CSCB20
Introduction to Databases and Web Application

Week 5 - Introduction to Web Development

Dr. Purva R Gawde

Database applications and the Web

Three tier architecture

- The Web Server:
  - Provides the user interface.
- The Application server:
  - middle tier, housing the business logic used to process user inputs
- The Database server:
  - Data or backend tier

What is Web Development?

- Web development is a specific field of software engineering that focuses on building web pages.
- Web pages, or web apps, are codebases that are downloaded and run in our web browser (e.g., Google Chrome) each time a user navigates to the website address.
- Two main categories: front end and back end.

Website vs Web application

- Websites:
  - accessible through browsers, display useful content.
  - navigable and can present digital content, images, video, and audio.
  - static,
  - built using HTML, CSS, and maybe some JavaScript.

- Web Applications:
  - websites with functionality and interactive elements.
  - Gmail, Facebook, YouTube, Twitter, etc.
  - computer software accessed through a web browser, often connected to a database

Frontend

- **Front end** is what we see when we open a web page or app.
- The front end is built out of three languages: HTML, CSS, and JavaScript.
- HTML:
  - allows us to put content on our page
- CSS:
  - used to style our page
- JavaScript:
  - makes our page dynamic

Backend

- This term usually refers to what happens ‘behind the scenes’: servers, databases, etc.
- Data storage (databases) and servers running to provide data for the front end.
- JavaScript, Ruby, Java, or Python
- The database logic required in back end development often utilize a database language, such as SQL or MongoDB.

