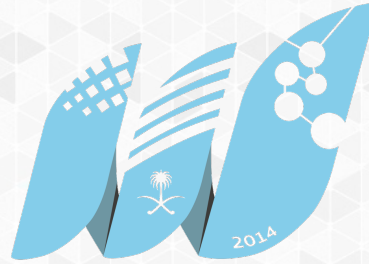


COLLAGE OF COMPUTER
SCIENCE & ENGINEERING

UNIVERSITY OF JEDDAH



جامعة جدة
University of Jeddah

كلية علوم
وهندسة الحاسب
جامعة جدة

Welcome to the Web Development Course

CCSW 321 (Web Development)

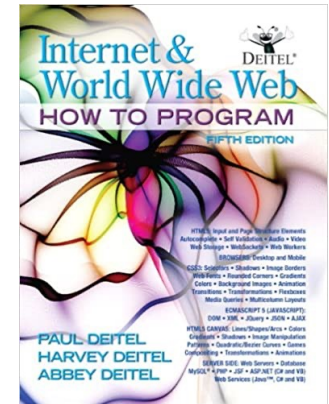
What is this course about?

- It is basically a **web** programming course.
- You are expected to have basic knowledge in Programming.
- You are expected to have basic Database & MySQL knowledge.
- We will mainly use a text editor (e.g., Sublime or VS Code) and your Google Chrome.
- You must ensure that you have a text editor installed, the recommend editor is VS Code.
- You must ensure you have Google Chrome on your machine as we will use the developer's tools that comes with the browser.

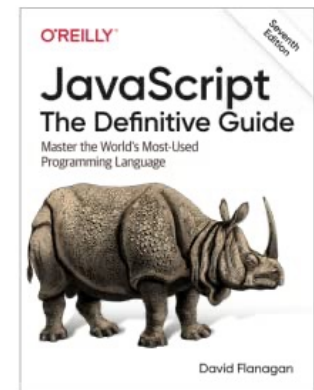


Recommended Books

- Deitel, "Internet and World Wide Web: How to Program", Pearson Education (Prentice Hall), 5th Edition, ISBN-13: 978-0132151009 ISBN-10: 0132151006 ©2013.



- David Flanagan, JavaScript: The Definitive Guide, (2020), 7th edition, O'Reilly. ISBN:1491952024



Learning Outcomes

- Explain web architecture, HTTP protocol and essential methods for Internet communication. Know the basic concepts related to web development.
- Use Hyper Text Markup Language (HTML5) and the Cascading Style Sheet (CSS3) to construct and format web-pages.
- Develop web-pages that are able to handle Client-Side events and required functionalities using JavaScript.
- Develop web-pages that are able to handle Server-Side scripting and communication to data layer using JavaScript & MySQL.
- Engage in a team and play roles in the design and the development of a web application.

Syllabus

- Everything you need to know about the course can be found in the Syllabus
- The syllabus can be found in the blackboard

Assessment Task	Week Due	Assessment Score
Quizzes	3,5,12	10%
Assignments	5,10	10%
Labs	Weekly	10%
Project	6,11	20%
Mid-term Exam	7	20%
Final Exam	12	30%

Quizzes

- We will have **three** quizzes.
- All quizzes are going to mainly consist of multiple-choice questions. They may contain code completion questions as well.
- All quizzes are going to be on the Blackboard.
- You will be assigned enough time for each quiz, but you must complete the quiz in a **single session**.
- It is **your responsibility** to make sure you have a working internet and a working laptop. Once you open a quiz, you must complete it.
- There will be **no second chances**.

Assignments

- You will have a total of two assignments.
- **Assignment 1** will be used to assess your knowledge in using HTML/CSS.
- **Assignment 2** will be used to assess your knowledge with JavaScript.
- They are **individual** assignments (Not a group).
- For each assignment, you will be given an **assessment sheet** that we will use to grade your work. Make sure you understand the requirements and what is being asked from you.

Labs

- You will have roughly one lab per week, starting from week 2.
- The labs are meant to help you gain some hands-on experience with web development every week, so make sure to utilize them.

Project

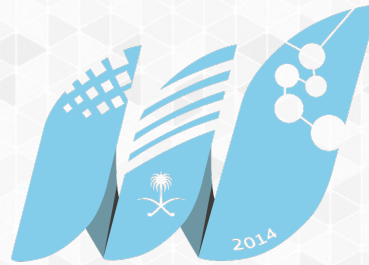
- You will need to form teams consisting of two to four members.
- All teams must be formed by the end of week 3, so start to make your team right now!
- This will be term-long project. It is worth 20% of your grade!
- You will have two submissions, one will be due before the midterms, and another before the final exams.
- At the end of the semester, you will present a demo of your project.

