



- [Go to your profile](#)
- [Hire a developer](#)
- [Apply as a developer](#)
- [Log in](#)

•

- [Top 3%](#)
- [Why](#)
- [Clients](#)
- [Enterprise](#)
- [Community](#)
- [Blog](#)
- [About Us](#)
- [Go to your profile](#)
- [Hire a developer](#)
- [Apply as a developer](#)
- [Log in](#)

- Questions?
- [Contact Us](#)
- 
- 
- 

- Questions?
- [Contact Us](#)
- 
- 
- 

[Hire a developer](#)

## 16 Essential HTML5 Interview Questions \*

- 956shares



•



•



•

[Submit an interview question](#)[Submit a question](#)

Looking for experts? Check out Toptal's [HTML5 developers](#).



What were some of the key goals and motivations for the HTML5 specification?

View the answer → Hide answer



HTML5 was designed to replace both HTML 4, XHTML, and the HTML DOM Level 2.

Major goals of the [HTML specification](#) were to:

- Deliver rich content (graphics, movies, etc.) without the need for additional plugins (e.g., Flash).
- Provide better semantic support for web page structure through the introduction of new structural element tags.
- Provide a stricter parsing standard to simplify error handling, ensure more consistent cross-browser behavior, and simplify backward compatibility with documents written to older standards.
- Provide better cross-platform support (i.e., to work well whether running on a PC, Tablet, or Smartphone).



What are some of the key new features in HTML5?

View the answer →Hide answer



Key new features of HTML5 include:

- Improved support for embedding graphics, audio, and video content via the new [<canvas>](#), [<audio>](#), and [<video>](#) tags.
- Extensions to the JavaScript API such as [geolocation](#) and [drag-and-drop](#) as well for [storage](#) and [caching](#).
- Introduction of “[web workers](#)”.
- Several new semantic tags were also added to complement the structural logic of modern web applications. These include the `<main>`, `<nav>`, `<article>`, `<section>`, `<header>`, `<footer>`, and `<aside>` tags.
- New form controls, such as `<calendar>`, `<date>`, `<time>`, `<email>`, `<url>`, and `<search>`.



What are “web workers”?

View the answer →Hide answer



[Web workers](#) at long last bring multi-threading to JavaScript.

A web worker is a script that runs in the background (i.e., in another thread) without the page needing to wait for it to complete. The user can continue to interact with the page while the web worker runs in the background. Workers utilize thread-like message passing to achieve parallelism.

**Find top HTML5 developers today.** Toptal can match you with the best engineers to finish your project.

[Hire Toptal's HTML5 developers](#)



How do you indicate the character set being used by an HTML5 document? How does this differ from older HTML standards?

View the answer →Hide answer



In HTML5, the encoding used can be indicated with the `charset` attribute of a `<meta>` tag inside the document's `<head>` element:

```
<!DOCTYPE html>
<html>
<head>
...
<meta charset="UTF-8">
...
</head>
...
</html>
```