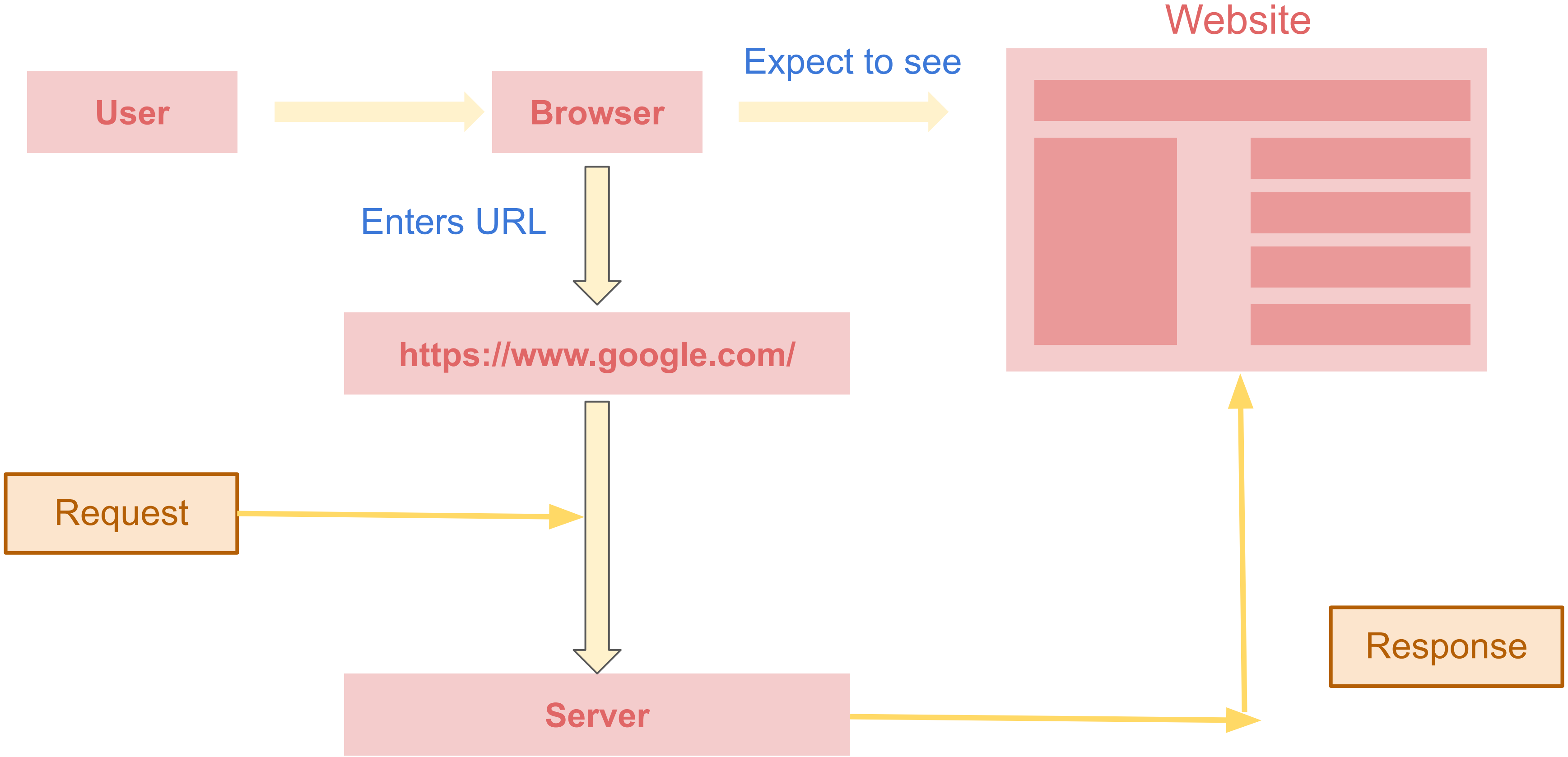
Framework

- Frameworks aren’t exactly new languages, but are more like add-ons to existing languages.
- Slightly change the rules and syntax of a language, but save us a lot of time and effort in writing web development code.
- For example, a CSS framework like Bootstrap will require us to write our CSS using slightly different rules than regular CSS.