Policies

- All submitted code **MUST BE** your **OWN** code.
 - No auto-generated code.
 - No sharing between students is allowed.
 - If instructors determines you broke this policy, all involved parties will receive **zero**.
- You will be asked to follow a submission guideline for all submissions. Make sure you read it properly and understand what is requested.
- This is a fun and practical class, try to make the best of it :)

**COLLAGE OF COMPUTER
SCIENCE & ENGINEERING**

UNIVERSITY OF JEDDAH



جامعة جدة
University of Jeddah

**كلية علوم
وهندسة الحاسب**

جامعة جدة

Introduction

CCSW 321 (Web Development)

What will be covered

- Introduction to WWW and its technologies.
- Web basics: Servers, DNS, URI, and URLs.
- The HTTP protocol, headers, response codes, and message types (GET, POST, PUT, and DELETE).
- Client-Server model.
- The Evolution of the Web.

Course Topics

- Introducing the **Internet** and World Wide Web (**WWW**).
- Building content with **HTML5**.
- Styling content with Cascaded Style Sheets **CSS3**.
- Adding client-side interactivity through **JavaScript**.
- Adding Server-side Scripting for content and state storage through **Node.js** and database integration with **MySQL**.

World Wide Web (WWW)

- The invention of the WWW is usually attributed to **Tim Berners-Lee**, who, along with the Belgian **Robert Cailliau**, published a proposal in 1990 for a hypertext system while both were working at CERN in Switzerland.



Introduction

Core Features of the Web

- **URL** to uniquely identify a resource on the WWW.
- **The HTTP protocol** to describe how requests and responses operate.
- A **front-end software** that uses **HTML, CSS, and JS** to publish documents and process HTTP Requests/Responses.
- A **browser** to make HTTP requests and display the HTML it receives.
- A **back-end software** that can respond to HTTP requests, can be done using a variety of languages, e.g., Python, PHP, JavaScript, etc.

Introduction

- Web programming technologies are designed to be **portable**, allowing to design web pages and applications that run across an enormous range of Internet-enabled devices.
- **Client-side programming** build web pages and applications that are run on a client's browser.
- **Server-side programming** build web apps that respond to requests from a client's browser, such as searching the Internet, checking bank-accounts , ordering a book from Amazon, etc.

HTML5

- **HTML** (Hypertext Markup Language) is a markup language designed to specify the content and structure of web pages in a portable manner.
- Web pages are written in **HTML**, so browsers display a web page by reading and interpreting its HTML.
- The HTML file might link to other resources, like images, videos, as well as JavaScript and CSS (stylesheet) files, which the browser then also loads.
- **HTML5** is the latest version of HTML. Previously, we had HTML4, XHTML, and others.

CSS3

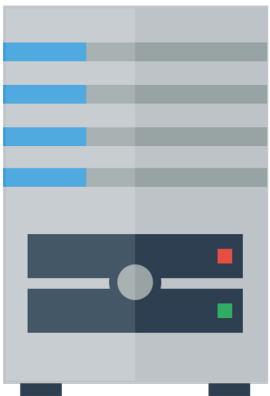
- **Cascading Style Sheets (CSS)** specify presentation or styling, of elements on a web page (e.g., fonts, spacing, sizes, colors, positioning). CSS designed to style portable web pages independently of their content and structure.
- Although HTML5 provides capabilities for controlling presentation, it's best practice **NOT to mix it** with content.
- **Separating styling from page content and structure**, can easily change the look and feel of the pages on an entire website, or a portion of a website, simply by swapping out one style sheet for another.
- **CSS3** is the current version of CSS.

JavaScript

- Helps build dynamic web pages (i.e., pages that can be modified “on the fly” in response to events, such as user input, time changes, and computer apps.
- Enables the **client-side programming** of web apps.
- JavaScript is a portable scripting language. Programs written in JavaScript can run in web browsers across many devices to create **client-side interactivity**.
- JavaScript can also run on the server for **server-side scripting**, e.g., to integrate with a database.

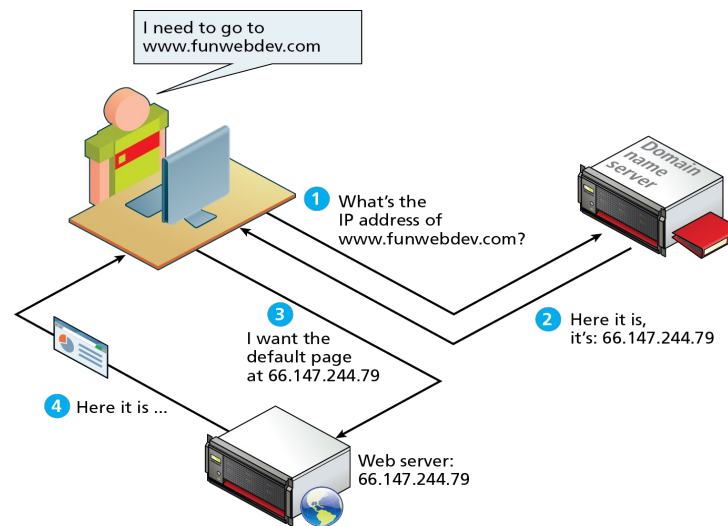
Web Basics: Web Server

- Any machine, including your own laptop, can be configured to act as a “server”.
- A web server is a program running on a computer that delivers web pages in response to requests.
- It either stores or generates the web page returned.



