

Html5

Atrayee Chatterjee¹ Karan Patadia² Harshad Raitani³ Jyoti Lahoti⁴

Assistant Professor, Department Of Bca, The Heritage Academy, Kolkata-700107, West Bengal,India.

Student, Department Of Bca The Heritage Academy, Kolkata-700107, West Bengal,India.

Student, Department Of Bca The Heritage Academy, Kolkata-700107, West Bengal,India.

Student, Department Of Bca The Heritage Academy, Kolkata-700107, West Bengal,India.

Corresponding Author: Atrayee Chatterjee

ABSTRACT

Web technology is a standard that allow developing web applications with the help of predefined sets of classes, objects, methods and properties available in a markup language, style sheet language, or programming language. HyperText Markup Language (HTML) is the main markup language for web pages. HTML elements are the basic building-blocks of webpages. HTML standard were and are created by World Wide Web consortium (W3C). HTML5 is the W3C's next major revision to HTML, which it started developing in 2004. HTML5 is not some new language or development tool. It is just HTML with an extended layer of standardized tags and attributes for graphic and visual effects that reduces the need for special plug-ins.

KEYWORDS: Html5, Geolocation, Mobile, Media Elements, Flash Rivalry, Current Downsides

Date of Submission: 20-05-2019

Date of acceptance:03-06-2019

I. INTRODUCTION

HTML5 is the fifth and current major version of HTML standard, and subsumes of XHTML. By itself, HTML focuses on presenting static documents, meaning text, links and little else. Through HTML 4, the HTML standard was one in the same with the HTML syntax used to accomplish this objective. The HTML5 standard has a much broader goal to describe the content, styling and application interfaces behind a Web page when it's loaded in your browser. More specifically, HTML5 establishes a single syntax for interacting with all the elements that Web page has loaded into your computer's memory. How does the browser load a page written according to the newer HTML5 standard and not the long-standing HTML 4 language? After all, we usually just rely on our browsers to load Web pages without worrying about the HTML behind them. The answer is simple: As long as the browser itself can support HTML5, it can handle anything you throw at it without needing to distinguish between HTML revisions.

To make this magic happen, the HTML5 standard integrates new versions of multiple technologies working together toward common goals. These goals include the following:

- Keeping the language simple and intuitive
- Ensuring the code is easy to read and maintain
- Addressing pages as interactive applications rather than static documents
- Relying on Cascading Style Sheets (CSS) for styling the content
- Recognizing JavaScript as a central scripting component for Web pages
- Embracing dynamic content from server-side technologies such as PHP and ASP

What Can We Do With Html5?

HTML has evolved a lot since the days when frames were still cool and was everyone's favourite HTML tag, which became annoying very quickly. Today's sites are created using JavaScript, Cascading Style Sheets (CSS), Flash, AJAX and a whole lot of other technologies that make the web more responsive than ever before. HTML5 will make all of the features of today's web pages much simpler to implement and also add a whole set of extra features as well.

As an example of just how powerful some of the new features of HTML5 are, three Google engineers ported the famous first-person shooter (FPS) game,

—Quake II, entirely into HTML5 code. All of the 3D graphics, networking, local game saving, and other features have

entirely been written in HTML code with some JavaScript.

