

# HTML

## 1. What does a doctype do?

- 1) The doctype declaration should be the very first thing in an HTML document, before the tag.
- 2) The doctype declaration is not an HTML tag; it is an instruction to the web browser about what version of the markup language the page is written in.
- 3) The doctype declaration refers to a Document Type Definition (DTD).
- 4) Tell the web browser to expect an HTML element
- 5) In HTML4.01, it refers to DTD since it is based on SGML. But HTML 5.0 is not based on SGML, so it does not require a reference to DTD.  
(Therefore, HTML4.01 needs three declarations and HTML 5.0 only needs one:  
<!doctype html>)

## 2. What is “Semantic HTML?”

- 1) Semantic HTML is the use of HTML markup to reinforce the semantics, or meaning, of the information in webpages and web applications rather than merely to define its presentation or look.
- 2) Semantic HTML is processed by traditional web browsers as well as by many other user agents.
- 3) Semantic markup should not be used to change the way that text looks. Their purpose is to describe the content of your web pages more accurately.

## 3. How many HTML tags are should be used for the most simple of web pages?

8 total. 4 pairs of tags.

< HTML> < HEAD > < TITLE > < /TITLE > < /HEAD > < BODY > < /BODY > < /HTML >

## 4. What is the difference between linking to an image, a website, and an email address?

- 1) To link an image, use <img> tags. You need specify the image in quotes using the source attribute, src in the opening tag. “alt” is the text description of the image when user could not see it. “title” is the additional information about the image.
- 2) For hyperlinking, the anchor tag, <a> , is used and the link is specified in the href attribute. Text to be hyperlinked should be placed between the anchor tags. Little known fact: href stands for “hypertext reference.”
- 3) When linking to an email, the href specification will be “mailto:send@here.com.”

See examples below:

< img src="HTMLrocks.jpg"> < /img>

< a href="www.allindiaexams.in"> All India Exams< /a>

< a href="mailto:admin@allindiaexams.in"> Email Me< /a>

## 5. Consider HTML5 as an open web platform. What are the building blocks of HTML5?

- 1) more semantic text markup
- 2) new form elements
- 3) video and audio
- 4) new javascript API
- 5) canvas and SVG
- 6) new communication API
- 7) geolocation API
- 8) web worker API
- 9) new data storage

## 6. What are some new HTML5 markup elements?

- 1) New Semantic/ Structural Elements: <article>, <aside>, <bdi>, <details>, <dialog>, <figure>, <figcaption>, <footer>, <header>, <main>, <mark>, <nav>, <progress>, <rp>, <rt>, <ruby>, <section>, <summary>, <time>, <wbr>
- 2) New Form Element: <datalist>, <keygen>, <output>
- 3) New input Types: a. Types: color, date, datetime, email, number, url  
b. Attributes: autofocus, autocomplete, form, list, required, min and max
- 4) New Attribute Syntax: a. Empty eg: <input type="text" value="John" disabled>,  
b. Unquoted eg: <input type="text" value=John>,  
c. Double-quoted eg: <input type="text" value="John Doe">,  
d. Single-quoted eg: <input type="text" value='John Doe'>
- 5) HTML5 Graphics: <canvas>, <svg>
- 6) New media element: <audio>, <video>, <source>, <track>

## 7. What are the new media-related elements in HTML5?

- 1) <audio> Define sound content
- 2) <embed> Defines containers for external applications (like plug-ins)
- 3) <source> Defines sources for <video> and <audio>
- 4) <track> Defines tracks for <video> and <audio>
- 5) <video> Defines video or movie content

## 8. What is the difference between SVG and <Canvas>?

Canvas draws 2D graphics, on the fly, via a JavaScript. The <canvas> element is only a container for graphics. You must use JavaScript to actually draw the graphics.

The HTML <svg> element is a container for SVG graphics. SVG is XML based, which means that every element is available within the SVG DOM. You can attach JavaScript event handlers for an element.

In SVG, each drawn shape is remembered as an object. If attributes of an SVG object are changed, the browser can automatically re-render the shape.