Web Basics: DNS

- Human beings do not enjoy having to recall long strings of numbers.
- Instead of IP addresses, we use the Domain Name System (DNS)
- An Internet Domain Name System (DNS) server keeps a database of hostnames and their corresponding IP addresses and performs automatic translations.



Web Basics: URIs and URLs

- URIs (**Uniform Resource Identifiers**) identify resources on the Internet. URIs that start with `http://` are called URLs (**Uniform Resource Locators**).
- A URL contains information that directs a browser to the resource that the user wishes to access.
- Web servers make resources available to web clients.
- Popular web servers include Apache's HTTP Server and Microsoft's Internet Information Services (IIS).

Web Basics: URL

- In order to allow clients to request a particular resources from the server, a naming mechanism is required so that the client knows how to ask the server for file.
- That naming mechanism is the **Uniform Resource Locator**

`http://www.funwebdev.com/index.php`

The diagram shows the URL `http://www.funwebdev.com/index.php` with three parts identified by blue lines and labels below them: `http` is labeled "Protocol", `www.funwebdev.com` is labeled "Domain", and `/index.php` is labeled "Path". The "Path" label is enclosed in a green rounded rectangle.

- A path includes the folder(s) / directory(ies) and the page filename and extension

Web Basics: URL

Part	Description
Domain	The domain identifies the server from which we are requesting resources
Port	The optional port attribute allows us to specify connections to ports other than the defaults
Path	The path is a familiar concept to anyone who has ever used a computer file system. The root of a web server corresponds to a folder somewhere on that server

Web Basics: URL Example

- <http://www.deitel.com/books/downloads.html>
- http://: A Hypertext Transfer Protocol to get a resource.
- www.deitel.com : Qualified hostname of the web-server
- The hostname is translated into an IP (Internet Protocol) address, a unique value that identifies the web server.
- An Internet Domain Name System (DNS) server keeps a database of hostnames and their corresponding IP addresses and performs automatic translations.

Web Basics: URL Example

- (/books/downloads.html) specifies resource's location /books and name (downloads.html) on the web server.
- The location could represent an actual directory on the web server's file system. For security reasons, though, the location is typically a virtual directory.
- The web server translates the virtual directory into a real location on the server, thus hiding the resource's true location.

The HTTP Protocol

- The World Wide Web is about **communication** between web **clients** and web **servers**. The **HTTP** or Hypertext Transfer Protocol handles communication between the **client** (i.e., browser) and the **server** (where files resides).
- The **client** identifies itself with the **user-agent**, which is any tool that makes requests for the benefit of the user. This is typically a web browser, although other applications interact with resources using HTTP as well.
- The web **server** is the entity that serves the documents, or resources, that are requested by the client. The server will also appear as a single machine but in reality, can be a network of servers.

The HTTP Protocol

- The HTTP protocol has main components are the client, the server, and the method of communication.
- HTTP has two message types: HTTP **Request** and HTTP **Response**.
- Each message contain: Headers, body, method type.
- The **headers** supply the server with additional information it may need.
- The **body** contains the content of the message.