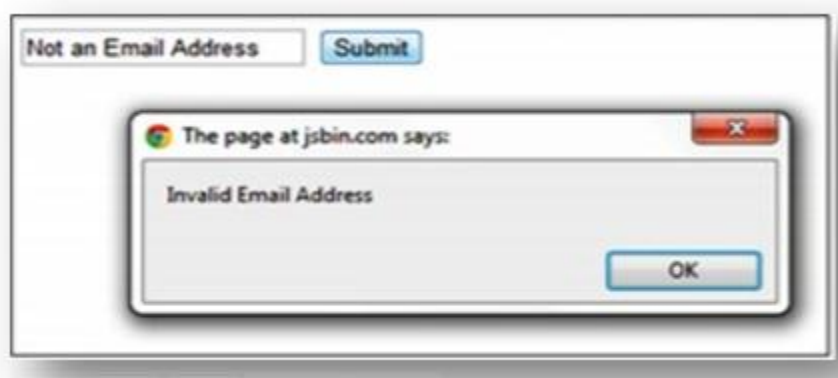
The following is an example that we can see on the internet everyday – A simple form for entering email addresses and using JavaScript code to ensure that given Email is valid or not.

```

<!DOCTYPE html>
<html>
  <head>
    <script language="JavaScript">
      function validate(form_id,email) {
        var reg = /^[A-Za-z0-9_\-\.]+\@[A-Za-z0-9_\-\.]+\.[A-Za-z]{2,4}$/;
        var address = document.forms[form_id].elements[email].value;
        if(reg.test(address) == false) {
          alert('Invalid Email Address');
          return false;
        }
      }
    </script>
  </head>
  <body>
    <form id="form_id" method="post" onsubmit="javascript:return validate('form_id','email');">
      <input type="text" id="email" name="email" />
      <input type="submit" value="Submit" />
    </form>
  </body>
</html>

```

Code for a simple form that lets users enter valid email addresses



Page that uses the code for the simple form validation above

While the above code satisfies our need, it can become untidy if we have to make a form that requires a name, an email address, a telephone number, a birth date, and a home page, all of which will need separate JavaScript codes to validate that is not ideal.

Let us look at the following code for the same page using some of HTML5's features.

```

<!DOCTYPE html>
<html>
  <head></head>
  <body>
    <form id="form_id" method="post">
      <input type="email" name="email" />
      <input type="submit" value="Submit" />
    </form>
  </body>
</html>

```

Code for the same page that uses some HTML5 features



Page that uses the code for the simple form validation with HTML5

The page has the same functionality (i.e., validating the email address) but with much tidier code and

no JavaScript in sight. The validation warning is a lot nicer looking than the pop-up message seen in the previous example. How did this happen? It is actually quite simple. In the previous example, we told the browser to put an input text field that users can enter data into. The browser happily rendered the text field for us, then we used JavaScript to ensure that the text users enter are actually email addresses. HTML5 has a new input type called `email`, which tells the browser we want users to be able to enter email addresses. The browser will render this to exactly look like a normal text field that understands that the input should be an email address and that does all of the validation for us. A host of other new types such as `tel`, `url`, `date`, and `number` also. We are only scratching the surface of what HTML5 lets us do.

Features

Canvas: HTML5 element gives the user an easy and powerful way to draw graphics using JavaScript. The `<canvas>` tag is used to draw different graphics. The `html <canvas>` tag is used for :

- Drawing Lines and curves
- Creating color gradients
- Canvas Scaling
- Canvas Rotation

Geolocation: The HTML Geolocation API is used to locate a user's address. The function `getCurrentPosition()` is used to retrieve the user's address and since this can compromise someone's privacy, the position would not be displayed until the user approves it. The HTML5 Geolocation element is not supported for all the browsers.

Drag & drop: HTML5 comes with a Drag and Drop (DnD) API that brings native DnD support to the browser and thus making it much easier to support on devices such as mobile phones. This includes dragging of content and files from outside the browser – Drag and Drop to upload files or photos.

Web Storage: Web storage provides a way for websites to store the information and retrieving it later on the computer. There is Local Storage which stores data with no time limit and there is Session Storage which stores data for one session only. Though it is similar to cookies, it is designed for larger quantities of information.

Native Audio & Video Controls: HTML5 supports Built-in media via

`<audio>` and `<video>` elements and so offering the ability to easily add media into HTML documents. IT also allows for video to play without plug-ins or other software, so you can see videos on iPhones and iPads.

