# Introduction

1. What is the story?
   1. Client / Server
   2. Request, Response, and Protocol
2. Response Journey
   1. What does web browser understand?
      1. HTML Content
      2. CSS Presentation
      3. JS (actions, content change, request data, browser functions, timers, etc)
   2. What is the main role of web server?
   3. Static vs Dynamic web site
3. Request Journey
   1. DNS Role
4. Team component
   1. UI/UX Designer
   2. Front-end Developer (s)
   3. Back-end Developer (s)
   4. Team Leader
   5. Business analyst
5. Web Design vs Web Development
6. Different approaches to web development
   1. Embedded code
      1. PHP
      2. JSP
      3. ASP
      4. ASP.net Web Forms
   2. MVC Style
      1. Laravel, Cake PHP
      2. Spring
      3. ASP.net MVC
      4. Python (Django and Flask)
   3. Mobile Development
   4. APIs and JSON
      1. Node JS
   5. Front-end Frameworks
      1. Angular
      2. React
      3. Vue
7. CSS and JS Libraries and Framework
   1. Bootstrap
   2. jQuery
8. Environment Setup
   1. VS Code
   2. Extensions (Live server, Prettier, Zen code, auto-rename tag)
   3. Settings (auto close tag, default formatter)
9. Hello world
10. Development tools
    1. Chrome
    2. Lighthouse extension

# HTML Section

1. What is HTML
   1. Markup not programming language
   2. Element = Tag (open tag, close tag)
   3. Content and Attributes
2. HTML Versions
3. Document Structure
   1. Doctype
   2. Html
   3. Head
      1. Title
      2. Meta-tags (charset)
   4. Body
4. Basic Elements
   1. Text Elements
      1. Headings
      2. Paragraph
      3. B => strong
      4. I => em
      5. Lists ol,ul
   2. Comments
   3. Images
   4. Hyperlinks
      1. #
      2. Target
   5. Tables
   6. DIV, SPAN
5. Block elements vs inline elements
6. HTML 5 Structure elements (Semantic)
   1. NAV
   2. Header
   3. Article
   4. Section
   5. Footer
7. Forms Elements
8. Advanced topics
   1. Advanced meta-tags
   2. Base URL tag
   3. IFrame

# CSS Fundamentals

1. What is CSS?
   1. Visual style and presentation
   2. Properties to format elements
   3. CSS Sources (Google, MDN)
2. Selector + style (property +value) += role
3. Inline, internal, and external styling
4. Text-styling
   1. Size, family, transformation, align, line-height, weight
   2. Web-font
   3. Direction and language
5. Image styling
   1. Cover, auto
6. Combining selectors
   1. Chain & combining (Hard to maintain)
   2. ID, Class selectors (Class is better)
7. Conflicts between selectors
   1. All rules are applied but (! important, inline, id, class, element, universal)
8. Working with colors
   1. RGB, RGBA, Hex short hex
   2. Background, color
9. Borders
10. Pseudo classes
    1. First-child, first, last-child, last, nth-child(2), odd, even, hover (styling tables)
11. Hyper links
    1. Color, text-decoration, a:link, visited, hover, active
12. Inheritance and universal selector
    1. Body -> font-family, strong inside p
    2. Borders not inherited
    3. Text only
13. Box model
    1. Fill area (height,width, border, padding)
    2. Margin
14. Margins and Padding
    1. 1 = all, 2 top,buttom and left ,right
    2. Collapsing margins
    3. 100%, px, auto, rem, fr
15. Centering page with container with auto margin
16. Inline boxes vs block boxes
    1. Inline (only width of content), no line breaks, box model without w,h, padding and margin only horizontally, a, img, strong, em, buttons
    2. Block 100%. Stacked vertically , box model, most of elements
    3. Inline-blocks (inline outside, block inside)
17. Positioning (Viewport)
    1. Relative,
    2. Absolute
18. Pseudo elements (::)
    1. First-letter, first-line, adjacent sibling (h3 + p), after, before (عاجل) inline
19. Build layouts
    1. Definition: ways to arrange components
    2. Steps
       1. Page layout, component layout
    3. Ways to arrange layouts
       1. Floats
          1. Left, right, clear is needed (same as absolute positioning, have no content) (Collapsing elements)
          2. Clear div, clearfix :after
          3. Box-sizing: border-box;
       2. Flexbox
          1. Display:flex (height = taller)
          2. Align-items
       3. Grid
20. Build layout and convert it to RTL
21. Build layout using bootstrap