1. **What is HTML?**
   HTML stands for Hyper Text Markup Language. HTML is not a programming language; it is a *markup language* that defines the structure of your content. HTML consists of a series of **elements**, which you use to enclose, or wrap, different parts of the content to make it appear a certain way, or act a certain way. The enclosing **tags** can make a word or image hyperlink to somewhere else, can italicize words, and can make font bigger or smaller, and so on.

   >> Resource

2. **What is CSS?**
   CSS stands for Cascading Style Sheets, and it's used to style/ decorate/ your html elements/ your web page, for example, to alter the font, colour, size and spacing of your content, split it into multiple columns, or add animations and other decorative features.

   >> Resource

3. **What is Javascript?**
   Javascript is a script or a programming language that allow us to do more complex functionality on web browsers (and at this time also allow us to create applications outside the browser, using node and other frameworks). Every page that you see, that is not totally static, and have dynamic content, animations, scrolling content.

4. **What role does HTML/CSS/Javascript each play in web development respectively?**
   A lot of people say that the HTML role is the structure, the skeleton of your page. The CSS is the style, it's the color, font, it's what makes the page appealing to the eye. The Javascript is the functionality, it's how you web page works, it's how it's behave.

5. **What type of things go inside of a webpage's `<head>` section?**
   The head tag in your html tag, it's basically a container for your page meta-data (Do you guys know what is metadata?) Basically all the information needed about your page, and everything that is inside the Head tag it won't be parsed into the body of your page. So basically Links to external required files like stylesheets, fonts, js files and etc.

6. **What is the difference between id and class?**
   Id's are unique and each element can only have one ID, and each page can just have one element with that ID. Classes are different because they can be used multiple times in one page, and one element can have multiple classes assigned to it.

   >>Resource

7. **What is the CSS Box Model?**
   CSS Box Model is a box that wraps all the html elements in your page. You can use the box model to design and layout your elements. The Box Model consists in four parts:
   -Margins Adds an area outside the border. The margin is transparent.
   -Borders A border that goes around the padding and content.
   -Padding Adds an area around the content. The padding is transparent.
   -Content (width and height) The content of the box, where text and images appear.

   >>Resource

8. **What is the order of specificity in CSS from general to specific?**
   The lower type is:
   0. Element selectors (e.g., `h1`, `p`, `div`) and pseudo-elements (e.g., `::before`).
   1. Class selectors (e.g., `.example`, `.headings`), attributes selectors (e.g., `[type="radio"]`) and pseudo-classes (e.g., `:hover`).
   2. ID selectors (e.g., `#example`).

Universal selector (*), combinators (+, >, ~, ' ') and negation pseudo-class (`:not()`) have no effect on specificity. (The selectors declared *inside* `:not()` do, however.)

Inline styles added to an element (e.g., `style="font-weight:bold"`) always overwrite any styles in external stylesheets, and thus can be thought of as having the highest specificity.

!IMPORTANT will overwrite it all, But never use it if is not really important.

>>Resource

9. **How does a differ from ?**
   Divs tags are by default block elements, which means that they will always be allocate below the previous element. (it takes up the full width of the parent if not specify.)
   Spans tags are by default inline elements, which means that they will always be allocate on the side of the previous element. (The width would be always justify to the content.)

10. **What is the difference between a relative and absolute path?**
    The difference between relative and absolute paths is that when using relative paths you take as reference the current working directory while with absolute paths you refer to a certain, well known directory. Relative paths are useful when you make some program that has to use resources from certain folders that can be opened using the working directory as a starting point.

    >>Resource

11. **What is Bootstrap?**
    Bootstrap is a *front-end framework* combining JavaScript and CSS to help rapidly prototype website layouts and components.
    Bootstrap saves you time by writing out many standard styles and functions as part of their library so that you don't have to. You can then use these pre-styled components and design functionality with classes defined in Bootstrap. Being proficient in Bootstrap is almost like being proficient in a programming language because you have to know and understand all the different class names to maximize how you can utilize Bootstrap. The more you read and understand the documentation on their website the better you will be with Bootstrap.

12. **What is the DOM? How can the DOM be accessed through Javascript?**
    Document Object Model is a api for HTML. The web browser creates the DOM as it reads the HTML file, and it's used to render to the screen, everything that is parsed from the html file. DOM allow developers to create, build, navigate and modify elements within it.
    We can access the DOM using the javascript object document, that contains

many different methods thats can be used within the DOM object.

13. **What happens when a web browser loads an HTML file?**
    When you load a page in your browser, there is a sequence of requests, responses, processes running, and within those processes there are more subprocesses. One of the processes it's to render and load the HTML file, and when this happen, the sub processes of loading a HTML file are:
    1- Process HTML markup and build the DOM tree.
    2- Process CSS markup and build the CSSOM tree.
    3- Combine the DOM and CSSOM into a render tree.
    4- Run the layout on the render tree to compute the geometry of each node.
    5- Paint the individual nodes to the screen.

14. **What is jQuery?**
    jQuery is a javascript library that makes easier the access to DOM elements, using pre-written functionalities.