Products Developers AI Agents Pricing

Guide Categories

Search across Guide

Press /

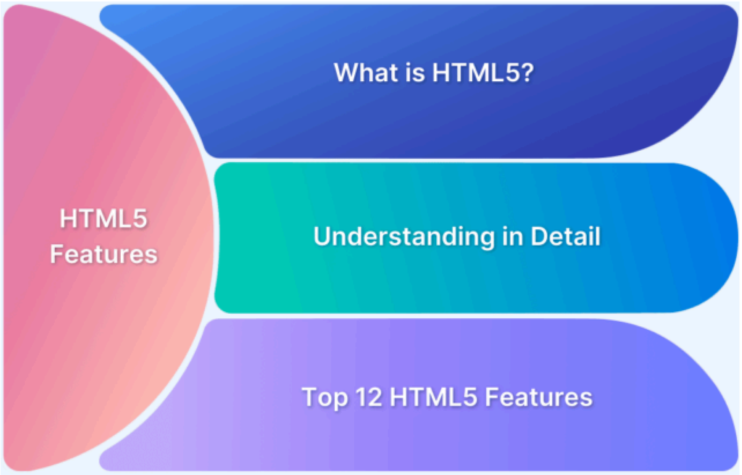
Home Testing on Cloud Debugging Best Practices Tools & Frameworks Tutorials

GET A DEMO Free Trial

Top 12 HTML5 Features in 2024

Understand all the new features of HTML5 and leverage them for better web experience

May 19, 2023 16 min read



Get Started free

Contact Sales

Home > Guide > Top 12 HTML5 Features in 2024

Top 12 HTML5 Features in 2024

HTML stands for Hyper Text markup language, and HTML 5 is the 5th version of it. A lot of dynamic parameters were missing in the HTML for which we have to depend on third-party libraries, however, in the latest version of HTML the trends, 90% of the websites are already using HTML 5 and the latest versions of numerous browsers such as Google Chrome, Safari, Firefox, and more have already established support for HTML5.

Overview

What is HTML 5

- HTML 5 is a sequel to HTML 4.
- It has emphasized its support for responsiveness, graphics, and rendering multimedia elements.
- All modern browsers support HTML 5 in all of their features.

Top 12 Features of HTML5

- Semantic Elements
- Audio and Video Support
- Canvas Elements
- Geolocation API
- Local Storage
- Responsive Images
- Web Workers

- Form Enhancements
- Web Sockets
- Micro Data
- Cross Document Messaging

This article highlights the most important HTML5 features every developer should know.

What is HTML 5

As discussed above, HTML 5 features are already going places after offering web developers more comfort while creating websites. HTML 5 is a sequel to HTML 4 and has never seen such a big update ever before. Moreover, HTML5 has emphasized its support for responsiveness, graphics, and rendering multimedia elements without having to rely upon any external content player like Adobe Flash Player.

Read More: HTML vs HTML5: Core Differences

Understanding HTML 5

HTML 5 gained a tremendous amount of popularity due to the fact that it is even more capable of creating interactive and dynamic web pages. The latest release of HTML, the 5th version, provides developers with greater flexibility

There are several very essential reasons why HTML 5 is vital. Compared to earlier editions, it offers additional functions for you to use. More power than ever over the webpage, it offers a better user experience, cross-device compatibility, SEO, and offline storage. Additionally, it has significantly decreased the need for plugins to play multimedia files like photographs, videos, and audio.

Also Read: How to Test HTML Code in a Browser?

Browser support for HTML5

All modern browsers, including Google Chrome, Mozilla Firefox, Opera Mini, Microsoft Edge, and Apple Safari, support HTML 5 in all of their features. Because HTML 5 capabilities are not adequately supported by earlier browser versions, developers must create alternative content to ensure cross-browser compatibility.

You might perform cross-browser testing for your web apps to make sure everything is working properly and your application is cross-browser compatible.