Canvas is rendered pixel by pixel. In canvas, once the graphic is drawn, it is forgotten by the browser. If its position should be changed, the entire scene needs to be redrawn, including any objects that might have been covered by the graphic.

Canvas	SVG
<ul style="list-style-type: none"> <li>• Resolution dependent</li> <li>• No support for event handlers</li> <li>• Poor text rendering capabilities</li> <li>• You can save the resulting image as .png or .jpg</li> <li>• Well suited for graphic-intensive games</li> </ul>	<ul style="list-style-type: none"> <li>• Resolution independent</li> <li>• Support for event handlers</li> <li>• Best suited for applications with large rendering areas (Google Maps)</li> <li>• Slow rendering if complex (anything that uses the DOM a lot will be slow)</li> <li>• Not suited for game applications</li> </ul>

## 9. Describe the difference between a cookie, sessionStorage and localStorage.

SessionStorage, localStorage and Cookies all are used to store data on the client side.

The localStorage object stores the data with no expiration date. The data will not be deleted when the browser is closed.

The sessionStorage object is equal to the localStorage object, except that it stores the data for only one session. The data is deleted when the user closes the browser(not the tab).

Cookie: Cookies are data, stored in small text files, on your computer. It stores data that has to be sent back to the server with subsequent requests.

In one word, Cookies are primarily for server-side reading (can also be read on client-side), localStorage and sessionStorage can only be read on client-side.

## 10. Why is it generally a good idea to position CSS <link>s between <head></head> and JS <script>s just before </body>? Do you know any exceptions?

When it comes to ordering your CSS and JavaScript, you want your CSS to come first. The reason is that the rendering thread has all the style information it needs to render the page so that the web page could load progressively.

The problem with putting stylesheets near the bottom of the document is that it prohibits progressive rendering in many browsers, including Internet Explorer.

If the JavaScript includes come first, the JavaScript engine has to parse it all before continuing on to the next set of resources. This means the rendering thread can't completely show the page, since it doesn't have all the styles it needs.

Exception:

- 1) When the JavaScript code has the rendering effect, for example, the Bootstrap library, we need to put the link in the header part.
- 2) Exceptions are if you grab the scripts asynchronously, or at least defer them to the end of the page.

## 11. What is progressive rendering?

Progressive rendering is the name given to techniques used to render content for display as quickly as possible.

It used to be much more prevalent in the days before broadband internet but it's still useful in modern development as mobile data connections are becoming increasingly popular (and unreliable!)

Examples of such techniques :

1) **Lazy loading of images** where (typically) some javascript will load an image when it comes into the browsers viewport instead of loading all images at page load.

2) **Prioritizing visible content** (or above the fold rendering) where you include only the minimum css/content/scripts necessary for the amount of page that would be rendered in the users browser first to display as quickly as possible, you can then use deferred javascript (domready/load) to load in other resources and content.

## 12. What are some of the major new API's that come standard with HTML5?

Media API, Text Track API, Application Cache API, User Interaction, Data Transfer API, Command API, Constraint Validation API, and the History API.

# CSS

## 1. What is the difference between classes and IDs in CSS?

- 1) Unlike the id selector, the class selector is most often used on several elements. This allows you to set a particular style for many HTML elements with the same class. The class selector uses the HTML class attribute, and is defined with a ".".
- 2) A simple way to look at it is that an id is unique to only one element.

## 2. Describe Floats and how they work.

When you float an element it becomes a block box. This box can then be shifted to the left or right on the current line. The markup options are "float: left", "float: right" or "float: none". A floated box is laid out according to the normal flow, then taken out of the flow and shifted to the left or right as far as possible. Content can flow down the right side of a left-floated box and down the left side of a right-floated box. You can also put several floats beside each other. If there isn't enough horizontal room on the current line for the floated box, it will move downward, line by line, until a line has room for it.

Elements following a floated element will wrap around the floated element. If you do not want this to occur, you can apply the "clear" property to these following elements. The four options are "clear: left", "clear: right", "clear: both" or "clear: none".