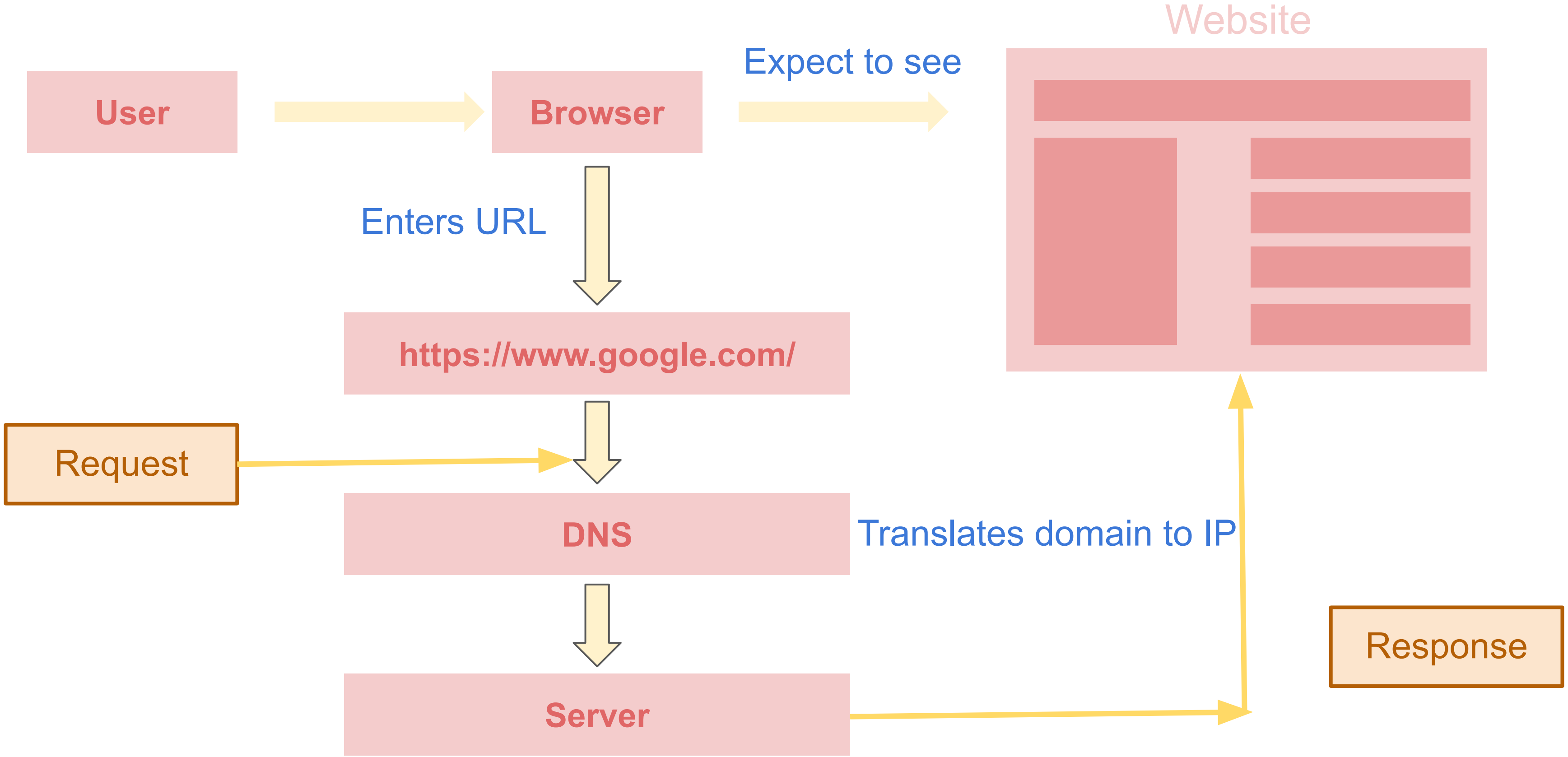
Tools used for web development

- Text editors
  - Visual Studio or Atom, Sublime (text editor)
  - CodePen (sandbox environment)
- Frontend:
  - Languages: HTML, CSS, Javascript
  - Framework: Angular, React, Vue.js
- backend:
  - Languages: Python, PHP, Node.js
  - Framework: Python Flask, Django

How the Web Works



How the Web Works



Frontend:
Introduction to HTML and CSS

HTML

- **HTML (Hyper Text Markup Language)** is one of the languages we use for web development.
- Read by the browser and then used as a blueprint for displaying information on your screen.
- **Markup language :**
  - specifically designed to render data in a graphical form (rather than execute tasks).

How to write HTML code?

- Writing Options:
  - Text editor: VSCode, Atom and Sublime
  - Sandbox environments: CodePen
- Execute
  - Web browser

HTML elements

- HTML is made up of elements.
    - boxes of content on our web page.
    - Different types of content will be contained in different boxes.
- We will be focusing on different display elements today:
- division elements (divs), headers, paragraphs, and spans.

HTML Tags

- <p>
  - The p (“paragraph”) element.
- <div>
  - The div element is a generic container. It is used primarily for grouping other HTML elements together.
- <span>
  - The span element is a generic text container. It does not create a new line like the p element does.
- <h1>-<h6>
  - These are header tags there are intended to be used as a way to present the subject matter of the page.

HTML attributes

- An attribute is used to define the characteristics of an HTML element and is placed inside the element's opening tag.
- Two parts – a name and a value
  - The name is the property you want to set.
    - <p align>
  - The value is what you want the value of the property to be set within quotations
    - <p align = “left”>

Inspect existing Webpage

- There is a tool built into every browser called the **inspector** that allows us to do this.
- Just right-click any element on a webpage and select “inspect” or “inspect element”, depending on the browser.
- This will pop up a sidebar that allows you to explore all the HTML on the web page.
- Can be overwhelming!!
- Good way to get a taste of what goes into building a complex, production web page.

Making your very own HTML File

HTML Validation to identify problems

https://validator.w3.org/

Introduction to CSS

Introduction to Cascading Style Sheets

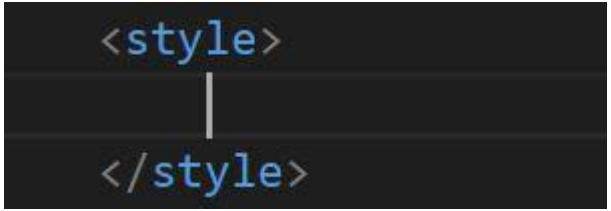
- Think of a web as a house
  - Material, plan, paint and decorations
  - Then electricity and plumbing
- It is a different language than HTML.
- It allows us to add color and style to our web pages.
- Once the browser reads the HTML, it will then read the CSS and give different styling rules to different elements in our HTML based on how we select them.

The style and link Elements

- Need to inform the browser that what it is reading is in fact CSS.
- How?
  - Through HTML elements known as the style and link elements.
  - <style> </style>
  - <link> </link>

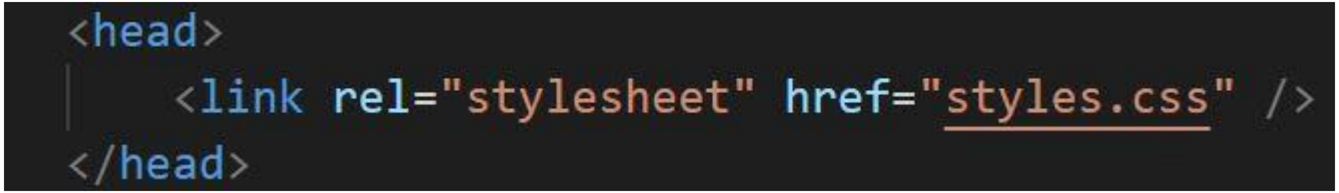
How to include CSS in HTML?