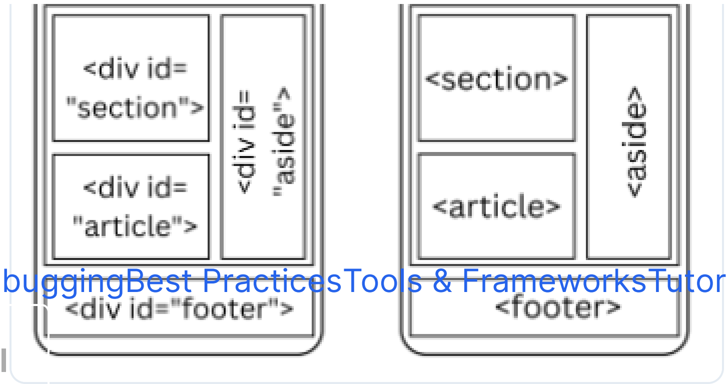
Read More: How to test Browser Compatibility for HTML5

Top 12 Features of HTML5 in 2024

1. Semantic Elements

When constructing an HTML script, the term "**semantic**" refers to the meaning of the text. Semantics, to put it simply, is the use of HTML tags that accurately reflect the content they contain. For instance, using the **<div>** tag doesn't suggest the content it will carry, however using the **<p>** tag suggests it can be used to hold paragraph information.

There are a lot of reasons why you should write Semantic tags instead of normal HTML tags, to leverage SEO, accessibility, and browser compatibility.



```
<!DOCTYPE html>
<html>
<title>Online HTML Editor</title>
<head>
</head>
<body>
<div class="header">
<div class="logo">
<img src="logo.png" alt="Company Logo">
</div>
<div class="menu">
<ul>
<li><a href="#">Home</a></li>
<li><a href="#">Products</a></li>
<li><a href="#">Services</a></li>
<li><a href="#">Contact Us</a></li>
</ul>
</div>
</div>
<div class="main">
<div>It is a paragraph</div>
</div>
<div class="footer">
&copy My website | 2023
</div>
</body>
</html>
```

Now, let’s create an HTML script replacing the normal syntax with semantic syntax.

```
<!DOCTYPE html>
<html>
```

```
<header>
<div class="logo">
<img src="logo.png" alt="Company Logo"></div>
</header>
<nav>
<ul>
<li><a href="#">Home</a></li>
<li><a href="#">Products</a></li>
<li><a href="#">Services</a></li>
<li><a href="#">Contact Us</a></li>
</ul>
</nav>

<section>
<p>It is a paragraph</p>
</section>
<footer>
&copy My website | 2023
</footer>
</body>
</html>
```

2. Audio and Video Support

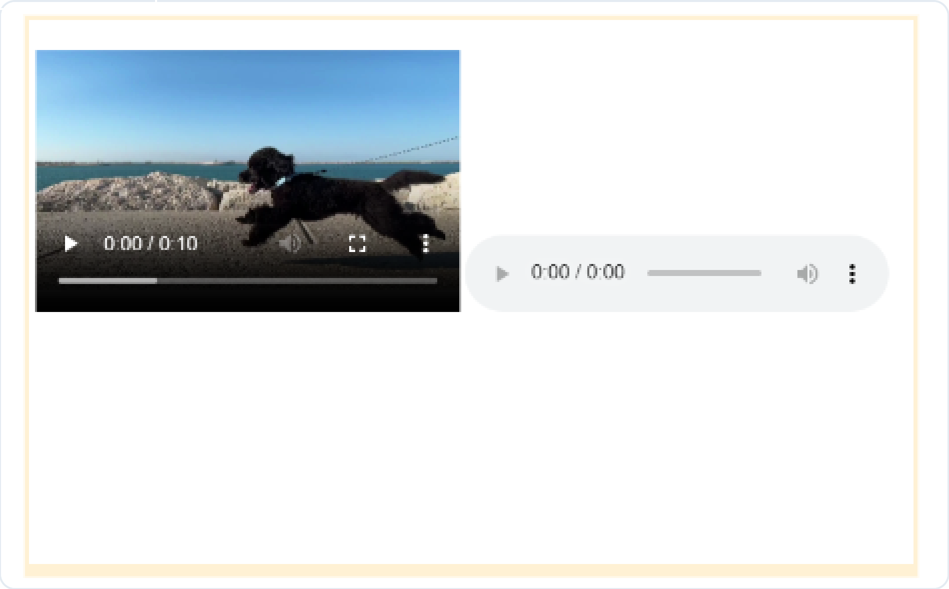
One of the many HTML 5 features is the support for audio and video. It has reduced the hassle of relying upon third-party services such as Adobe Flash player. To embed Audio and Video into your HTML document, you may use the following two tags, **<audio>** and **<video>** tags.