This is a slightly simpler syntax from older HTML standards, which did not have the `charset` attribute. For example, an HTML 4.01 document would use the `<meta>` tag as follows:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
...
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
...
</head>
...
</html>
```



Discuss the differences between an HTML **specification** and a browser's **implementation** thereof.

View the answer → Hide answer



HTML specifications such as HTML5 define a set of rules that a document must adhere to in order to be “valid” according to that specification. In addition, a specification provides instructions on how a browser must interpret and render such a document.

A browser is said to “support” a specification if it handles valid documents according to the rules of the specification. As of yet, no browser supports all aspects of the HTML5 specification (although all of the major browser support *most* of it), and as a result, it is necessary for the developer to confirm whether the aspect they are making use of will be supported by all of the browsers on which they hope to display their content. This is why cross-browser support continues to be a headache for developers, despite the improved specifications.

In addition, while HTML5 defines some rules to follow for an invalid HTML5 document (i.e., one that contains syntactical errors), invalid documents may contain anything, and it is impossible for the specification to handle all possibilities comprehensively. Thus, many decisions about how to handle malformed documents are left up to the browser.



Briefly describe the correct usage of the following HTML5 semantic elements: `<header>`, `<article>`, `<section>`, `<footer>`.

View the answer → Hide answer



The `<header>` element is used to contain introductory and navigational information about a section of the page. This can include the section heading, the author's name, time and date of publication, table of contents, or other navigational information.

The `<article>` element is meant to house a self-contained composition that can logically be independently recreated outside of the page without losing its meaning. Individual blog posts or news stories are good examples.

The `<section>` element is a flexible container for holding content that shares a common informational theme or purpose.

The `<footer>` element is used to hold information that should appear at the end of a section of content and contain additional information about the section. Author name, copyright information, and related links are typical examples of such content.



Can a `<section>` contain `<article>` elements? Can an `<article>` contain `<section>` elements? Provide usage examples.

View the answer →Hide answer



The answer to both questions is yes; i.e., a `<section>` can contain `<article>` elements, and an `<article>` can contain `<section>` elements.

For example, a personal dashboard page might contain a `<section>` for social network interactions as well as a `<section>` for the latest news articles, the latter of which could contain several `<article>` elements.

Conversely, an `<article>` might contain a `<section>` at the end for reader comments.



Can a web page contain multiple `<header>` elements? What about `<footer>` elements?

View the answer →Hide answer



Yes to both. In fact, both the `<header>` and `<footer>` tags are designed to serve their respective purposes in relation to whatever their parent “section” may be. So not only can the page `<body>` contain a header and a footer, but so can every `<article>` and `<section>` element. In fact, a `<header>` should be present for all of these, although a `<footer>` is not always necessary.



Describe the relationship between the `<header>` and `<h1>` tags in HTML5.

View the answer →Hide answer



In previous specifications of HTML, only one `<h1>` element was typically present on a page, used for the heading of the entire page. HTML5 specifies that `<h1>` represents the top-level heading of a “section”, whether that be the page `<body>`, or an `<article>` or `<section>` element. In fact, every `<header>` element should at least contain an `<h1>` element. If there is no natural heading for the section, it is a good indication it should not use an `<article>` or `<section>` tag.



Give a simple implementation of the <video> tag to embed a video stored at [http://www.example.com/amazing\\_video.mp4](http://www.example.com/amazing_video.mp4). Give the video a width of 640 pixels by 360 pixels. Provide the user with controls.

View the answer → Hide answer



Here is one simple implementation:

```
<video src="http://www.example.com/amazing_video.mp4" width="640" height="360" controls></video>
```

Alternatively, the source file may be indicated with a separate <source> tag inside the <video> element, as in:

```
<video width="640" height="360" controls>
  <source src="http://www.example.com/amazing_video.mp4">
</video>
```



Write the code necessary to create a 300 pixel by 300 pixel <canvas>. Within it, paint a blue 100 pixel by 100 pixel square with the top-left corner of the square located 50 pixels from both the top and left edges of the canvas.

View the answer → Hide answer



Here is one simple implementation:

```
<canvas id="c" width="300" height="300"></canvas>

<script>
  var canvas = document.getElementById( "c" );
  var drawing_context = canvas.getContext( "2d" );
  drawing_context.fillStyle = "blue";
  drawing_context.fillRect( 50, 50, 100, 100 );
</script>
```



What is HTML5 Web Storage? Explain localStorage and sessionStorage.

View the answer → Hide answer



With HTML5, web pages can store data locally within the user's browser.

Earlier, this was done with cookies. However, Web Storage is more secure and faster. The data is not included with every server request, but used ONLY when asked for.

The data is stored in name/value pairs, and a web page can only access data stored by itself. Unlike cookies, the storage limit is far larger (at least 5MB) and information is never transferred to the server.

The difference between localStorage and sessionStorage involves the lifetime and scope of the storage.

Data stored through localStorage is permanent: it does not expire and remains stored on the user's computer until a web app deletes it or the user asks the browser to delete it. sessionStorage has the same lifetime as the top-level window or browser tab in which the script that stored it is running. When the window or tab is permanently closed, any data stored through sessionStorage is deleted.

Both forms of storage are scoped to the document origin so that documents with different origins will never share the stored objects. But sessionStorage is also scoped on a per-window basis. If a user has two browser tabs displaying documents from the same origin, those two tabs have separate sessionStorage data: the scripts running in one tab cannot read or overwrite the data written by scripts in the other tab, even if both tabs are visiting exactly the same page and are running exactly the same scripts.



What is the difference between span and div?

View the answer → Hide answer



The difference is that span gives the output with `display: inline` and div gives the output with `display: block`.

span is used when we need our elements to be shown in a line, one after the other.



What is the Geolocation API in HTML5?

View the answer → Hide answer



HTML5's Geolocation API lets users share their physical location with chosen web sites. JavaScript can capture a user's latitude and longitude and can send it to the back-end web server to enable location-aware features like finding local businesses or showing their location on a map.

Today, most browsers and mobile devices support the Geolocation API. The Geolocation API works with a new property of the global navigator object.

A Geolocation object can be created as follows:

