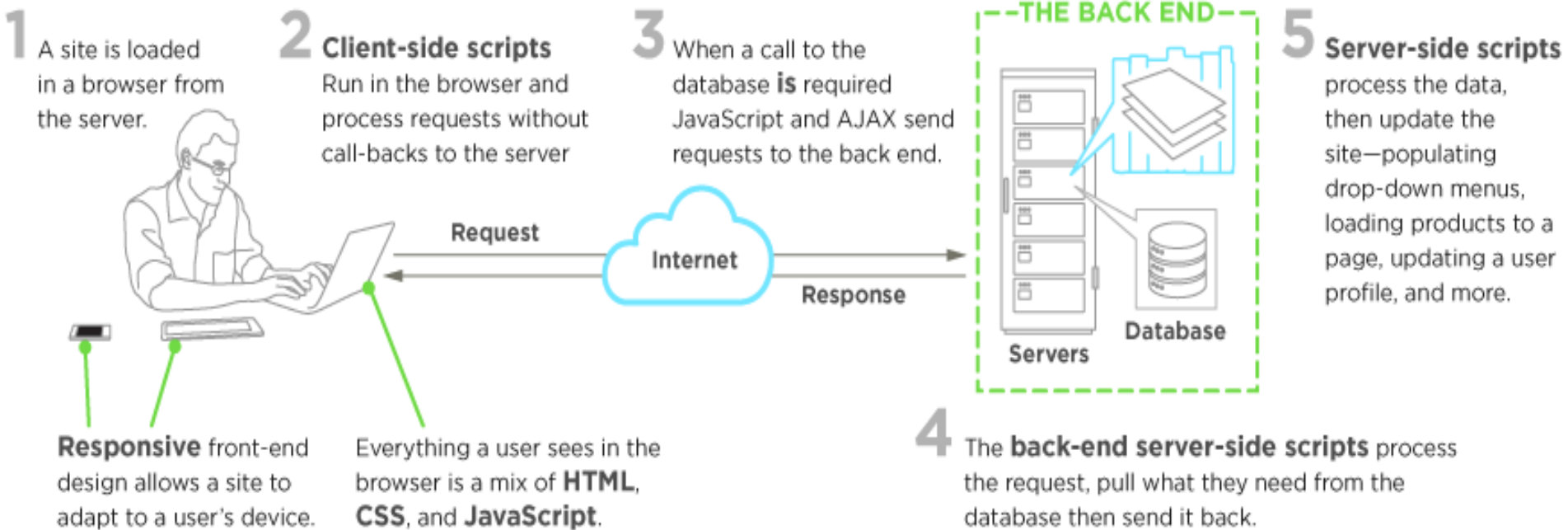


Scripting

- ❑ **Client-side scripting** is executed by a browser,
 - The **processing** takes place on the **end users** computer.
 - The source code is **transferred** from the web server to the users computer over the internet and **run directly in the browser**.
 - Client-side scripting with **JavaScript** can be used to validate user input, to interact with the browser, to enhance web pages, and to add client/server communication between a browser and a web server.

- ❑ **Server-side scripting** is executed by a web server.
 - A user's request is **fulfilled** by running a script directly on the web server to generate dynamic HTML pages. This HTML is then sent to the client browser.
 - It is usually used to provide **interactive** web sites that interface to databases or other data stores on the server.
 - Programmers have more **flexibility** with server-side scripts, which often generate custom responses for clients.

Scripting...



Scripting...

❑ Client-side scripting

- Allow for **more interactive** by immediately responding to user actions
- Execute **quickly** because they don't require a trip to the server
- Can give the developers more **control over the look and behavior** of their web pages

❑ Server-side scripting

- is used to **connect to the database** that reside on the web server
- can **access the file system** residing at the web server
- The used languages have a **wider range of capabilities**

Scripting...

- ❑ Client-side scripting does have **limitations**,
 - **Browser dependency**; the browser or scripting host must support the scripting language and capabilities.
 - Client-side **scripts can be viewed by the client** by using the browser's source-viewing capability.
 - **Sensitive information**, such as passwords or other personally identifiable data, should not be on the client.
 - All client-side data **validation should be mirrored on the server**.
 - placing certain operations in JavaScript on the client can open web applications to security issues.