Web Workers: Web Workers are basically a API specification that lets you create background JavaScript threads to process CPU intensive tasks. Normally in browsers a single thread is created to handle all the JavaScript code. So whatever JavaScript code is run in the browser, all of them is executed in one single thread; whether you are doing some calculation or updating page elements.

Web Sockets: Web Sockets is a technique for two-way communication over one (TCP) socket, a type of PUSH technology. Web sockets can replace long polling. This is an interesting concept; the client sends a request to the server – now rather than the server responding with data it may not have, it essentially keeps the connection open until the fresh, up-to-date data is ready to be sent – the client next receives this, and sends another request.

New Semantic Elements: HTML is a descriptive or `semantic` markup language, which means it labels logical, structural parts to a site in its code. In HTML4, a lack of well-structured semantic elements like `<section>` or `<footer>` meant that it was very difficult for search engines to know what was what within a webpage. HTML5's new semantic elements, like `<header>`, `<nav>`, and `<article>` make the page's sections crystal clear.

Html5 And Mobile

Mobile technology is becoming more popular these days. The adoption of Mobile Devices continues to grow very rapidly and in turn the user will be using mobile more as time goes. HTML5 is the most mobile ready tool for developing mobile sites and apps. With Adobe announcing the death of mobile Flash, HTML5 will counted upon more for Web application development.

HTML5 is incredibly supportive of mobile development. Mobile browsers have fully adopted HTML5 so creating mobile ready projects is as easy as designing and constructing for their smaller touch screen displays — hence the popularity of Responsive Design. The new expandable navigation and collapsible menus that were not available in previous version of HTML makes it easier to build in feature on a mobile site. There are some great Meta tags that also allow you to optimize for mobile:

- **Viewport:** allows you to define viewport widths and zoom settings
- **Full screen browsing:** IOS specific values that allow Apple devices to display in full screen mode
- **Home Screen Icons:** like favicons on desktop, these icons are used to add favourites to the home screen of an IOS and Android mobile device

Developers for Web application targeted at mobile platforms like Apple iOS and Android have raved about the

benefits they can gain by programming those apps in HTML5. In January 2011, Facebook application developer Cory Ondrejka posted a note in the company’s engineering blogpraising HTML5’s potential and describing how the company was experimenting with the technology as a gaming platform. Ted Woodbury of AT&T cautioned that HTML5 still had some painful limitations, though it had the potential to overcome those as the standard became more refined. Only time will tell to what extent mobile apps will embrace HTML5, but evidence today suggests that developers are enthusiastic about its future. [Sources: Ondrejka, Gohring]

The major benefits of using HTML5 for mobile application development are the portability across mobile platforms and an easier testing, deployment, and distribution process, because applications do not need to be installed on mobile devices. The distribution aspect, especially, can be challenging for distributed teams operating in edge environments. However, the major drawback is that the APIs to access the mobile devices’ hardware and on-board sensors are limited for use in edge applications.

HTML5 compatibility on mobile and tablet browsers with testing on real devices:

| Feature | Safari iOS | Android Browser | Samsung Internet | Google Chrome | Amazon Silk | BlackBerry Browser | Nokia Browser | Internet Explorer | Opera Mobile | Opera mini | Firefox | |
|--|-------------|--------------------|------------------|---------------|---------------|----------------------|------------------|----------------------------|-------------------|------------------|-----------------|------------|
| Platform | Phone, iPad | Phones & Tablet | Android devices | Android 4.0+ | Kindle Fire | Phones, BB10, Tablet | Nokia X, Symbian | Windows Phone, Windows 8.x | Android & Symbian | Java iOS Android | Android, Meeko* | Firefox OS |
| Network Information API W3C API Old spec Connection, Bandwidth | | ✓ 2.2+ Old Spec | | ✓ 30+ | ✓ Old Spec | ✓ | | | | | ✓ 12+ | ✓ |
| XMLHttpRequest 2.0 W3C API XHR 2.0: upload files, progress | ✓ 8.0+ | ✓ 3.0+ | ✓ | ✓ | ✓ Partial | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CORS W3C API Cross-Origin Resource Sharing, for cross domain AJAX requests | ✓ | ✓ 3.0+ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Server-Sent Events W3C API Event-driven pattern to maintain the connection to the server open | ✓ 4.1+ | | ✓ | ✓ | ✓ 2.0+ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Web Sockets W3C API Bidirectional protocol over HTTP | ✓ 4.2+ | | ✓ | ✓ | ✓ 2.0+ | ✓ 6.1+ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Media Capture Stream (getUserMedia) W3C API Camera access for video element | | | ✓ | ✓ | | ✓ 10.0 only | | | ✓ | ✓ 12.x only | ✓ | ✓ |
| WebRTC W3C API Real-time communication | | | ✓ | ✓ | | | | | ✓ | ✓ 12.x only | ✓ | ✓ |

