**HTML and CSS Questions**

## 1. Semantic Elements in HTML5

HTML5 offers new semantic elements to define different parts of a web page:

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

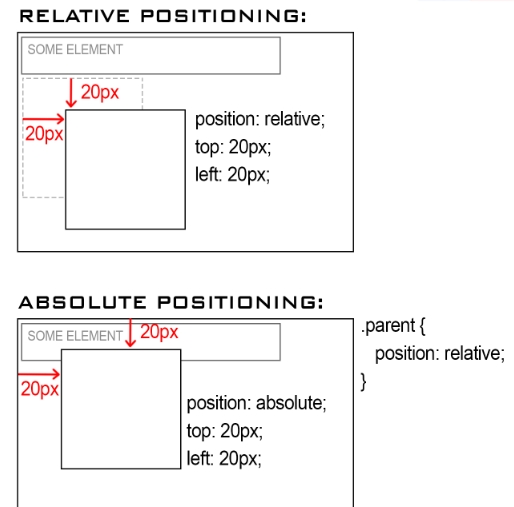
# 2. HTML5 New Elements

|  |  |
| --- | --- |
| **Tag** | **Description** |
| <article> | Defines an article in a document |
| <aside> | Defines content aside from the page content |
| <bdi> | Isolates a part of text that might be formatted in a different direction from other text outside it |
| <details> | Defines additional details that the user can view or hide |
| <dialog> | Defines a dialog box or window |
| <figcaption> | Defines a caption for a <figure> element |
| <figure> | Defines self-contained content |
| <footer> | Defines a footer for a document or section |
| <header> | Defines a header for a document or section |
| <main> | Defines the main content of a document |
| <mark> | Defines marked/highlighted text |
| <menuitem> | Defines a command/menu item that the user can invoke from a popup menu |
| <meter> | Defines a scalar measurement within a known range (a gauge) |
| <nav> | Defines navigation links |
| <progress> | Represents the progress of a task |
| <rp> | Defines what to show in browsers that do not support ruby annotations |
| <rt> | Defines an explanation/pronunciation of characters (for East Asian typography) |
| <ruby> | Defines a ruby annotation (for East Asian typography) |
| <section> | Defines a section in a document |
| <summary> | Defines a visible heading for a <details> element |
| <time> | Defines a date/time |
| <wbr> | Defines a possible line-break |

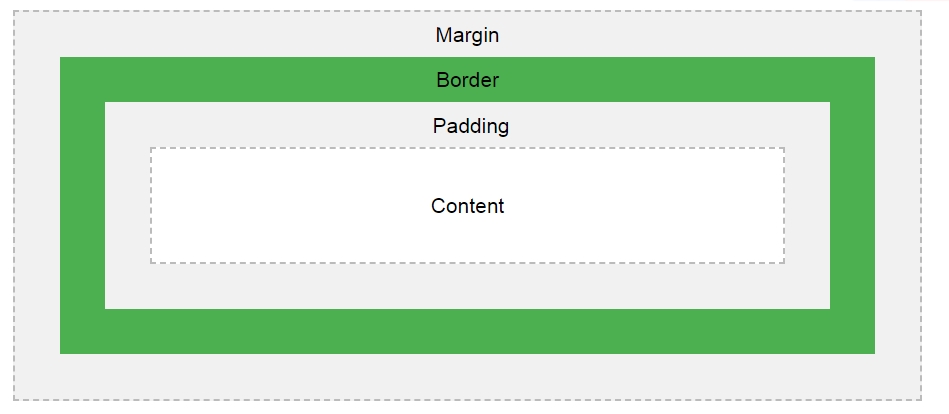
## **3. HTML5 New Input Types**

|  |  |
| --- | --- |
| **New Input Types** | **New Input Attributes** |
| * color * date * datetime * datetime-local * email * month * number * range * search * tel * time * url * week | * autocomplete * autofocus * form * formaction * formenctype * formmethod * formnovalidate * formtarget * height and width * list * min and max * multiple * pattern (regexp) * placeholder * required * step |

1. **Difference between position absolute and relative.**



**5. Boxmodel in css.**



1. **One div browser center style in css.**

#demo {

position: absolute;

width: 100px;

height: 100px;

background-color: blue;

left: 0px;

right: 0px;

top: 0px;

bottom: 0px;

margin: auto;

}

# 7. What is the difference between block and inline-block?

## CSS display: inline

In our first example, we start by using a [<span>](https://www.computerhope.com/jargon/h/html-span-tag.htm) tag, which by default is an inline element. As can be seen in the example below, the span text is red and is contained on the same line (inline) of the text before and after the span.

Example text to give an example of how span text can be as an inline, block, or inline-block element and how it can change the appearance of text.

**HTML code**

<span style="color:red;">span text</span>

## CSS display: block

In the next example, we've changed the default of the <span> tag to display as a block. Because a block element occupies its own line, it gives the appearance that an [enter or return](https://www.computerhope.com/jargon/e/enterkey.htm) was pressed after "how" and "text" in our example.

Example text to give an example of how

span text

can be an inline, block, or inline-block element and how it can change the appearance of text.

**HTML code**

<span style="color:red; display: block;">span text</span>

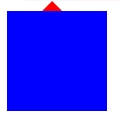
## CSS display: inline-block

Finally, in the next example, we've changed the default of the <span> tag to display as an inline-block. Unlike a block element an inline-block remains inline with all text around the element and appears the same as an inline.

Example text to give an example of how span text can be an inline, block, or inline-block element and how it can change the appearance of text.

**HTML code**<span style="color:red; display: inline-block;">span text</span>

1. **Created div style.**



<div id=’demo’></div>

#demo{

position:absolute;

width: 100px;

height: 100px;

background-color: blue;

}

#demo::after{

position: absolute;

border-left: 10px solid transparent;

border-right: 10px solid transparent;

border-bottom: 10px solid red;

content: '';

top:-10px;

left: 35px;

}

1. **CSS position: sticky; property.**

https://www.w3schools.com/cssref/pr\_class\_position.asp

<https://www.w3schools.com/cssref/tryit.asp?filename=trycss_position_sticky>

The element is positioned based on the user's scroll position

A sticky element toggles between relative and fixed, depending on the scroll position. It is positioned relative until a given offset position is met in the viewport - then it "sticks" in place (like position:fixed).

**Note:** Not supported in IE/Edge 15 or earlier. Supported in Safari from version 6.1 with a -webkit- prefix.

**10. CSS resize property.**

resize: both;  
resize: horizontal;

resize: vertical;

1. **CSS Pseudo-classes**

<a href=”#”></a>

<p>Demo</p>

a:link {}

a:visited {}

a:hover {}

a:active {}

p:first-letter{}

p:first-child{}

p:last-child{}