

CSS Interview Questions Continue

May 15, 2013

Share this :

[Facebook2](#)

[Twitter0](#)

[Pinterest0](#)

[Google+0](#)

Hope you enjoyed the read so far. We've more recommendations for you.

READ : [jQuery Interview Questions and Answers](#)

READ : [HTML5 interview Questions and Answers on DGlobaltech](#)

READ : [Front-end coding Question : Create function to reverse words in a sentence](#)

9. What are some ways you might target IE (or IE6) only, without affecting other browsers?

Below are the example for browser specific Style Sheet, which targets defined browser.

Target ALL VERSIONS of IE

```
<!--[if IE]>
  <link rel="stylesheet" type="text/css" href="all-ie-only.css" />
<![endif]-->
```

Target everything EXCEPT IE

```
<!--[if !IE]><!-->
  <link rel="stylesheet" type="text/css" href="not-ie.css" />
<!--<![endif]-->
```

Target IE 6 ONLY

```
<!--[if IE 6]>
  <link rel="stylesheet" type="text/css" href="ie6.css" />
<![endif]-->
```

READ : [Difference between CSS and CSS3 : CSS2 vs. CSS3](#)

READ : [CSS Interview Questions and Answers for Experienced](#)

READ : [AJAX Interview Questions and Answers : 15 Commonly asked questions](#)

READ : [SQL Interview Questions and Answers : Commonly asked Questions](#)

10. What's the difference between an inline element and a block element?

A block-level element is an element that creates large blocks of content like paragraphs or page divisions. They start new lines of text when you use them, and can contain other blocks as well as inline elements and text or data.

An inline element is an element that define text or data in the document like **STRONG** makes the enclosed text strongly emphasized and **Q** says the enclosed text is a quotation. They don't start new lines when you use them, and they generally only contain other inline tags and text or data. Or they include nothing at all, like the **BR** tag.

Some properties that inline elements ignore include:

- width and height
- max-width and max-height
- min-width and min-height

The CSS property `display` lets you change an inline property to block or a block to inline or either of them to not display at all.

```
display: block;
display: inline;
display: none;
```

When to Change the Display Property

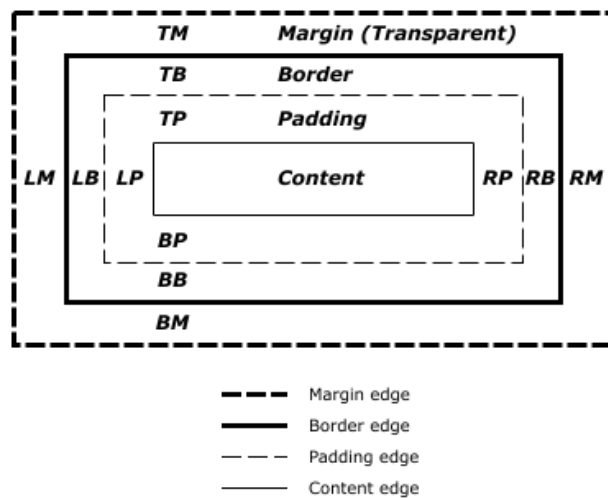
- **Horizontal list menus** - Lists are block-level elements, but if you want your menu to display horizontally, you need to convert the list to an inline element, so that newlines aren't added between each list item.
- **Headers in the text** - Sometimes you might want a header to remain in the text, but have the HTML header values. Changing the `h1-h6` values to inline will let the following text flow next to it.
- **Removing the element** - And of course, if you want to remove the element completely from the document flow, you can set the display to none.

11. Explain how the CSS box model works. Draw a box and then explain what the border is, what the margin is, what the padding is, etc.? If you assign a width: or height: to something, what specifically does that refer to?

All HTML elements can be considered as boxes. In CSS, the term "box model" is used when talking about design and layout.

The CSS box model is essentially a box that wraps around HTML elements, and it consists of: margins, borders, padding, and the actual content.

The box model allows us to place a border around elements and space elements in relation to other elements.



Explanation of the different parts:

- **Margin** - Clears an area around the border. The margin does not have a background color, it is completely transparent
- **Border** - A border that goes around the padding and content. The border is affected by the background color of the box
- **Padding** - Clears an area around the content. The padding is affected by the background color of the box
- **Content** - The content of the box, where text and images appear

Here is the snippet to draw box model to your web page.

```
<head>
<style>
div.test
{
width:220px;
padding:10px;
border:5px solid gray;
margin:0px;
}
</style>
</head>

<body>

<div class="test">The picture above is 250px wide. The total width of this element is also 250px.</div>

</body>
```

12. Explain vertical margin collapsing.

Top and bottom margins of blocks are sometimes combined (collapsed) into a single margin whose size is the largest of the margins combined into it, a behavior known as **margin collapsing**.