The HTTP Protocol

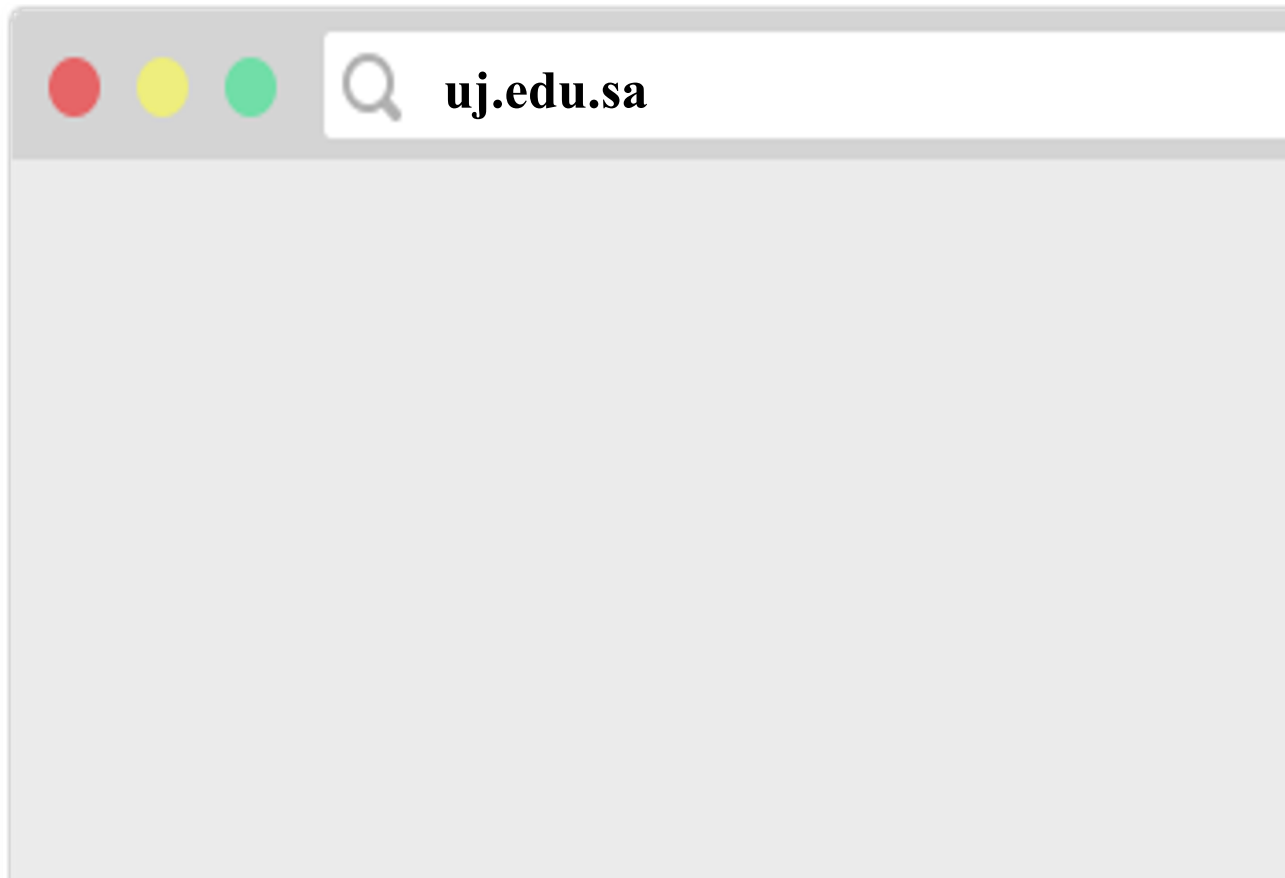
- The **method** specifies what operation is to be performed. Many methods exist, but the most common are:
 - **GET**: used to request data from a specified resource.
 - **POST**: used to send data to a server to create/update a resource.
 - **PUT**: used to send data to a server to create/update a resource.
 - **DELETE**: used to delete a specified resource.
- More details on GET vs. POST will be discussed later.

The HTTP Protocol

- Communication between clients and servers is done by requests and responses as follows:
 - A client (a browser) sends an HTTP request to the web.
 - A web server receives the request.
 - The server runs an application to process the request.
 - The server returns an HTTP response (output) to the browser.
 - The client (the browser) receives the response.