- There are two ways of including CSS in our HTML.
- We can write our CSS directly between two style tags:



```
<style>

</style>
```

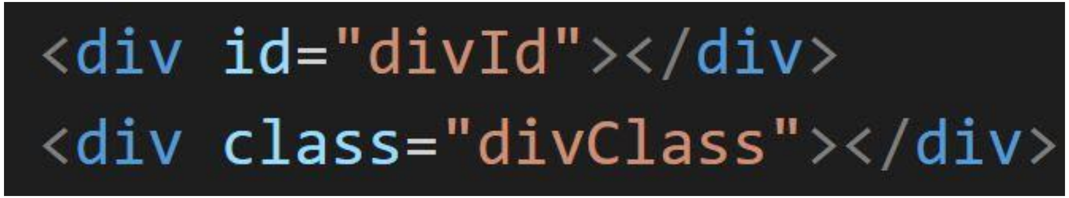
- We can link to an external CSS file using the link element.
    - This element will include two attributes: rel and href.



```
<head>
    <link rel="stylesheet" href="styles.css" />
</head>
```

Intro to CSS Selectors

- **Selectors**: which elements to apply rules to

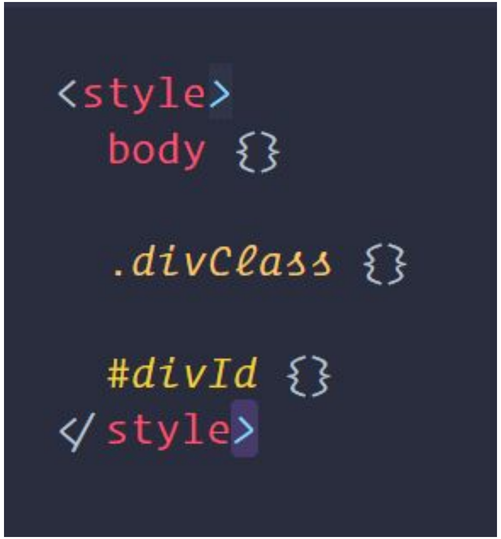


```html
<div id="divId"></div>
<div class="divClass"></div>
```

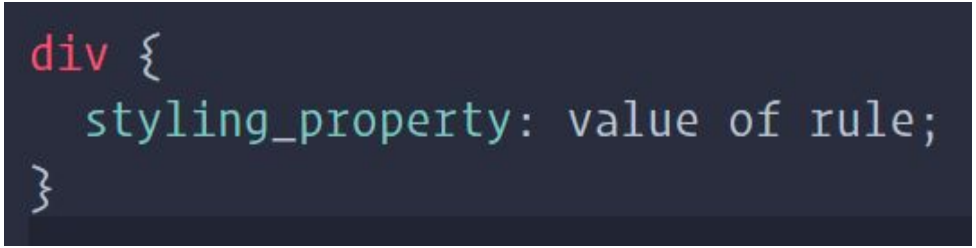
- **Ids:** are titles that can only appear on a single element. Think of it as you would your driver's license number. ONLY you have that one number.
- **Classes:** on the other hand can apply to multiple elements.
- **Individual elements:** We do not need to add anything to use every element of a certain type as a selector. CSS does that for us already.

Anatomy of Styling Rules (Syntax)

- Style tags:
  - Classes will always begin with **.**
  - Ids will always begin with **#**
  - Elements will begin with neither and just have the element name.



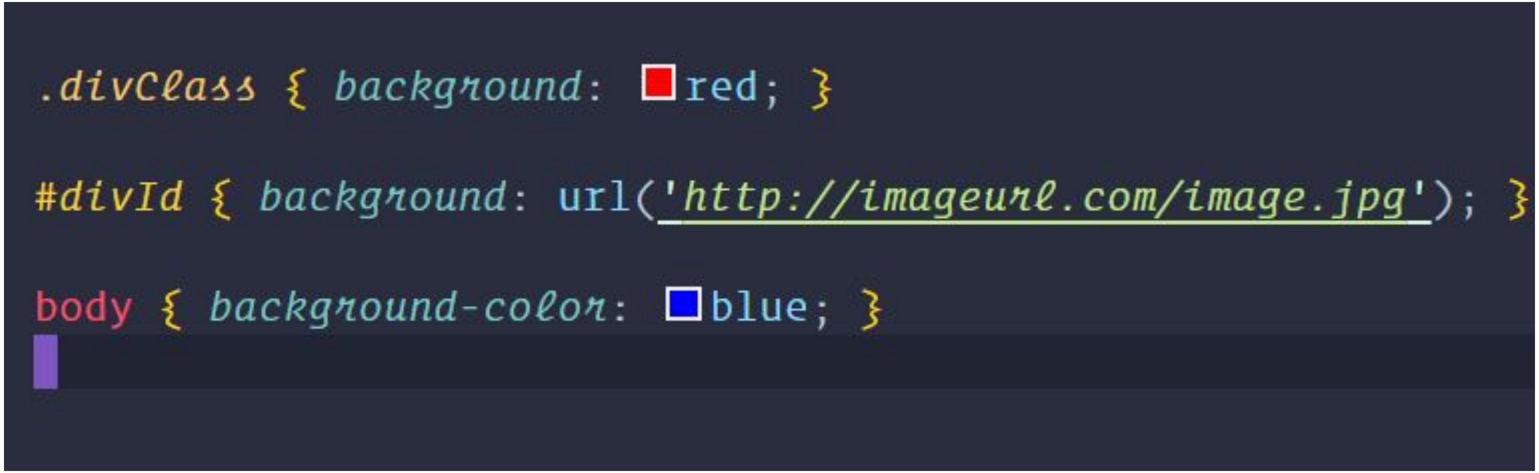
```
<style>
  body {}

  .divClass {}

  #divId {}
</style>
```



```
div {
  styling_property: value of rule;
}
```