| Feature | Safari iOS | Android Browser | Samsung Internet | Google Chrome | Amazon Silk | BlackBerry Browser | Nokia Browser | Internet Explorer | Opera Mobile | Opera mini | Firefox | |
|---|----------------|----------------------|------------------|---------------|-------------|----------------------|------------------|--------------------------|-------------------|--------------------|-----------------|---------------|
| Platform | Phone, iPad | Phones & Tablet | Android devices | Android 4.0+ | Kindle Fire | Phones, BB10, Tablet | Nokia X, Symbian | Windows Phone, Windows 8 | Android & Symbian | Java iOS Android | Android, Meeko* | Firefox OS |
| WebGL Khronos Group API 3D Canvas for the web | ✓ 8.0+ | ✓ Specific device | ✓ | ✓ | | ✓ | ✓ | ✓ | ✓ | ✓ 12+ (Android) | ✓ | ✓ |
| Navigation Timing API W3C API Performance events for VFPD | ✓ 9.0+ | ✓ 4.0+ | ✓ | ✓ | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| File API W3C API Opening local files through input type | ✓ 8.0+ | ✓ 3.0+ | ✓ | ✓ | ✓ 2.0+ | ✓ | ✓ | ✓ | ✓ | ✓ 12+ (partial) | ✓ | ✓ |
| File System API W3C API Virtual File System for persistent storage | | | ✓ | ✓ | ✓ 2.0+ | ✓ | ✓ | | ✓ | ✓ | | |
| HTML Media Capture W3C API Taking pictures, record video and audio from an input file type | ✓ 8.0+ | ✓ 3.0+ | ✓ | ✓ | | ✓ | ✓ | | ✓ | ✓ | ✓ | ✓ |
| Web Speech API W3C API Speech Recognition and Synthesizer | ✓ 7.0+ | | | ✓ | | | | | ✓ | ✓ | ✓ | |
| Home Screen Webapp NO API Add icon to the home screen with fullscreen button | ✓ meta tags | | | ✓ | ✓ 3.2+ | | ✓ App.js | ✓ file file | ✓ | ✓ | ✓ manifest | ✓ manifest |