The HTTP Protocol

1. When you open your browser and type in a URL, which is the address of the HTML file on the internet.

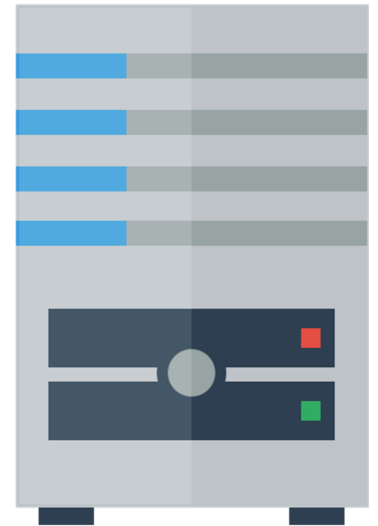


The HTTP Protocol

2. The browser asks the web server that hosts the document to send that document (e.g., sends a **request**).



request

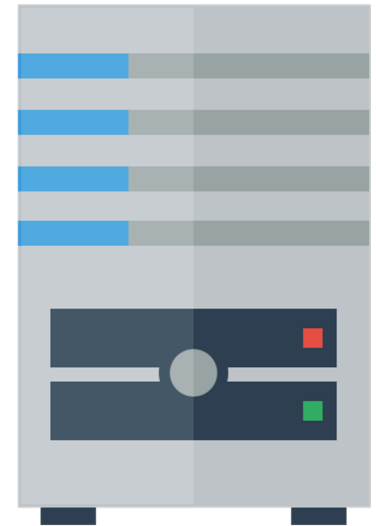


The HTTP Protocol

3. The web server **responds** to the browser with the HTML file that was requested.

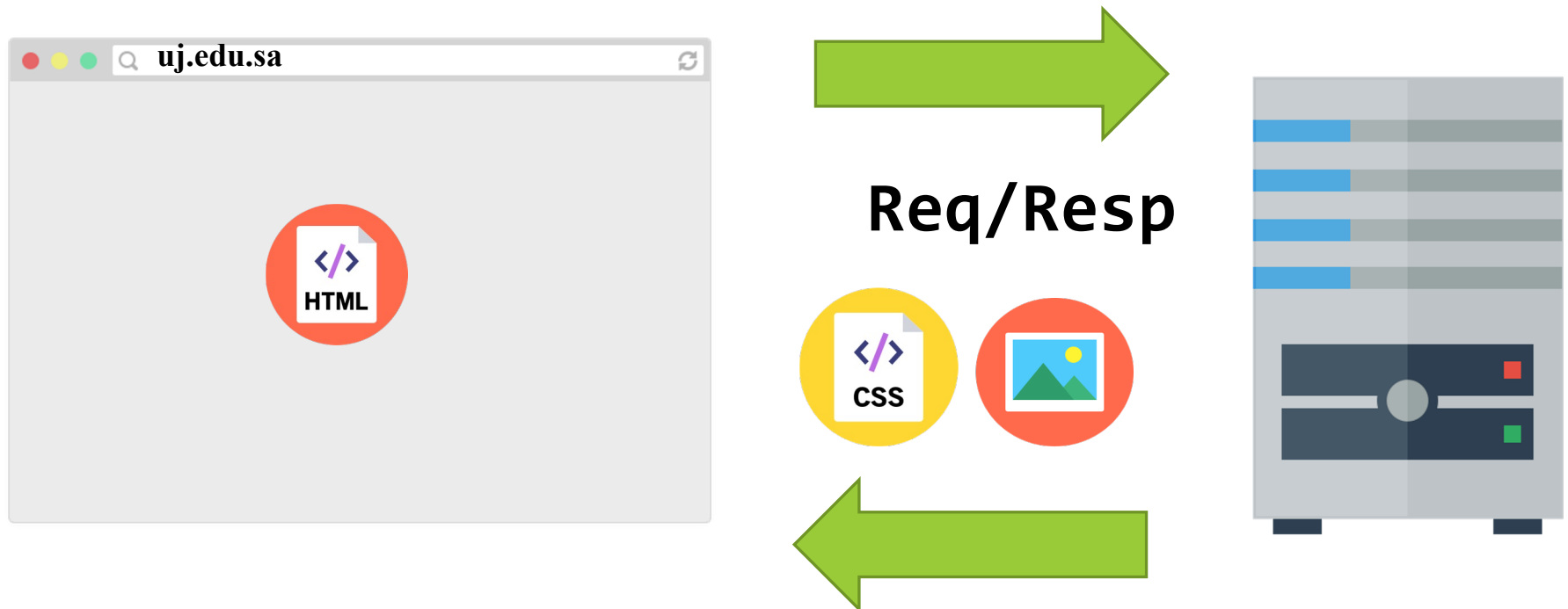


response



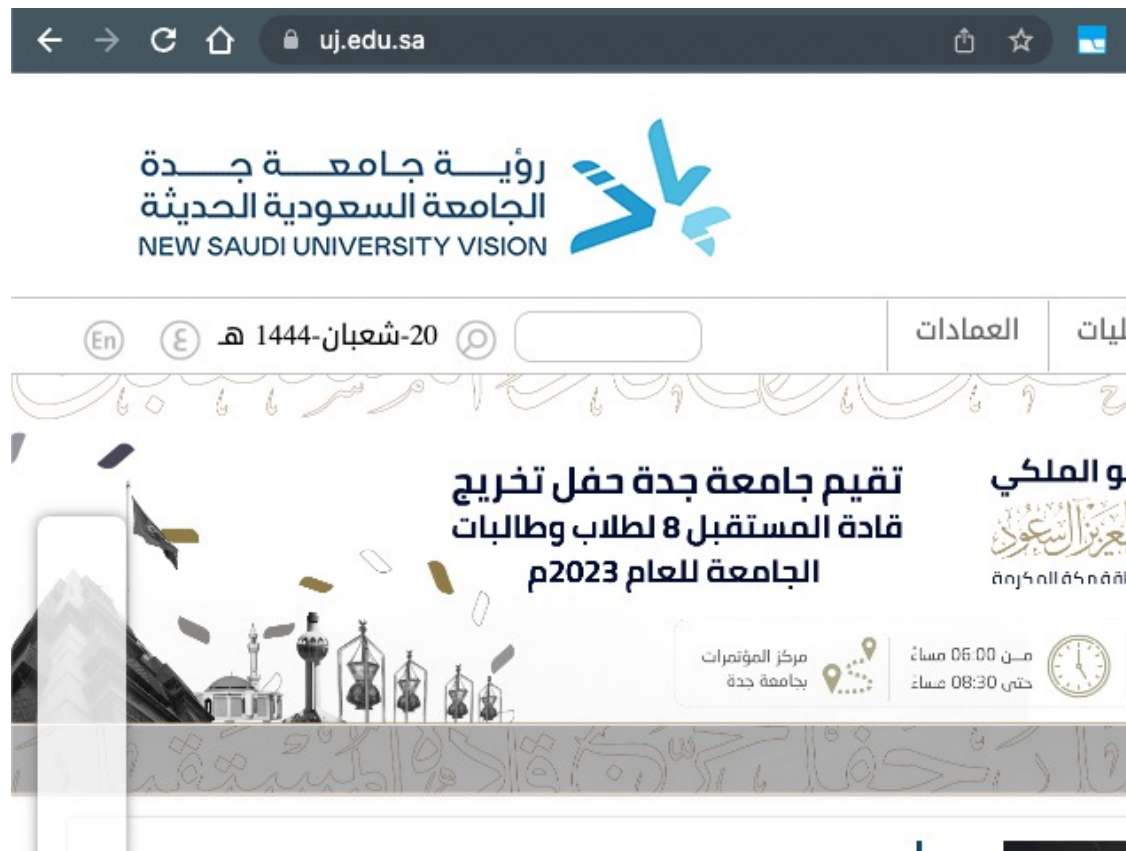
The HTTP Protocol

4. The browser reads the HTML, sees the embedded resources and asks the server for those as well.



The HTTP Protocol

5. The web page is loaded when all the resources are fetched and displayed.



The HTTP Protocol Headers

- Headers are one of the most powerful aspects of HTTP, they define what kind of response the server will send.
- **Request headers** include data about the client machine.
 - Web developers use this information for analytic reasons and for site customization.
 - Some the Request headers include: *Host*, *User Agent* , *Accept*, *Connection*,...