These two tags are launched with the release of HTML 5 and support a numerous range of attributes such as height, width, and more that offers developers to leverage the customization of HTML documents.

Now let’s take an example and add audio and video using these tags to our HTML website.

```
<!DOCTYPE html>
<html lang="en">

<head>
<title>HTML5</title>
```

```
<video width = "300" height = "200" controls autoplay>
<source src = "./dog.mp4" type ="video/mp4" />
</video>

<!-- code to setup audio -->

<audio controls>
<source src="dog.mp3" type="audio/mp3">
</audio>
</body>
</html>
```

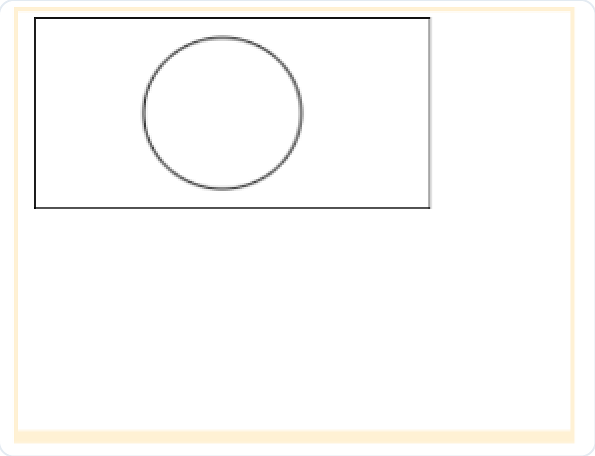


3. Canvas Elements

Canvas Elements is a top-notch feature that has made the tedious task of handling graphics easier for developers. With the help of Canvas elements, you can easily draw graphics using JavaScript. It is optimum for creating simple animations and drawing photo compositions.

To initialize the Canvas element **<canvas>** tag is used with only two specific attributes, width and height. Once, the element is initialized, it is then taken care of with the help of JavaScript to draw shapes, lines, images, and more.

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>HTML5</title>
</head>
<body>
<canvas id="Canvas" width="200" height="100" style="border:1px solid #000000;">Canvas<
</body>
<script>
var c = document.getElementById("Canvas");
var ctx = c.getContext("2d");
ctx.beginPath();
ctx.arc(95,50,40,0,2*Math.PI);
ctx.stroke();
</script>
</html>
```



4. Geolocation API

The Geolocation API is an HTML feature that is used to access the geographical position of a user, however, it is not accessed unless the user approves of it. If you’re wondering where this feature is useful, these come in handy while creating apps like taxi apps, food order tracking apps, fitness tracking apps, and more.

create all these fancy applications.

```
<script>
// check if geolocation is supported
// if supported get the location using .getCurrentPosition method

var x = document.getElementById("demo");
function getLocation() {
if (navigator.geolocation) {
navigator.geolocation.getCurrentPosition(showPosition);
} else {

// if geolocation is not supported display the error.

x.innerHTML = "not supported";
}
}

// showPosition method shows the lattitude and longitude of the location

function showPosition(position) {
x.innerHTML = "Latitude: " + position.coords.latitude +
"<br>Longitude: " + position.coords.longitude;
}
</script>
```

5. Local Storage

It is a modern feature of HTML and several browsers that typically store data in the user's browsers and can access them with the help of JavaScript APIs. This feature is useful for creating offline applications where data is need to be stored locally.

Moreover, using this feature you can reduce the transactions between the application and the backend server, creating a fast application. However, there are some limitations, such as there is limited storage, and you can not access more storage than that.