| Feature | Safari iOS | Android Browser | Samsung Internet | Google Chrome | Amazon Silk | BlackBerry Browser | | | Nokia Browser | | Internet Explorer | | Opera Mobile | Opera mini | Firefox | |
|---|--------------|-----------------|------------------|---------------|-------------|--------------------|------|--------|---------------|---------|-------------------|------------|-------------------|------------------|----------------|------------|
| Platform | iPhone, iPad | Phones & Tablet | Android Devices | Android 4.0+ | Kindle Fire | Phones | BB10 | Tablet | Nokia X | Symbian | Windows Phone | Windows 8+ | Android & Symbian | Java iOS Android | Android, Meego | Firefox OS |
| CSS 3 Basic W3C Standard Backgrounds, text effects, rounded corners | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | 6.0 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | partial | ✓ | ✓ |
| CSS 3 Transforms 2D W3C Standard rotate, translate, scale, skew, matrix | ✓ | 2.0+ | ✓ | ✓ | ✓ | ✓ | 6.0 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | partial | ✓ | ✓ |
| CSS 3 Transforms 3D W3C Standard rotate3d, translate3d, Perspective, Backface | ✓ | 3.0+ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | 11+ | ✓ | 14+ | ✓ | 11+ | ✓ |
| CSS 3 Transitions W3C Standard Animations between two states | ✓ | 2.0+ | ✓ | ✓ | ✓ | ✓ | 6.0 | ✓ | ✓ | ✓ | 10+ | ✓ | ✓ | ✓ | ✓ | ✓ |
| CSS 3 Animations W3C Standard Animations with keyframes | ✓ | 2.0+ | ✓ | ✓ | ✓ | ✓ | 6.0 | ✓ | ✓ | ✓ | 10+ | ✓ | 12.1+ | ✓ | ✓ | ✓ |
| CSS 3 Regions W3C Standard Content flowing between different elements | ✓ | 7.0+ | ✓ | 30+ flag | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | 10+ | ✓ | 17+ flag | ✓ | ✓ | ✓ |
| Position: fixed support W3C Standard Ability to fix an element fixed in the viewport while scrolling / zooming | ✓ | 2.2+ | ✓ | ✓ | ✓ | 7.0+ | ✓ | ✓ | ✓ | ✓ | 10+ | ✓ | 14+ | ✓ | Partial 11+ | ✓ |
| Position: sticky support W3C Standard Element remains until it goes out of the viewport when it gets fixed | ✓ | 6.0+ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | 14+ | ✓ | 25+ | ✓ |

| Feature | Safari iOS | Android Browser | Samsung Internet | Google Chrome | Amazon Silk | BlackBerry Browser | | | Nokia Browser | | Internet Explorer | | Opera Mobile | Opera mini | Firefox | |
|---|--------------|-----------------|------------------|---------------|-------------|--------------------|-------------|------------|---------------|--------------|-------------------|------------|-------------------|------------------|----------------|------------|
| Platform | iPhone, iPad | Phones & Tablet | Android Devices | Android 4.0+ | Kindle Fire | Phones | BB10 | Tablet | Nokia X | Symbian | Windows Phone | Windows 8+ | Android & Symbian | Java iOS Android | Android, Meego | Firefox OS |
| Versions tested | 3.2 to 9.0 | 1.5 to 4.3 | 1.0 to 1.6 | 18 to 40b | 1.0 to 2.0 | 6.0 to 7.1 | 10 to 10.2b | 1.0 to 2.1 | 1.0 | ~3 to S60FP2 | 9 to 11 | 10 to 11 | 11 to 20 | 9 to 7.5 | 9 to 34b | 1.0 |
| Application Cache W3C API Offline package installation | ✓ | 2.1+ | ✓ | ✓ | ✓ | 6.0+ | ✓ | ✓ | ✓ | ✓ | 10+ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Web storage W3C API Persistent and session storage | ✓ | 2.0+ | ✓ | ✓ | ✓ | 6.0+ | ✓ | ✓ | ✓ | ✓ | 10+ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Web SQL storage W3C API (obsolete) Persistent SQL-like storage | ✓ | 2.0+ | ✓ | ✓ | ✓ | 6.0+ | ✓ | ✓ | ✓ | ✓ | 10+ | ✓ | ✓ | ✓ | ✓ | ✓ |
| IndexedDB W3C API Asynchronous database system (replacement for Web SQL) | ✓ | 6.0+ | ✓ | ✓ | ✓ | 2.0+ | ✓ | ✓ | ✓ | ✓ | 10+ | ✓ | 14+ | ✓ | ✓ | ✓ |
| Geolocation W3C API Determination & tracking using GPS, cell or Wi-Fi | ✓ | ✓ | ✓ | ✓ | 2.0+ | 6.0+ | ✓ | ✓ | ✓ | ✓ | 10+ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Multimedia W3C API Video & Audio Players | ✓ | 2.3+ | ✓ | ✓ | ✓ | 7.0+ | ✓ | ✓ | ✓ | ✓ | 10+ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Web Workers W3C API Threaded and background process communications | ✓ | 5.0+ | ✓ | ✓ | ✓ | 6.0+ | ✓ | ✓ | ✓ | ✓ | 10+ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Viewport definition W3C API Viewport support | ✓ | dpi | ✓ | dpi | ✓ | dpi 7.0+ | dpi | ✓ | dpi 400+ | ✓ | also css | also css | also css | 8+ | ✓ | ✓ |