- **Response headers** have information about the server answering the request and the data being sent.

The HTTP Protocol Headers

Request Header	Description
Host	it allows multiple websites to be hosted off the same IP address
User-Agent	the most referenced header in modern web development. It tells us what kind of operating system and browser the user is running
Accept	It tells the server what kind of media types the client can receive in the response
Connection	this header specifies whether the server should keep the connection open, or close it after response

The HTTP Protocol Headers

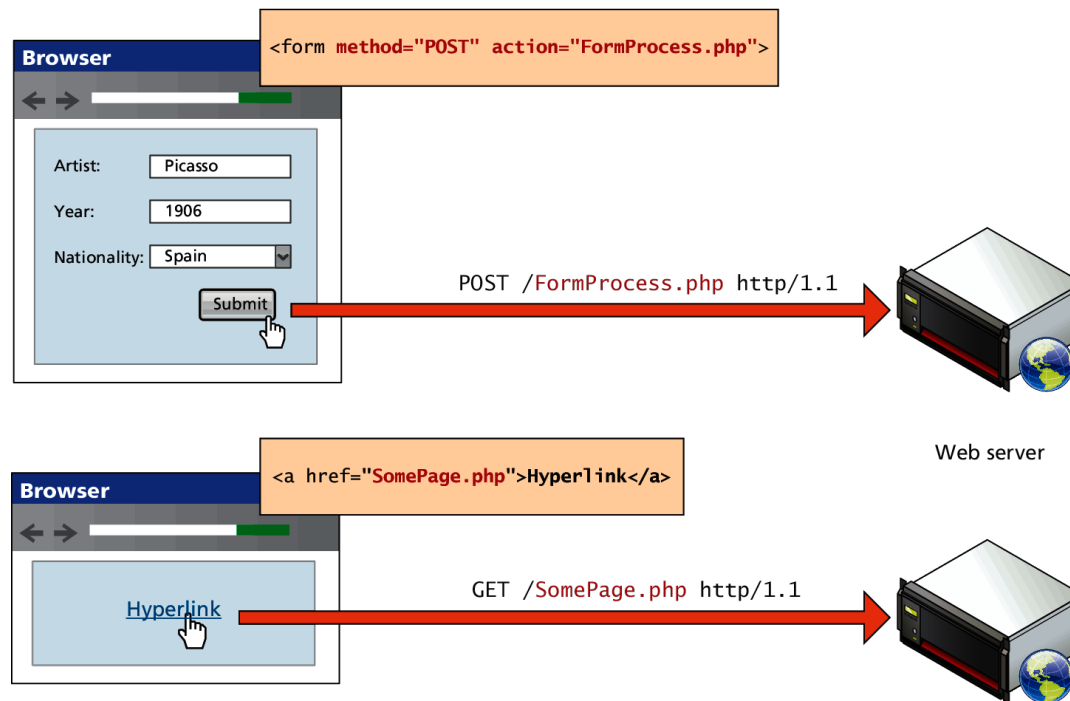
Response Header	Description
Response codes	are integer values returned by the server as part of the response header
Server	tells the client about the server. It includes the type of operating system the server is running as well as the web server software that it is using
Last-Modified	contains information about when the requested resource last changed
Content-Type	To accompany the request header Accept, it tells the browser what type of data is attached in the body of the message
Content-Length	specifies how large the response body (message) will be