The float property allows you to take an element in normal flow and place it as far to the left or right of the containing element as possible. Anything else that sits inside the containing element will flow around the element that is floated.

When using the float property, you should also use the width property to indicate how wide the floated element should be. If you do not, results can be inconsistent but the box is likely to take up the full width of the containing element.

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

- 1) If boxes overlap, the elements that appear later in the HTML code sit on top of those that are earlier in the page. The z-index controls which elements sit on top. The higher the number, the closer that element is to the front.
- 2) The stacking context is formed as if the blocks have been stacked on top of each other on a z axis.

The z-index property in CSS controls the vertical stacking order of elements that overlap.

Positioning and assigning a z-index value to an HTML element creates a stacking context.

### **4. Describe Box model.**

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 every HTML element. It consists of: margins, borders, padding, and the actual content.

- 1) Content - The content of the box, where text and images appear
- 2) Padding – the space between the border of a box and any content within it. It can increase the readability of its contents.
- 3) Border - A border that goes around the padding and content, and separates the edge of one box from another.
- 4) Margin – sit outside the edge of the border

### **5. What are your favorite image replacement techniques and which do you use when?**

H5BP Image Replacement 2 -> class = "visuallyhidden"

CSS Image Replacement is beneficial for replacing company logo text with the logo image or when using images along with text for your headers on your web pages. Text in a header tag is more beneficial for SEO(Search Engine Optimization) than images with an alt description.

### **6. How would you approach fixing browser-specific styling issues?**

A lot of cross-browser issues happens when you didn't specify something, and different browsers make different assumptions. So we need to specify everything:

- 1) Declare a valid doctype.
- 2) Use Web standards.
- 3) Validate your HTML and CSS.
- 4) Consider a CSS Reset.
- 5) Use a JavaScript library for DOM interactions
- 6) Test in multiple browsers, deal with IE last.
- 7) Fail Gracefully.
- 8) Check the bare HTML.
- 9) Consider test-driven progressive enhancement.

## 7. What are the different ways to visually hide content (and make it available only for screen readers)?

- 1) visibility: hidden; display: none;  
This style will hide text from all users. The text is removed from the visual flow of the page and is ignored by screen readers.
- 2) width: 0px; height: 0px; font-size: 0px; line-height: 0  
An element with no height or width is removed from the flow of the page. But all of these techniques may result in search engine penalties as they may be interpreted to be malicious.
- 3) text-indent: -10000px;  
This approach moves the content to the left 10000 pixels - thus off the visible screen. Screen readers will still read text with this style. However, if a link or form element is given this style, it may result in a focus indicator that extends from where the element should be located in the page to the place it is actually located.
- 4) position: absolute !important; clip: rect(1px, 1px, 1px, 1px);  
Using CSS to hide or clip content that does not fit into a 1 pixel visible area will essentially hide the content visually, but still allow it to be read by modern screen readers.

## 8. What are some of the “gotchas” for writing efficient CSS?

- 1) Use efficient CSS selectors.  
ID selectors qualified by class and/or tag selectors; Class selectors qualified by tag selectors.
- 2) Avoid a universal key selector.  
Allow elements to inherit from ancestors, or use a class to apply a style to multiple elements.
- 3) Make your rules as specific as possible.  
Prefer class and ID selectors over tag selectors.
- 4) Remove redundant qualifiers.  
These qualifiers are redundant:
  - a. ID selectors qualified by class and/or tag selectors
  - b. Class selectors qualified by tag selectors (when a class is only used for one tag, which is a good design practice anyway).
- 5) only use class for CSS and use Id for Javascript to avoid priority issues.  
use a tab to show a variation of the styling of an element
- 6) group shared properties with selectors
- 7) Avoid using descendant selectors, especially those that specify redundant ancestors.  
For example, the rule `body ul li a {...}` specifies a redundant body selector, since all elements are descendants of the body tag.
- 8) Put CSS in the document head.