| Feature | Safari iOS | Android Browser | Samsung Internet | Google Chrome | Amazon Silk | BlackBerry Browser | | | Nokia Browser | | Internet Explorer | | Opera Mobile | Opera mini | Firefox | |
|--|--------------|-----------------|------------------|---------------|-------------|--------------------|------|--------|---------------|---------|-------------------|------------|-------------------|------------------|----------------|------------|
| Platform | iPhone, iPad | Phones & Tablet | Android Devices | Android 4.0+ | Kindle Fire | Phones | BB10 | Tablet | Nokia X | Symbian | Windows Phone | Windows 8+ | Android & Symbian | Java iOS Android | Android, Meego | Firefox OS |
| Web Audio API W3C API Low-level audio playing | ✓ | 6.0+ | ✓ | 20+ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | 10+ | ✓ | 25+ | ✓ |
| Notifications API W3C API Background alert notifications | ✓ | ✓ | ✓ | 40b+ | 2.0+ | 10.1+ | 2.0+ | ✓ | ✓ | ✓ | ✓ | ✓ | 14+ | ✓ | ✓ | ✓ |
| Service Workers W3C API Background processing / offline | ✓ | ✓ | ✓ | 43 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | 14+ | ✓ | 33+ flag | ✓ |
| Animation Timing API W3C API Performance hints for HTML5 animations | ✓ | 6.0+ | ✓ | ✓ | 2.0+ | ✓ | ✓ | 2.1+ | ✓ | ✓ | 10+ | ✓ | 14+ | ✓ | 11+ | ✓ |
| Fullscreen API W3C API Allows the application to get a full screen mode | ✓ | 2.0b+ | ✓ | ✓ | 20+ | ✓ | ✓ | ✓ | ✓ | ✓ | 11+ | 11+ | 14+ | ✓ | Partial | Partial |
| Page Visibility API W3C API Determines current visibility state | ✓ | 7.0+ | ✓ | ✓ | ✓ | ✓ | ✓ | 2.1+ | ✓ | ✓ | ✓ | ✓ | 12.1+ | ✓ | 18+ | ✓ |
| Battery Status API W3C API Information about current battery and charging status | ✓ | ✓ | ✓ | 30+ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | 12.1+ | ✓ | old spec | old spec |
| Ambient Light Events W3C API Information about current ambient lighting status in Luminance | ✓ | ✓ | ✓ | 40b+ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | 12.1+ | ✓ | 18+ | ✓ |
| Vibration API W3C API Vibrate a mobile device for notification | ✓ | ✓ | ✓ | 32+ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | 14+ | ✓ | 18+ | ✓ |
| Remote Debugger Ability to attach a remote debugger, even as Web Inspector | ✓ | 6.0+ web | ✓ | ✓ | ✓ | 7.0+ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | DragonFly | ✓ | 14+ | ✓ |