The Response Codes

Code	Description
200: OK	The 200 response code means that the request was successful.
301: Moved Permanently	Tells the client that the requested resource has permanently moved. Codes like this allow search engines to update their databases to reflect the new location of the resource. Normally the new location for that resource is returned in the response.
304: Not Modified	If the client so requested a resource with appropriate Cache-Control headers, the response might say that the resource on the server is no newer than the one in the client cache. A response like this is just a header, since we expect the client to use a cached copy of the resource.
307: Temporary redirect	This code is similar to 301, except the redirection should be considered temporary.
400: Bad Request	If something about the headers or HTTP request in general is not correctly adhering to HTTP protocol, the 400 response code will inform the client.
401: Unauthorized	Some web resources are protected and require the user to provide credentials to access the resource. If the client gets a 401 code, the request will have to be resent, and the user will need to provide those credentials.
404: Not found	404 codes are one of the only ones known to web users. Many browsers will display an HTML page with the 404 code to them when the requested resource was not found.
414: Request URI too long	URLs have a length limitation, which varies depending on the server software in place. A 414 response code likely means too much data is likely trying to be submitted via the URL.
500: Internal server error	This error provides almost no information to the client except to say the server has encountered an error.

HTTP Methods

- The HTTP protocol defines several different types of requests, each with a different intent and characteristics.
- The most common requests are the GET and POST request



HTTP Methods: GET

- A GET request typically retrieves information from a server, such as an HTML document, or search results based on a user-submitted search term.
- GET requests are only used to request data (not modify)
- GET requests should never be used when dealing with sensitive data.
- A GET request typically has length restrictions due to its use of **query strings** for data sending, so it's often necessary to send large amounts of information using the POST method instead.

HTTP Methods: GET Querystring

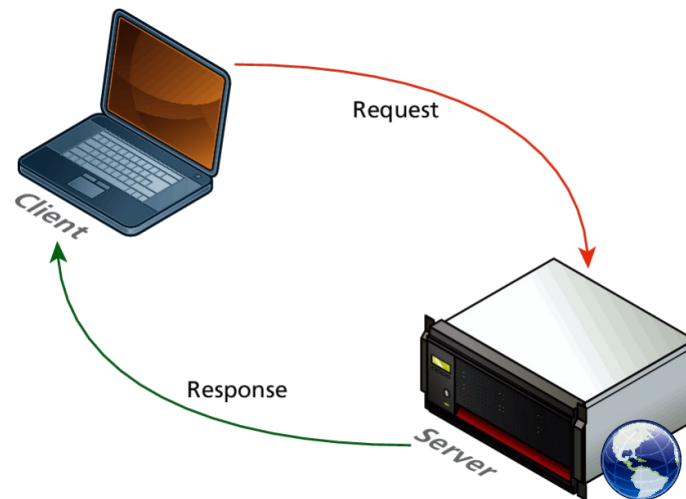
- A GET request appends data to the URL, e.g., **www.google.com/search?q=deitel:**
- search is the name of Google server-side form handler
- q is the name of a variable in Google's search form
- deitel is the search term
- ? separates the query string from the rest of the URL
- A name/value pair is passed to the server with the name and the value separated by an equals sign (=).
- If more than one name/value pair is submitted, each pair is separated by an ampersand (&).
- The server uses data passed in a query string to retrieve and send an appropriate resource.

HTTP Methods: POST

- A **POST request** typically sends data to a server (e.g. form data or documents).
- A POST request sends form data as part of the HTTP message body, not as part of the URL.
- POST requests have no restrictions on data length.
- The POST method is also **preferred** for sending **sensitive data** because it hides the submitted data from the user by embedding it in an HTTP message.
- If a form submits several hidden input values along with user-submitted data, the post method might generate a URL like `www.searchengine.com/search`.