## **9. What are the advantages/disadvantages of using LESS preprocessors?**

Advantages:

- 1) Nested syntax
- 2) Ability to define variables
- 3) Ability to define mixins
- 4) Mathematical functions
- 5) Operational functions (such as "lighten" and "darken")
- 6) Joining of multiple files

Disadvantages:

- 1) Debugging is harder
- 2) Compilation time slows down development

## **10. Describe pseudo-elements and discuss what they are used for.**

Pseudo-elements are used to address sub-parts of elements.

A CSS pseudo-element is used to style specified parts of an element.

They allow you to set style on a part of an element's content beyond what is specified in the documents. In other words they allow logical elements to be defined which are not actually in the document element tree. Logical elements allow implied semantic structure to be addressed in CSS selectors.

## **11. List as many values for the display property that you can remember.**

inline, block, flex, inline-block, inline-flex, inline-table, list-item, run-in, table, table-caption, table-row-group, table-column-group, table-header-group, table-footer-group, table-row, table-column, table-cell, none, initial, inherit

## **12. What's the difference between inline and inline-block?**

Inline:

- 1) Respects left and right margins and padding
- 2) Doesn't respect top and bottom margins and padding
- 3) Doesn't have a width and height set
- 4) Allow other elements to sit to its left and right

Block:

- 1) Respects all margins and paddings
- 2) force a line break after the block element

Inline-block:

- 1) Placed like an inline element (on the same line) but behaves as block
- 2) Respects all margins and paddings
- 3) Respects width and height
- 4) Allow other elements to sit to its left and right



### **13. What's the difference between a relative, fixed, absolute and statically positioned element?**

Relative:

Moves an element from its normal position with the top, left, bottom and right CSS properties  
Relative positioning moves an element in relation to where it would have been in normal flow.

Fixed:

- 1) Position in relation to the browser window
- 2) Allows to display toolbars, buttons or navigation menus, which are anchored at a fixed position and scroll with the page
- 3) Fixed positioning is a type of absolute positioning that requires the position property to have a value of fixed.

Absolute:

- 1) Displayed on the page in relation to the document, or a parent element with a positioning different from static
- 2) Using top, left, bottom and right CSS properties, exact position can be specified
- 3) The box is taken out of normal flow and no longer affects the position of other elements on the page

Static:

Means that every element is positioned in the natural order it is added to the page  
In normal flow, each block-level element sits on top of the next one, which is the default way that browsers treat HTML elements.

### **14. The 'C' in CSS stands for Cascading. How is priority determined in assigning styles (provide few examples)? How can you use this system to your advantage?**

Last Rule:

If two selectors are identical, the latter of the two will take precedence.

Specificity:

If one selector is more specific than the others, the more specific rule will take precedence over more general ones.

Important:

You can add !important after any property value to indicate that it should be considered more important than other rules that apply to the same element.

CSS priority is determined by specificity and inheritance.

- 1) Specificity: ID > class, pseudo-class > element, pseudo-element
- 2) Inheritance: specified value → computed value → used value → actual value

### **15. Describe the difference between progressive enhancement and graceful degradation?**

Progressive Enhancement is when you create a web site that is available to everyone, but then add advanced functionality in layers that improves the experience for those who can access it. For instance, a basic text website can be upgraded to include a design (stylesheets), interactive functionality (javascript), and video (flash). But the website continues to function as just a text-based website.

Graceful degradation is an aspect of fault-tolerant systems where your design continues to function even if certain points of the design can't work. The wikipedia article lists HTML as a good example: HTML5 works in all browsers because HTML parsers themselves don't break if there are unrecognized tags. But since older browsers don't recognize those tags, they also can't provide the functionality associated with them (such as the various new input types like range, number, date, time, color, etc.).

The difference between the two is where you start. Progressive enhancement involves starting with a solid base and improving it from there, while ensuring continued operation if the improvements aren't supported. Graceful degradation involves starting with an advanced application, but making sure that if certain elements aren't supported, it continues to work. Obviously, this means there's a crossover point between the two concepts at some point.

