**Why we need HTML, CSS and JS?**

HTML provides the basic structure of sites, which is enhanced and modified by other technologies like CSS and JavaScript.

CSS is used to control presentation, formatting, and layout. It makes the websites looks good.

JavaScript is used to control the behavior of different elements in DOM.

JavaScript can be used to modify website content and make it behave in different ways in response to a user's actions.

**What is DOM?**

The **Document Object Model** connects web pages to scripts or programming languages. The DOM model represents a html document with **a logical tree**. DOM methods allow **programs and scripts dynamically access** to the tree elements. with using them you can change the document's structure, style or content. DOM nodes can have event handlers attached to them. Once an event is triggered, the event handlers get executed.

**What is box model?**

All HTML elements can be considered as boxes. It consists of: margins, borders, padding, and the actual content.

**Content**- The content of the box, where text and images appear

**Padding**- Clears an area around the content. The padding is transparent

**Border**- A border that goes around the padding and content

**Margin**- Clears an area outside the border. The margin is transparent

There are actually two types of box model, one is W3C standard, the other is IE model. Basically they all calculate the element width and height but their formula are different.

**What is responsive web design?**

Let’s say we a button on the web page. If we don’t make it responsive on my mobile device, this button could so small for my finger to touch on it. That’s not what we want. That’s why we use responsive design to make sure the view is adjusted based on different devices like laptops, tablets, phones. Make it easy to use. There are four ways to achieve responsive web design. They are viewport, bootstrap 4 Grid system.

Media Query, responsive images.

**What is the difference between Inline, Inline- block and Block?**

**Block** elements always stack on top of each other. Block element will start on a new line and occupy the entire width of its parent element by default. And you can set width and height values. You can also set margin and padding of it.

**Inline** element doesn’t start a new line and only occupy the width of content inside of itself. You can’t set the width or height of inline elements. Top and bottom Margin and padding won’t affect the layout of inline elements but left and right margin and padding will affect layout. If you want to change the size of it, you could change the font-size or change it to inline-block.

**Inline- block** is like combination of inline element and block element. It doesn’t start a new line, but you can set up width, height, margin and padding. One common use for using inline- block for creating navigation bar.

**Absolute, relative,fix**

**Difference between null and undefined**

Undefined means a variable has been declared but has not yet been assigned a value. On the other hand, null is an assignment value. It can be assigned to a variable as a representation of no value. Also, undefined and null are two distinct types: undefined is a type itself (undefined) while null is an object.

**What is callback function in JavaScript?**

A callback function is a function passed into another function as an argument, which is then invoked inside the outer function to complete some kind of routine or action. document. Example: getElementById.addEventListener(“click”, callback function)

**what is this?**

This is the object that is currently executing the function. You can also change this object with using bind, apply and call method. Apply (pass arguments as an array) and call will execute the function but bind will return a function object.

Var person = {

Name : “DiZhu”

sayName: function(){

Console.log(this.name)

}

};

Person.sayName( ); “DiZhu”

**What is Hoisting**

Hoisting is a JavaScript mechanism where variables and function declarations are moved to the top of their scope before code execution.In other words, a variable and a function can be used before it has been declared. For function expression, Javascript will hoist the variable but not the function.

**Difference between In and hasOwnProperty?**

The **in**operator will check specified property is in the object or its prototype chain. The **hasOwnProperty()** will check the object’s own property. This means that in will return true for inherited properties, whereas hasOwnProperty() will return false for inherited properties. When we check an object whether contains tostring method, in would return true and hasOwnProperty would return false.

**Enumerable and iterable?**

For-in loop iterates the keys or properties of an object. For-of loop iterates the value of some objects. These object are array, string, map but not user defined objects. The for-in loop also enumerates prototype properties, while Object.keys() returns only own (instance) properties.

When you are adding properties to an object, the default enumerable value of properties set to be true. This mean you can iterate all the properties using a for-in loop.

**What is IIFE?**

An IIFE is a function expression that is defined and then called immediately to produce a result. That function expression can contain any number of local variables that aren’t accessible from outside that function. They are very useful because they don't pollute the global object, and they are a simple way to isolate variables declarations.

**What is Closure?**

**Closure** is a function plus variables out of its own scope. For example, I want to design an add button. Every time I click the add button the number will increase. Let’s say we have a variable add equals to a self-invoking function. We call this self-invoking function an outer function. The outer function has a variable counter and the outer function will return another function. We call this function inner function. The inner function will make the counter increase by 1. so the inner function is a closure. It access the counter variable in the outer scope. The advantage of using closure is it give the inner function ability to access the outer scope, even after the outer function has returned. Like the example I just mentioned. The immediately invoked function has returned, but inner function can still use outer function’s variable counter.

**Difference Absolute and Relative**

Absolute will make the elements jump out the normal flow of document but relative will make the elements stay in the normal flow. When you set an element’s position absolute, other elements will replace its position. Usually we use relative and absolute positing together. we will set parent element as relative and set child element as absolute, so the we change the position of child element relative to the parent element.