| Feature | Safari iOS | Android Browser | Samsung Internet | Google Chrome | Amazon Silk | BlackBerry Browser | | | Nokia Browser | | Internet Explorer | | Opera Mobile | Opera mini | Firefox | |
|---|-------------|-----------------|------------------|---------------|-------------|--------------------|------|-----------|---------------|-----------------|---------------------|-----------------|-------------------|------------------|----------------|------------|
| | Phone, iPad | Phones & Tablet | Android Devices | Android 4.0+ | Kindle Fire | Phones | BB10 | Tablet | Nokia X | Symbian | Windows Phone | Windows 8.x | Android & Symbian | Java iOS Android | Android, Meego | Firefox OS |
| Canvas API W3C API 2D Drawing API | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| SVG W3C Working Group Scalable Vector Graphics | ✓ | ✓ 3.0+ | ✓ | ✓ | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Motion Sensors W3C Standard Accelerometer, Gyroscope, Magnetometer | ✓ 4.2 | ✓ 3.0+ | ✓ | ✓ 3.0+ | ✓ 2.0+ | | ✓ | ✓ | ✓ | | ✓ 11+ | ✓ 11+ | ✓ 12+ | | ✓ | ✓ |
| Form Virtual Keyboards W3C Standard Text inputs with different keyboards | ✓ | ✓ 4.0+ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ Anna+ | ✓ 10+ | ✓ | ✓ | | ✓ | ✓ |
| Form New Controls W3C API All: Range, Date, Time, Month, Range | ✓ 5.0+ | | ✓ | ✓ | | ✓ 8.0+ | ✓ | ✓ 2.0+ | ✓ | | ✓ 10+ only range | ✓ only range | ✓ | | ✓ | no range |
| Touch Events W3C API touchstart, touchend, touchmove, touchcancel | ✓ | ✓ 2.1+ | ✓ | ✓ | ✓ | ✓ 8.1+ | ✓ | ✓ | ✓ | ✓ Belle FP2+ | | | ✓ (android) | | ✓ | ✓ |
| Pointer Events W3C API pointerdown, pointerup, pointermove, etc. | | | | | | | | | | | ✓ 10+ | ✓ | | | | |

Media Elements And The Flashrivarly

One of the most favourite features of HTML5 is the new approach it took to embed media on a Web Page. HTML5 does not require an external browser plugin for the embedding. The following are the media elements available inHTML5:

- **audio** -- embeds audio in the page and includes attributes for specifying how to play it; supported file formats will vary between Webrowsers
- **video** -- embeds video in the page and includes attributes for specifying how to play it; supported file formats will vary between Webrowsers
- **source** -- used with audio or video elements to identify a source; multiple sources could be specified for a singleitem
- **embed** -- embeds and specifies the media type for content that might lack support within other media elements
- **canvas** -- sets aside part of the page, or the computer screen, where JavaScript can draw images; coming up, we'll find out more about how this important new featureworks
- **svg**-- embed vector graphics encoded with SVG markup language, allowing them to be scaled dynamically to the area of the page in which they're loaded without losing any graphicquality

We communicate in an increasingly mobile way, and because Flash cannot outfit the mobile demands of modern business, it is trailing behind in the rat race. The HTML5 vs. Flash debate is disappearing from the global conversation.

Where new elements were added to enable users to manipulate layout for greater SEO, certain esoteric CSS qualifiers were eliminated to increase ease of use. All in all, HTML5 functions with greater audio and video capabilities, and has reached a state of totalinteroperability.

Though Adobe Flash is more familiar to developers and is adopted by the vast majority of legacy browsers it requires Adobe Flash Player plug-in to operate that crashes frequently, has security issues and newer browsers are no longer supporting the Flash Player and also the mobile browsers support has been dropped altogether. is incompatible with iOS and fumbles withStage3D.

HTML5 is SEO friendly and an open-source language that can support Flash-like technologies, evolving interface, Adobe Canvas allows users to draw graphics, make photo collages, animations, and real-time video rendering with JavaScript, less maintenance required, secure, IOS and Android compatible, interoperability, requires less processing power because of itslight weight framework, storage options. The only issue for HTML5 is that App Stores does not acknowledge HTML5, has fewer offline capabilities and less than ideal gaming functionality and also it is not fully supported by all major browsers.