**graceful degradation** is the practice of building your web functionality so that it provides a certain level of user experience in more modern browsers, but it will also *degrade gracefully* to a lower level of user experience in older browsers. This lower level is not as nice to use for your site visitors, but it does still provide them with the basic functionality that they came to your site to use; things do not break for them.

**Progressive enhancement** is similar, but it does things the other way round. You start by establishing a basic level of user experience that all browsers will be able to provide when rendering your web site, but you also build in more advanced functionality that will automatically be available to browsers that can use it.

In other words, graceful degradation starts from the status quo of complexity and tries to fix for the lesser experience whereas progressive enhancement starts from a very basic, working example and allows for constant extension for future environments. Degrading gracefully means looking back whereas enhancing progressively means looking forward whilst keeping your feet on firm ground.

## 16. What is flex-box ? How do you vertically center align content with and without flex-box ?

The Flex-box Layout module aims at providing a more efficient way to lay out, align and distribute space among items in a container, even when their size is unknown and/or dynamic.

With -> display: flex; align-items: center;

Without -> display: table-cell; vertical-align: middle; text-align: center;

## 17. What is responsive design all about?

Responsive Web design is the approach that suggests that design and development should respond to the user's behavior and environment based on screen size, platform and orientation. The practice consists of a mix of flexible grids and layouts, images and an intelligent use of CSS media queries.

## 18. Few code exercises. eg: Explain to me what's going on in this CSS selector: [role=navigation] > ul a:not([href^=mailto]) { }

This selects anchor links which meet two requirement:

- 1) are not email links
- 2) are decedents of an unordered list and the unordered list's parent is any element with a role attribute of 'navigation'.

## 19. Create a 4x4 responsive grid.

HTML:

```
<ul class="flex-container">
  <li class="flex-item">1</li>
  <li class="flex-item">2</li>
  <li class="flex-item">3</li>
  <li class="flex-item">4</li>
  <li class="flex-item">5</li>
  <li class="flex-item">6</li>
  <li class="flex-item">7</li>
  <li class="flex-item">8</li>
  <li class="flex-item">9</li>
  <li class="flex-item">10</li>
  <li class="flex-item">11</li>
  <li class="flex-item">12</li>
  <li class="flex-item">13</li>
  <li class="flex-item">14</li>
  <li class="flex-item">15</li>
  <li class="flex-item">16</li>
</ul>
```

CSS:

```
.flex-container {
  padding: 0;
  margin: 0;
  list-style: none;
  display: flex;
  justify-content: space-around
}
.flex-item {
  background-color: red;
  width: 300px;
  height: 300px;
  margin-top: 10px;
  line-height: 300px;
  color: white;
  text-align: center;
}
```

## 20. What benefits and demerits do External Style Sheets have?

Benefits:

- 1) One file can be used to control multiple documents having different styles.
- 2) Multiple HTML elements can have many documents, which can have classes.
- 3) To group styles in composite situations, methods as selector and grouping are used.

Demerits:

- 1) Extra download is needed to import documents having style information.
- 2) To render the document, the external style sheet should be loaded.
- 3) Not practical for small style definitions.

## JavaScript

### 1. What is the difference between == and ===?

== operator means equality. Abstract equality comparison.

=== operator means identity. (strictly to the same value type) Strict equality comparison

### 2. Create a for loop that iterates up to 100 while outputting "fizz" at multiples of 3, "buzz" at multiples of 5 and "fizzbuzz" at multiples of 3 and 5.

```
for( var i = 1; i <= 100; i++) {  
    if(i % 3 === 0 && i % 5 === 0)  
        console.log('fizzbuzz');  
    else if(i % 3 === 0)  
        console.log('fizz');  
    else if(i % 5 === 0)  
        console.log('buzz');  
}
```

### 3. What tools and techniques do you use debugging JavaScript code?

Firebug

console.log

alert()