Margin collapsing occurs in three basic cases:

Adjacent siblings

The margins of adjacent siblings are collapsed (except when the later sibling needs to be [cleared](#) past floats).

Parent and first/last child

If there is no border, padding, inline content, *or clearance* to separate the `margin-top` of a block with the `margin-top` of its first child block, or no border, padding, inline content, *height*, *min-height*, or *max-height* to separate the `margin-bottom` of a block with the `margin-bottom` of its last child, then those margins collapse. The collapsed margin ends up outside the parent.

Empty blocks

If there is no border, padding, inline content, height, or min-height to separate a block's margin-top from its margin-bottom, then its top and bottom margins collapse.

vertical margins collapse between certain boxes.

To understand how vertical margins collapse lets start with a basic html and CSS example.

Adjacent Elements With Positive Margin

Vertical adjoining margins collapses. If two elements has positive vertically adjoining margin than only the maximum of both will take effect.

CSS Code

```
1 #parent{ background-color:yellow; width:300px; }
2 #red {background-color:red; height:50px; margin-bottom:30px;}
3 #blue {background-color:blue; height:50px; margin-top:20px;}
```

Browser Result

Margins of [floating](#) and [absolutely positioned](#) elements never collapse.

There are other situations where elements do not have their margins collapsed:

- inline-block elements
- elements with `overflow` set to anything other than `visible` (They do not collapse margins with their children.)
- cleared elements (They do not collapse their top margins with their parent block's bottom margin.)
- the root element



13. How do you float an element, and what does that mean? What's Clearing?

With CSS float, an element can be pushed to the left or right, allowing other elements to wrap around it.

Float is very often used for images, but it is also useful when working with layouts.

Elements are floated horizontally, this means that an element can only be floated left or right, not up or down.

A floated element will move as far to the left or right as it can. Usually this means all the way to the left or right of the containing element.

Below is the example, which shows using css float property you can set image to the right of your web page.

```
<head>
<style>
img
{
float:right;
}
</style>
</head>
```

```
<body>
```

```
<p>
```

This is some text. This is some text. This is some text.
This is some text. This is some text. This is some text.
This is some text. This is some text. This is some text.
This is some text. This is some text. This is some text.
This is some text. This is some text. This is some text.
This is some text. This is some text. This is some text.
This is some text. This is some text. This is some text.
This is some text. This is some text. This is some text.
This is some text. This is some text. This is some text.
This is some text. This is some text. This is some text.

□

14. What's the difference between position: relative, position: fixed & position: absolute?

Here, I've listed basic difference between css positioning properties.

Position-Relative. This type of positioning is probably the most confusing and misused. What it really means is "relative to itself". If you set `position: relative;` on an element but no other positioning attributes (top, left, bottom or right), it will no effect on it's positioning at all, it will be exactly as it would be if you left it as `position: static;` But if you DO give it some other positioning attribute, say, `top: 10px;`, it will shift it's position 10 pixels DOWN from where it would NORMALLY be.

Position-Absolute. This is a very powerful type of positioning that allows you to literally place any page element exactly where you want it. You use the positioning attributes top, left bottom and right to set the location. Remember that these values will be relative to the next parent element. If there is no such parent, it will default all the way back up to the `<html>` element itself meaning it will be placed relatively to the page itself.

Position-Fixed. This type of positioning is fairly rare but certainly has its uses. A fixed position element is positioned relative to the *viewport*, or the browser window itself. The viewport doesn't change when the window is scrolled, so a fixed positioned element will stay right where it is when the page is scrolled.

15. What does z-index do?

The z-index property specifies the stack order of an element.

An element with greater stack order is always in front of an element with a lower stack order.

Here is the Example, where using z-index property you can display in front of image.

```
<head>
```

```

<style>
img
{
position:absolute;
left:0px;
top:0px;
z-index:-1;
}
</style>
</head>

<body>
<h1>This is a heading</h1>

<p>Because the image has a z-index of -1, it will be placed behind the text.</p>
</body>

```

READ : [How to prepare for a job interview : Successful face to face interview tricks and tips](#)

READ : [JavaScript Interview Questions and Answers on DGlobaltech](#)

Related Articles on DGlobalTech.com



- [CSS Interview Questions and Answers : 15 Common CSS Questions](#)



- [Drupal Interview questions : Omega Responsive theme tutorial](#)



- [Difference between CSS and CSS3 : CSS2 vs. CSS3](#)