**What is semantic markup? Why semantic markup is important? Which tags come with HTML5?**

Semantic markup is a way of writing and structuring your HTML (Hypertext Markup Language) so that it reinforces the semantics, or meaning, of the content rather than its appearance.

* <article>
* <aside>
* <details>
* <figcaption>
* <figure>
* <footer>
* <header>
* <main>
* <mark>
* <nav>
* <section>
* <summary>
* <time>

**How do you do responsive design? What is the difference between responsive and adaptive design? How do you write media queries? How do you test it?**

In creating responsive design you need to:

1. Add responsive meta tags in your HTML document.
2. Apply media queries to your layout.
3. Make images and embedded videos responsive.
4. Ensure your typography will be easily readable on mobile devices.

In the simplest terms, responsive design utilizes one layout and adjusts the content, navigation, and elements of the page to fit the user’s screen. The responsive design will reconfigure all design elements whether it’s viewed on a desktop, laptop, tablet, or mobile phone. With adaptive design, different fixed layouts are created that adapt to the users screen size. As opposed to the more fluid responsive approach, adaptive design employs multiple sizes of a fixed design.

Media queries are useful when you want to modify your site or app depending on a device's general type (such as print vs. screen) or specific characteristics and parameters (such as screen resolution or browser viewport width).

Media queries are used for the following:

* To conditionally apply styles with the CSS @media and @import at-rules.
* To target specific media for the <style>, <link>, <source>, and other HTML elements with the media= attribute.
* To test and monitor media states using the Window.matchMedia() and MediaQueryList.addListener() JavaScript methods.

**What is a Mobile-first approach? Why it is popular? What advantages does it bring?**

Mobile First Approach refers to the practice of designing and/or developing an online experience for mobile before designing for desktop web or any other device. Taking a Mobile First approach aims to reverse the workflow of designing for desktop and scaling down the design for mobile afterwards. Mobile first design is important to consider as you could potentially struggle to remove or compress down information after designing with a lot more space. This means content will either be left out and the user will have much less of an experience, or the smaller display will appear cramped and too full which will severely affect your customer’s experience. However, if you start of small and condense your information down at the beginning of the design stage you will realize it is easier to add more features to your screen as your total space increases.

**How many data types do you know?**

These are the following data type I know:

Integer (int)

Floating Point (float)

Character (char)

String (str or text)

Boolean (bool)

Varchar

Array

Date

Time

Datetime

Timestamp

**Explain how "this" works in JavaScript.**

A function's this keyword behaves a little differently in JavaScript compared to other languages. It also has some differences between strict mode and non-strict mode.

In an object method, “this” refers to the "owner" of the method.

**What is a closure, and how/why would you use one?**

A closure is the combination of a function bundled together (enclosed) with references to its surrounding state (the lexical environment). In other words, a closure gives you access to an outer function’s scope from an inner function. In JavaScript, closures are created every time a function is created, at function creation time.

**What's the difference between == and ===?**

== is used for comparing two variables, but it ignores the data type of variable while === is used for comparing two variables, but this operator also checks data type and compares two values.

**How the browser reads "css"?**

When a browser displays a document, it must combine the document's content with its style information. It processes the document in a number of stages, which we've listed below. Bear in mind that this is a very simplified version of what happens when a browser loads a webpage, and that different browsers will handle the process in different ways. But this is roughly what happens.

1. The browser loads the HTML (e.g. receives it from the network).
2. It converts the HTML into a DOM (Document Object Model). The DOM represents the document in the computer's memory. The DOM is explained in a bit more detail in the next section.
3. The browser then fetches most of the resources that are linked to by the HTML document, such as embedded images and videos ... and linked CSS! JavaScript is handled a bit later on in the process, and we won't talk about it here to keep things simpler.
4. The browser parses the fetched CSS, and sorts the different rules by their selector types into different "buckets", e.g. element, class, ID, and so on. Based on the selectors it finds, it works out which rules should be applied to which nodes in the DOM, and attaches style to them as required (this intermediate step is called a render tree).
5. The render tree is laid out in the structure it should appear in after the rules have been applied to it.
6. The visual display of the page is shown on the screen (this stage is called painting).