**4. Some coding question that use the concept of loops, conditions, DOM manipulation, data structures etc. eg. Write a program to sort an array containing only 1's and 0's , by only iterating it once , and not using extra memory.**

```
function sortZeroesAndOnes(nums) {
    var left = 0, right = nums.length - 1;
    while(right > left) {
        while(nums[left] === 0 && right > left)
            left++;
        while(nums[right] === 1 && right > left)
            right--;
        if(nums[left] > nums[right]) {
            var temp = nums[left];
            nums[left] = nums[right];
            nums[right] = temp;
        }
        left++;
        right--;
    }
    return nums;
}
```

## **5. What are JavaScript types?**

There are just six types in JavaScript: Object, Number, String, Boolean, Null, and Undefined. Objects include arrays, functions, and ordinary objects. Numbers may be integers or floating point or the special values NaN and Infinity . Strings include the empty string.Booleans have only two values: true and false. The last two primitive types are a little unusual: the only value whose type is Null is null, and the only value whose type is Undefined is undefined.

## **6. What is 'this' keyword in JavaScript?**

In JavaScript, the thing called 'this', is the object that "owns" the JavaScript code. The value of 'this', when used in a function, is the object that "owns" the function. The value of 'this', when used in an object, is the object itself. The 'this' keyword in an object constructor does not have a value.

"This" always refers to one object, usually the object in which the function operates.

## **7. How generic objects can be created?**

There are two ways to create an object in JavaScript.

```
1) var testObject = new Object();
   testObject.name = "Ming";
   testObject.age = 11;
2) function testObject(name, age) {
   this.name = name;
   this.age = age;
}
var object1 = new testObject("Ming", 11);
```

## 8. Explain window.onload and onDocumentReady?

- 1) The ready event occurs after the HTML document has been loaded, which means the browser has parsed the HTML and built the DOM tree. But the on load event occurs later, when all content ( e.g. Images) also has been loaded.
- 2) The unload event is a standard event in the DOM, while the ready event is specific to jQuery.
- 3) The purpose of the ready event is that it should occur as early as possible after the document has loaded, so that code that adds functionality to the elements in the page doesn't have to wait for all content to load.

## 9. Define event bubbling?

Event bubbling directs an event to its intended target, it works like this: A button is clicked and the event is directed to the button. If an event handler is set for that object, the event is triggered. If no event handler is set for that object, the event bubbles up ( like a bubble in water) to the objects parent.

When an event happens on an element, it first runs the handlers on it, then on its parent, then all the way up on other ancestors.

With bubbling, the event is first captured and handled by the innermost element and then propagated to outer elements.

## 10. How to set a cookie?

```
document.cookie = "username = John Doe; expires = Thu, 18 Dec 2013 12:00:00 UTC; path = /"
```

## 11. How to store objects in local storage?

```
if (typeof(Storage) !== "undefined") {  
    var testObject = {'one' : 1, 'two' : 2, 'three' : 3};  
    localStorage.setItem('testKey', JSON.stringify(testObject));  
} else {  
    alert(Sorry! No Web Storage support..)  
}
```

## 12. Pros and cons of using cookie vs local storage.

Cookies and local storage serve different purposes. Cookies are primarily for reading server-side, local storage can only be read client-side.

Cookies:

Pros:

- 1) Legacy support (it's been around forever)
- 2) Persistent data
- 3) Expiration dates

Cons:

- 1) Each domain stores all its cookies in a single string, which can make parsing data difficult
- 2) Data is unencrypted, which becomes an issue because though small in size, cookies are sent with every HTTP request Limited size (4KB).
- 3) SQL injection can be performed from a cookie.

Local Storage:

Pros:

- 1) Support by most modern browsers
- 2) Persistent data that is stored directly in the browser
- 3) Same-origin rules apply to local storage data
- 4) Is not sent with every HTTP request
- 5) ~5MB storage per domain (that's 5120KB)

Cons:

- 1) Not supported by anything before: IE 8 Firefox 3.5 Safari 4 Chrome 4 Opera 10.5 iOS 2.0 Android 2.0
- 2) If the server needs stored client information you purposefully have to send it.