The conclusion is that HTML5 will eventually surpass Adobe Flash, and we cannot deny it. The facts are there. Steve Job's public condemnation of Flash for its inferiority to HTML5 in a publication titled, "Thoughts on Flash," initiated Flash's dissent toward obsolescence. IOS's incompatibility with Flash and YouTube's shift from Flash to HTML5 for supporting videos tipped the scale further in HTML5's direction.

Current Downsides

HTML5 specification is not yet completed. It will take longer to evolve since it is a combination of standards that are developed by different groups.

No one could agree on what the standard video support should be within HTML5. That means there is so many different video supports out there today that are based on the browser you prefer to use. There are three primary video formats currently used: Ogg Theory, H.264, and VP8/WebM. The first is supported by everything except Internet Explorer. The second is supported by everything except Firefox. As for the third, it is fully supported by everything, though it may require a manual installation. It requires modern browsers to access it.

JavaScript is the only scripting language of HTML5. It is a very capable language, ideal for numerous applications. From a gaming perspective, however, there is a lack of features which are necessary for a strong gaming experience. Custom name spaces, member access, interfaces, and inheritance all struggle under JavaScript. There are plenty of work-arounds available which are suitable to get your work done. It is not, however, a first-choice language option from a purely gaming standpoint.

The HTML5 standard was evolved in a relatively short period and this produced discrepancies in the implementation of CSS attributes, HTML tags and JavaScript APIs. Sometimes these features behave differently on different platforms. However, there are tools available to resolve these discrepancies.

Future Of Html5

HTML5 is rapidly advancing and will continue to evolve. While HTML5 competes directly against technologies like Flash, Flex and Silverlight, it seems to be working its way into everything. Currently, it may not be capable of creating as impressive a user experience as its competitors, but the sheer audience it will have access to will make it a serious player. HTML5 still will not totally solve the mobile device fragmentation. However, it will act as a strong catalyst to increase convergence of the market, as it sets a new standard with many features, and the first signs indicate that this standard is moving much faster than any previous attempts made by the W3C alone, thanks to the WHATWG.

To get the most out of your mobile site, achieve the widest reach and get the best experience on each mobile device you would still need specialized solutions that can adjust media rendering based on the device and/or browser recognition and are able to adapt the level of HTML5 features based on the current available browser support.

II. CONCLUSION:

HTML5 has been intended to deliver almost everything a user would desire to do. There's more to it. It has already replaced Adobe flash plugin in diverse grounds like video playing and in gaming concepts. Though it is not a completed version but still it has made its way to the top by replacing Flash. Whether it's through a framework, via an application wrapper or as the basis for a mobile web app, HTML5 is going to continue to be an important driving force for mobile application development. In fact, as the technology evolves, we wouldn't be surprised to see more HTML5 elements popping up in native desktop applications as well.

The drawback of the language is that although parts of the language are very stable, the language itself is considered a work in progress therefore any of the elements could change at any time which can complicate things further.

REFERENCES:

- [1]. http://www.ijceronline.com/papers/Vol2_issue5/AR02514301437.pdf
- [2]. <https://www.aspiresys.com/WhitePapers/Html5.pdf?pdf=html5-whitepaper>
- [3]. <https://www.upwork.com/hiring/development/advantages-of-html5/>
- [4]. <http://www.trendmicro.it/media/misc/html5-attack-scenarios-research-paper-en.pdf>
- [5]. <https://computer.howstuffworks.com/html-five1.htm>
- [6]. <https://www.w3.org/TR/2000/REC-DOM-Level-2-Core-20001113/>
- [7]. <https://tympanus.net/codrops/2011/11/24/top-10-reasons-to-use-html5-right-now/>
- [8]. <https://www.aspiresys.com/WhitePapers/Html5.pdf?pdf=html5-whitepaper>
- [9]. <https://www.upwork.com/hiring/development/advantages-of-html5/>

Atrayee Chatterjee" Html5"International Journal of Computational Engineering Research (IJCER), vol. 09, no. 6, 2019, pp 01-07