Styling Rules

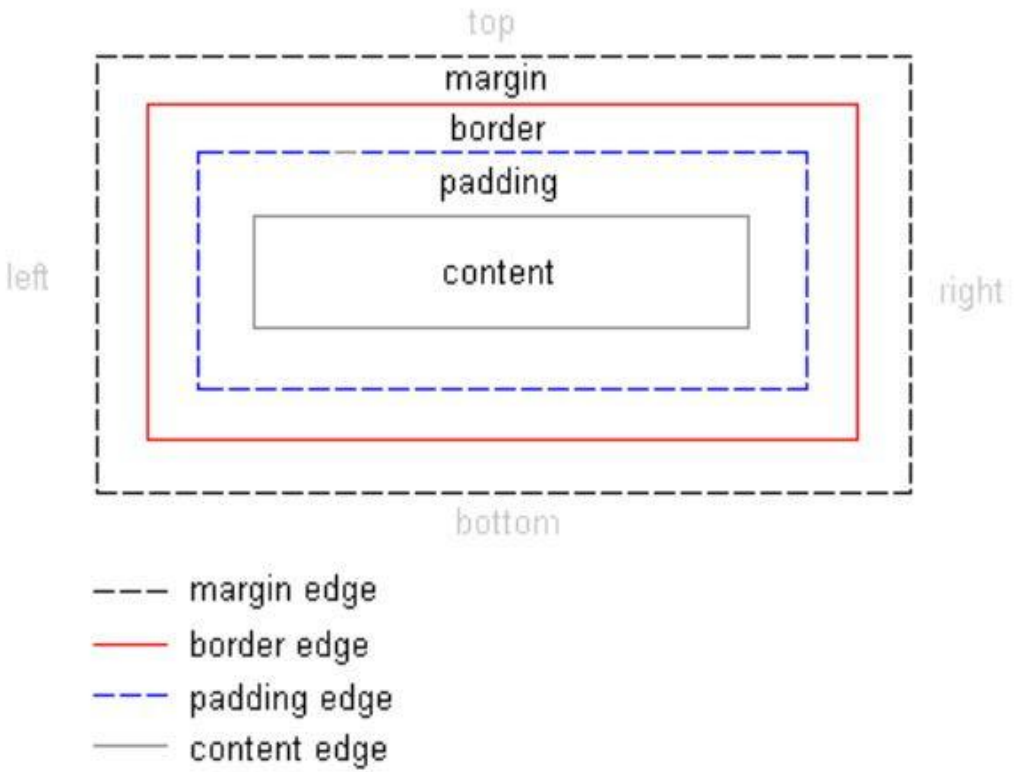
- background or background-color
- font-size



```
.divClass { background: 🟥red; }

#divId { background: url('http://imageurl.com/image.jpg'); }

body { background-color: 🟦blue; }
```

Introduction to the Box Model

- height and width
- margin
- Border
    - a list of border styles can be found here: https://developer.mozilla.org/en-US/docs/Web/CSS/border
- padding



Backend:
Introduction to Python Flask

Python Flask

- What is Flask?
  - Flask is an API of Python that allows to build web applications
  - Web application framework
    - Collection of modules and libraries that helps the developer to write applications without writing the low-level codes
  - Flask is based on WSGI(Web Server Gateway Interface) toolkit and Jinja2 template engine.
- Why Flask?
  - Microframework
  - easier to learn

Python Flask Installation

- Python Version:
- We recommend using the latest version of Python 3.
- Flask supports Python 3.5 and newer, Python 2.7, and PyPy.
- Flask:
  - https://flask.palletsprojects.com/en/3.0.x/