```
var geolocation = navigator.geolocation;
```

The geolocation object is a service object that allows widgets to retrieve information about the geographic location of the user's device.



What's one main result if you do not specify a doctype in an HTML page?

View the answer → Hide answer



New HTML5-specific tags will not be interpreted by the browser.



What's the difference between the <svg> and <canvas> elements?

View the answer → Hide answer



The <svg> element is a container for SVG graphics. SVG has several methods for drawing paths, boxes, circles, text, and even bitmap images.

SVG is a language for describing 2D graphics, but <canvas> allows you to draw 2D graphics on the fly using JavaScript.

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.

\* There is more to interviewing than tricky technical questions, so these are intended merely as a guide. Not every "A" candidate worth hiring will be able to answer them all, nor does answering them all guarantee an "A" candidate. At the end of the day, [hiring remains an art, a science — and a lot of work.](#)

Submit an interview question

Submitted questions and answers are subject to review and editing, and may or may not be selected for posting, at the sole discretion of Toptal, LLC.

---

Name
Email
Enter your question here
Enter your answer here
All fields are required
<input type="checkbox"/> I agree with the Terms and Conditions of Toptal, LLC's <a href="#">Privacy Policy</a>
<input type="button" value="Submit a Question"/>

---

Thanks for submitting your question.

Our editorial staff will review it shortly. Please note that submitted questions and answers are subject to review and editing, and may or may not be selected for posting, at the sole discretion of Toptal, LLC.

Looking for HTML5 experts? Check out Toptal's [HTML5 developers](#).

[View full profile »](#)

[Charles Cook, Ph.D.](#)

United States

Charles has a Ph.D. in aerospace engineering and spent three years developing custom data processing and analysis programs for NASA. He specializes in scalable, enterprise-level application development and engineering solutions for exceptionally high throughputs. He is also the founder and owner of GreatVocab.com, for which he developed the core system using novel concepts in data analysis and control theory.

[HTML5C#JavaScriptASP.NET Web Forms+9 more](#)

[Hire Charles](#)

[View full profile »](#)

[Bryce Ott](#)

United States

With more than fifteen years working as an engineer, architect, director, vice president, and CTO, Bryce brings a deep understanding of enterprise software, management, and technical strategy to any project. His specialties include Amazon Web Services (AWS), real-time systems, business intelligence, big data, enterprise web apps, scalability, education, and open-source software.

[HTML5SQLXMLJavaScript+9 more](#)

[Hire Bryce](#)

[View full profile »](#)

[Alejandro Hernandez](#)

Argentina

Alejandro got his bachelor's degree in software engineering in 2005 and has since been working for software companies of all sizes from all around the globe as a freelancer. Currently, he enjoys working as a full-stack architect in JavaScript projects, where his experience and his deep understanding of architecture and theory are most impactful.

[HTML5JavaScriptNode.jsReact](#)[Hire Alejandro](#)

Toptal connects the [top 3%](#) of freelance talent all over the world.

## Join the Toptal community.

[Hire a developer](#)

or

[Apply as a developer](#)

### Highest In-Demand Talent

- [iOS Developers](#)
- [Front-End Developers](#)
- [UX Designers](#)
- [UI Designers](#)
- [Financial Modeling Consultants](#)
- [Interim CFOs](#)
- [Digital Project Managers](#)

### About

- [Top 3%](#)
- [Clients](#)
- [Freelance Developers](#)
- [Freelance Designers](#)
- [Freelance Finance Experts](#)
- [Freelance Project Managers](#)
- [Freelance Product Managers](#)
- [About Us](#)

### Contact

- [Contact Us](#)
- [Press Center](#)
- [Careers](#)
- [FAQ](#)

### Social



Hire the top 3% of freelance talent™

- © Copyright 2010 - 2019 Toptal, LLC
- [Privacy Policy](#)
- [Website Terms](#)

By continuing to use this site you agree to our [Cookie Policy](#).  
Got it

Find a world-class HTML5 developer for your team. [Hire Toptal's HTML5 developers](#)