Client-Server Model

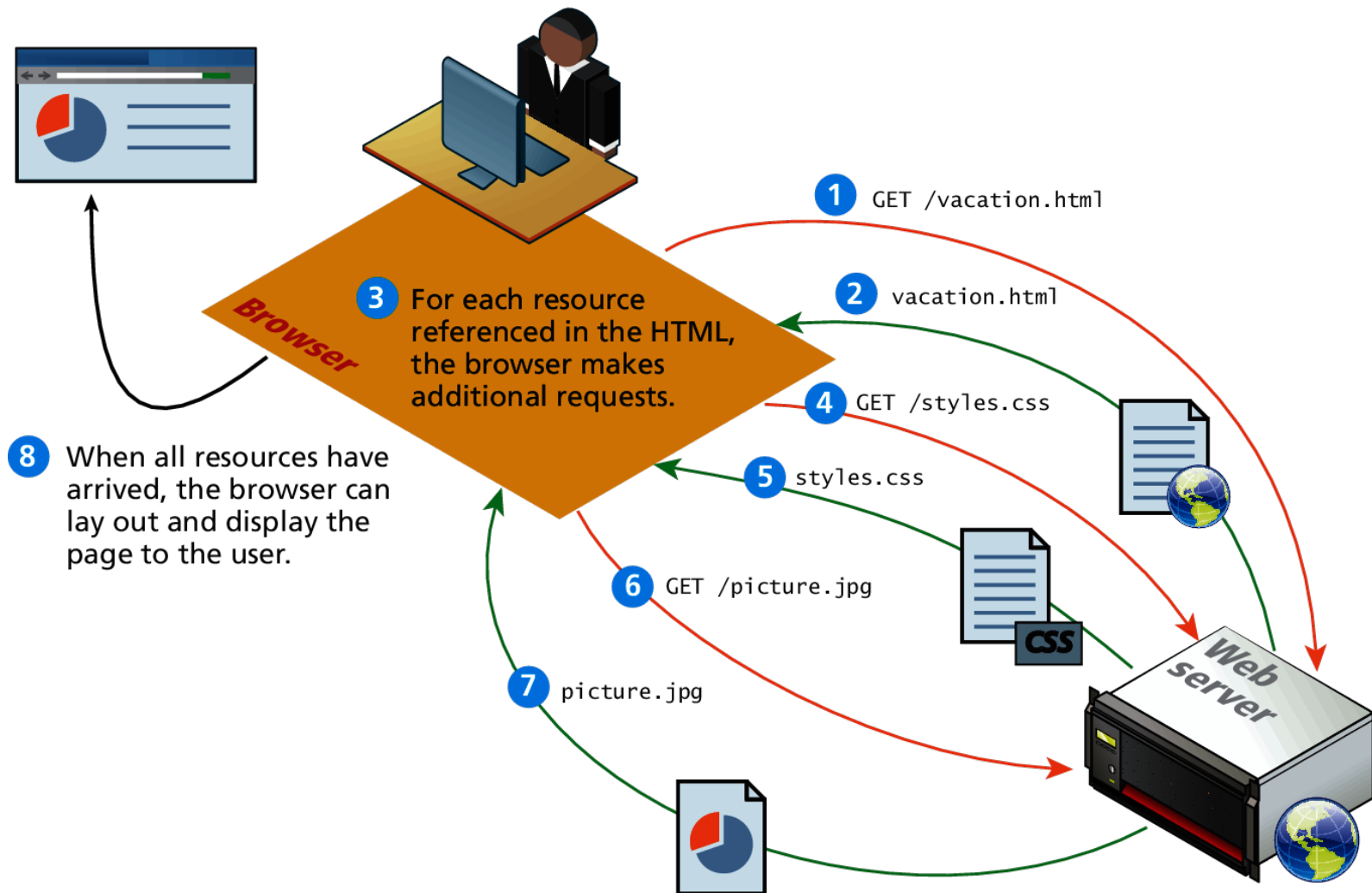
- Within the client-server model, the **request-response loop** is the most basic mechanism on the server for receiving requests and transmitting data in response
- The client initiates a **request** to a server and gets a response that could include some resource like an HTML file, an image or some other data



Client-Server Model

- In the **client-server model**, there are two types of actors: clients and servers.
- The **server** is a computer agent that is normally active 24 hours a day, 7 days a week (or simply 24/7), listening for queries from any client who make a request.
- A **client** is a computer agent that makes requests and receives responses from the server, in the form of response codes, images, text files, and other data.

Client-Server Model



Evolution of WWW

- **Web 1.0:** It was read-only. You can only search and view web documents, i.e., websites.
- **Web 2.0:** Embraced an architecture of participation: a design that inspires user interaction and community contributions. This is how social media websites started! Users now can participate in the creation of web documents.
- **Web 3.0:** The decentralized approach where everything runs on the blockchain.



Any questions?
Please feel free to raise your
hands and ask.