1. **What is the "Box Model" in CSS? Which CSS properties are parts of it?**

The CSS box model is a container that contains multiple properties including borders, margin, padding, and the content. It is used to create the design and layout of webpages.

Properties of the css model-

Content-The content area consists of content like image, text, or other forms of media content. The height and width properties help to modify the box dimensions.

### Padding-The padding area is the space around the content area and within the border-box. It can be applied to all sides of the box selected side(s) - top, right, bottom, and left.

**.box1{**

**border-color:** blue**;**

**}**

### Border-The border area surrounds and can be applied to all the sides of the box or to selected side(s) - top, right, bottom, and left.

**.fcontainer{**

**padding: 15px;**

**border-style: solid;**

**border-color: blue;**

**margin: 20px;**

**width: 100px;**

**height: 100px;**

box-sizing**:** content-box**;**

**}**

### Margin-The margin area consists of space between the border and the margin. The margin does not its own background color It shows the background color of the text

Eg- .box1 {

 padding: 10px;

 border-style: solid black;

  }

1. **Describe what a “reset” CSS file does and how it’s useful. Are you familiar with normalize.css? Do you understand how they differ?**

different web browsers have different default styling for elements, and if you don’t deal with that at all, designers who can be develop website looking not good at different in browsers and more breaking problems

Resetting is meant all default browser styling on texts

Normalizing preserves useful default styles rather than "unstyling" everything incode.

**3. What are the various techniques for clearing floats?**

way to clear floats applying to the element

**floats to create web layouts**. .

<div style=”clear:both;”></div>

**4. What is the difference between inline, inline-block, and block?**

**Block Element:** The block elements always start on a new line. They will also take space for an entire row or width. List of block elements are=== <div>, <p>.

**Inline Elements:** Inline elements don’t start on a new line, they appear on the same line as the content and tags beside them. Some examples of inline elements are ===<a>, <span> , <strong>, and <img> tags.

**Inline Block Elements:** Inline-block elements are similar to inline elements, except they can have padding and margins added on all four sides

**5. Describe z-index and how stacking context is formed.**

Z-index elements with a higher z-index use in front of elements with a lower z-index

It appears so simple, so most of us not take time to read

Every element in an HTML document can be either in front of or behind every other element in the document..

Use for determine this order are in the spec, but as I’ve already stated, they’re not fully understood by most time in code

**6. Style applied to Elements with same id.**

Applying the same id to multiple elements is invalid HTML and should be avoided.

I noticed that when I write id’s value inside single quotation marks (’ ') I get wrong message. Is it mandatory to write everything inside quotation marks (" ") which is because of the matter of Specificity of the elements.

Using too many “id” can cause the document hard to edit that content used in that

For example we have a 3 house, with no colors on the walls,

So we use class to make all walls white by default & we use different id in that “house” so we can add different for those blank or empty rooms

**7. How padding is applied when we provide value with 3 parameters?**

padding: 25px 50px 75px;

top padding is 25px

right and left paddings are 50px

bottom padding is 75px

**8. List down the behaviour of different combination of position values in a parent-child relation element.**

There are multiple combination of index

Relatively positioned-In this elements are offset a given amount from their normal position within the document

<div class="box" id="one">One</div>

<div class="box" id="two">Two</div>

box {

display: inline-block;

width: 100px;

height: 100px;

background: red;

color:redd;

}

Absolute position – element or text remain at same position and not change due to any addtitional part are taken in that

<h1>Absolute positioning</h1>

p{

background: aqua;

border: 3px solid blue;

padding: 10px;

margin: 10px;

.positioned {

position: absolute;

background: yellow;

top: 30px;

left: 30px;

}

In this eg suppose we add p properties in box type then also not changes will occurred in that

Fixed positioning -is similar to absolute positioning, with the exception that the element's

Egg---one {

position: fixed;

top: 80px;

left: 10px;

background: blue;

}

.out of one {

width: 500px;

height: 300px;

overflow: scroll;

padding-left: 150px;

}

Sticky positioning -can be thought of as a hybrid of relative and fixed positioning when it nearest scrolling is viewport. Top right bottom left are some view points off that

**9. List down the pseudo-classes available with the browser compatibility.**

* Active pseudo-class is used to style an element that has been “activated” either by a pointing location on a touchscreen device.

a:active{

color:purplecarpel;

}

* Focus pseudo- pseudo-class is used to style an element that has gained. It’s used a lot in form elements.

a:focus{

color: green;}

* visited- The :visited pseudo-class is used in links that have been visited.

a:visited{

color: blue;

}

* Link-The  pseudo-class represents the “normal” state of links and is used to select links that have not yet been visited .this: :link, :visited, :hover, :active

a:link{

color: orange;}

* Clear--- If you clear a slide of an element, floating elements will not be accepted on that side. With 'clear' set to 'left', an element will be moved below any floating element on the left side. clear is used to stop wrap of an element around a floating element.

Example

<style>

.clearLeft{

clear: left;

}

</style>

<div class="container">

<div class="box"><span>1</span></div>

<div class="box clear Left"><span>2</span></div>

<div class="box"><span>3</span></div>

<div class="box"><span>4</span></div>

<div class="box"><span>5</span></div>

</div>

## Position-absolute, place an element exactly where you want to place it. absolute position is actually set relative to the element's parent. if no parent available then relatively place to the page itself.

## Overflow- overflow property deals with the content if content size exceeds the allocated size for the content. You can make extra content visible, hidden, scroll or auto (viewport default