```
localStorage.setItem("Age", "50");

//Retrieve data from the local storage
document.getElementById("output").innerHTML = localStorage.getItem("Age");
</script>
```

6. Responsive Images

Earlier to create responsive images on the web, you have to rely upon several lines of CSS and sometimes JavaScript, however, HTML 5 makes the process handy by including **srcset** attribute to specify multiple versions of an image at different screen resolutions.

```
There’s one more attribute, the sizes attribute which is used to specify how much s ac

<img
alt="image alt text" src="doggo.jpg" srcset=" small.jpg 240w, medium.jpg 300w, large.j
sizes=" (min-width: 960px) 720px, 100vw "
>
```

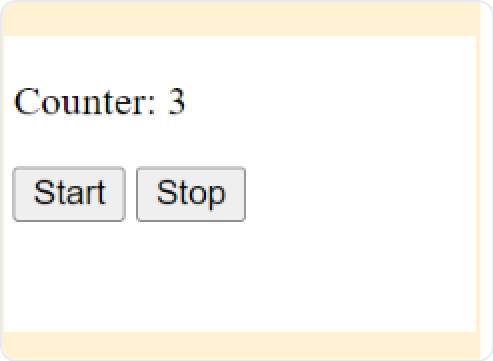
Read More: How to test Responsive Images

7. Web Workers

When you are performing several demanding tasks at once, the browser gets sluggish and responds slowly. Because of this, web workers were added to HTML 5 to allow scripts to operate in the background without interfering with the UI thread.

create a simple counter app that starts the counter when you click **'start'** and stop the counting when you click **'stop'**.

```
<html>
<body>

<p>Counter: <output id="output"></output></p>
<button onclick="start()">Start</button>
<button onclick="stop()">Stop</button>

<script>
var w;

function start() {
if(typeof(Worker) !== "undefined") {
if(typeof(w) == "undefined") {
w = new Worker("demo_workers.js");
}
w.onmessage = function(e) {
document.getElementById("output").innerHTML = e.data;
};
} else {
document.getElementById("output").innerHTML = "Sorry, your browser does not support We
}
}

function stop() {
w.terminate();
w = undefined;
}
</script>

</body>
</html>
```



Counter: 3

Drag and drop is among the most unique features of HTML5 that allow you to grab any element in the DOM and drop it to a different location. To create an element able to drag and drop, set the attribute “**draggable**” on the tag and put its value to true. Let’s understand this process using a practical coding example.

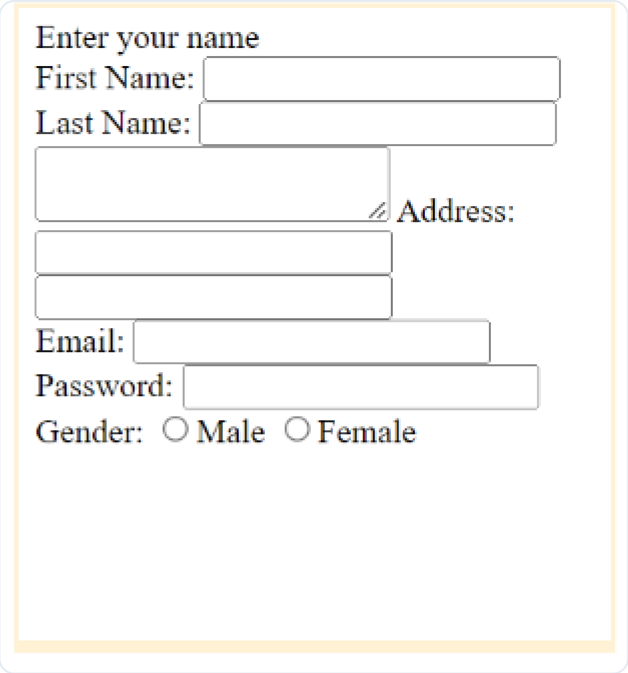
```
<script>
function draggable(ev) {
// Add the target element's id to the data transfer object
ev.dataTransfer.setData("text", ev.target.id);
}

window.addEventListener("DOMContentLoaded", () => {
// Get the element to be dragged by id
const element = document.getElementById("drag");
// add an event listener to make it able to be dragged
element.addEventListener("dragstart", draggable);
});
</script>

<p id="drag" draggable="true">Draggable element</p>
```

9. Form enhancements

HTML 5 introduces new features for your existing forms on HTML to create a more robust user experience. Some of the key enhancements in the new HTMI 5 include new input times such as email, URL, and more, placeholder text, required fields feature, validation, and more.

Let’s demonstrate what’s new in HTML 5 forms with an example where you will test all the new updates.

```
<title>HTML5</title>
</head>
<body>
<form>
Enter your name <br>
First Name: <input type="text" name="firstname"/> <br/>
Last Name: <input type="text" name="lastname"/> <br/>
<textarea rows="2" cols="20"></textarea>
<label for="firstname">Address: </label> <br/>
<input type="text" id="firstname" name="firstname"/> <br/>
<input type="text" id="lastname" name="lastname"/> <br/>
<label for="email">Email: </label>
<input type="email" id="email" name="email"/> <br/>
<label for="email">Password: </label>
<input type="password" id="password" name="password"/> <br/>
<label for="gender">Gender: </label>
<input type="radio" id="gender" name="gender" value="male"/>Male
<input type="radio" id="gender" name="gender" value="female"/>Female <br/>
</form>
</body>
</html>
```



10. Web Sockets

In the previous versions of HTML, when a client sends a request to the backend server, the server then responds afterward. However, in HTML 5 we

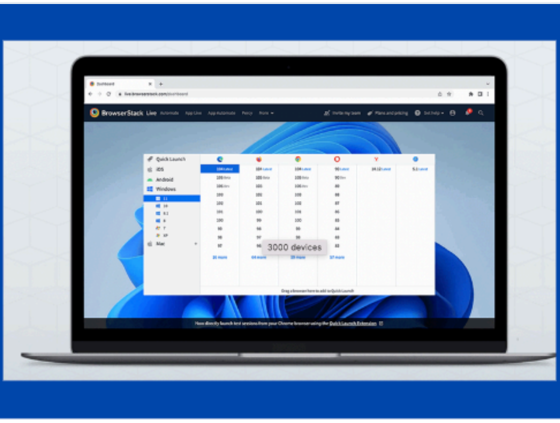
It has its usage in large-scale applications due to its capability of handling larg amounts of connections simultaneously between the server and the client suc as online gaming applications, live trading, and other live low-latency applications.

```
<script>

// create a new WebSocket object
var socket = new WebSocket('ws://localhost:8080');

// handle socket events
socket.onopen = function() {
console.log('WebSocket connection opened');
};

socket.onmessage = function(event) {
console.log('WebSocket message received:', event.data);
};

socket.onclose = function(event) {
console.log('WebSocket connection closed:', event.code, event.reason);
};

socket.onerror = function(error) {
console.error('WebSocket error:', error);
};

// send a message to the server
socket.send('Hello, server!');

</script>
```

11. Micro Data

Micro Data in simpler words is a further deeper level to provide semantics to your webpage. This feature is used to structure data in HTML documents by embedding your own customized elements.

1. **itemref**,
2. **itemid**,
3. **itemscope**,
4. **itemprop**, and
5. **itemtype**



These attributes are used to give the HTML elements more context to define the role they play in the webpage. Microdata may be used to specify a variety of structured data, such as details about events, items, companies, and more. Here is an example of how microdata can be used in HTML.

```
<div
itemprop="aggregateRating"
itemscope
itemtype="https://schema.org/AggregateRating">
RATING:
<span itemprop="ratingValue">10</span> (
<span itemprop="ratingCount">8.4</span> ratings )
</div>
```

In general, web browsers don’t let web pages from different domains influence each other, this is done for several security reasons, however, even if the web pages don’t intend to intrigue the privacy or harm the other web page. There are several scenarios, in which you want to access cross-document communication to make your web page more interactive with less effort.

Some of these scenarios are when you want to embed data from third-party sites such as maps, photos, videos, and more. This can be achieved in HTML 5 with the feature of cross-document messaging, to enable real-time communication between different parts of websites opened in different frames, windows, etc.

```
Let’s understand cross-document messaging with a practical example. For this to exp ai

otherWindow.postMessage(message, targetOrigin);

window.addEventListener("message", receiveMessage, false);

function receiveMessage(e) {
// Verify sender's identity and origin
if (e.origin !== "http://example.com") return;

// Access the message data
console.log(e.data);
}
```

Conclusion

The introduction of additional new features in HTML 5 has increased the dependability of web development for the creation of dynamic and interactive websites. The top 12 features of HTML 5 have been covered in this article, including some very important ones like embedding video and audio, utilizing capabilities have also made it possible for web developers to rely less on outside services.

Guide Categories

Press /

Home Testing on Cloud Debugging Best Practices Tools & Frameworks Tutorials

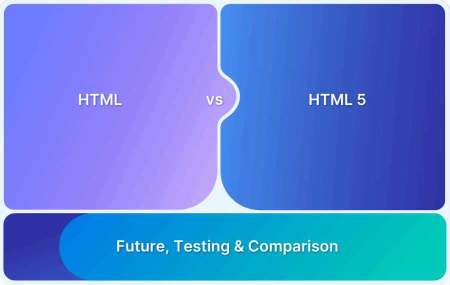
TAGS

Automation Testing

Website Testing

GET A DEMO Free Trial

Related Guides



HTML vs HTML5: Core Differences

Understand the core difference between...

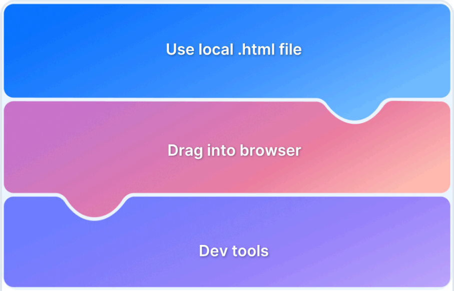
October 29, 2024
11 min read

HTML5 Standard & Purpose
Browser Compatibility for HTML5
Handling Cross Browser Compatibility

How to test Browser Compatibility for HTML5

Ensure seamless HTML5 performanc...

March 21, 2025
13 min read



How to Test HTML Code in a Browser?

Learn how to test HTML code in...

August 1, 2025
11 min read

AI Agents

Pricing

Guide Categories

Press /

HomeTesting on CloudDebuggingBest PracticesTools & FrameworksTutorials

GET A DEMO Free Trial

Get answers on our Discord Community

Join our Discord community to connect with others! Get your questions answered and stay informed.

Join Discord Community

Test on 3500+ Real Devices & Browsers

Try BrowserStack Live to test on 3500+ real Devices and Browser combinations under real world conditions for accurate test results and first hand user-like experience

Contact Sales

AI Agents Pricing

Guide Categories

Press /

Home Testing on Cloud Debugging Best Practices Tools & Frameworks Tutorials

GET A DEMO Free Trial

PRODUCTS

Live

Testing Toolkit

App Automate

App Live

Automate

Accessibility Testing

Low Code Automation

Percy

App Accessibility Testing

Accessibility Automation

Test Management

Test Management for Jira

Test Reporting & Analytics

App Percy

Requestly

Website Scanner

WHY BROWSERSTACK

Customers

Case Studies

AI Agents

Browsers & Devices

Enterprise

Data Centers

Real Device Features

Security

RESOURCES

Support

Status

Release Notes

Blog

Events

Community

BrowserStack Talks

Meetups

Champions

Guide

Partners

Find a partner

Trust Center

Test University (Beta)

COMPANY

About Us

Careers

Open Source

Press

Newsletter

SOCIAL

Contact Us

We are available 24 / 7

More Resources Cross Browser Testing • Selenium • Test Management • Emulators vs Real Device • Mobile App Testing

Test On Devices Test on iPhone • Test on iPad • Test on Galaxy • Test In IE • Test on Android • Test on iOS • Test on Right Devices • Mobile Emulators

Tools SpeedLab • Screenshots • Responsive • Nightwatch.js

AI Agents Pricing

Guide Categories

Press /

HomeTesting on CloudDebuggingBest PracticesTools & FrameworksTutorials

GET A DEMO Free Trial