

FEATURES OF HTML5 AND ITS FUTURE TRENDS

PANKAJ KUMAR, RESEARCH SCHOLAR, B. R. AMBEDKAR BIHAR UNIVERSITY,
MUZAFFARPUR, BIHAR, INDIA
(pankaj@glug4muz.org)

RAJESH KUMAR, RESEARCH SCHOLAR, B. R. AMBEDKAR BIHAR UNIVERSITY,
MUZAFFARPUR, BIHAR, INDIA
(rajeshk_muz@yahoo.in)

A. K. SINGH, READER, P. G. DEPARTMENT OF MATHEMATICS, B. R. AMBEDKAR BIHAR
UNIVERSITY, MUZAFFARPUR, BIHAR, INDIA
(ajaypunamsingh@gmail.com)

ABSTRACT

This paper highlights the new trends of HTML coding to design and develop a web page. The latest version of HTML is HTML5 which consists of lot of new features from the point of view of design and functionality of web pages. Some of the newly added features in HTML5 are drawing canvas, detecting browser geographical location, displaying web pages in offline mode, increasing the amount of data to be stored on the client computer, improving input form functionality and a lot of semantic tags to improve the semantic of content of web page. Although all new features are not supported in all browsers but their support in most of the modern browsers is increasing very fast. This paper discusses the features of new elements as well as their support and implementation in modern browsers along with its future trends.

Keywords: HTML5, Canvas, geolocation, localStorage, offline browsing, manifest

INTRODUCTION

HTML is the native markup language for design and development of web pages. All Internet Browsers are nothing but the interpreter of programs written in this language. The latest version of this language is HTML5 in which several new elements have been introduced to improve the look and feel as well as functionality of web pages to be developed in this language. One of the main purpose of adding these new elements is to improve the semantic of web pages such that the meaning of content of the web pages can be easily understood by search engines. All newly added elements to this language can be categorized into three categories according to their nature.

The first category of elements can be called semantic elements whose main purpose is to enhance the meaning of content of web page. Some of the important semantic elements added in HTML5 are section, article, nav, header, footer, time. Although there is no change in visual appearance of text enclosed in these tags but the use of these tags enhance the meaning of content written within these tags. However the visual effects of content of these tags can also be enhanced by defining selectors for these tags in the CSS code.

Another category of newly added elements can be those elements which are having effects on the visual appearance on the content of the page but their functionality does not depend upon the availability of JavaScript support in the browser. Such tags belong to the integration of audio and video in the web page as well as form to take input from the user. In fact in case of tags related to the form, attributes of input tag has been enhanced. Some new values have been introduced for the "type" attribute of the input tag as well as some new attributes have been defined for input tag.

The last category of the newly added elements can be those elements whose appearance and functionality depends upon the availability of the JavaScript support in the browser. Such tags can be used

to draw images on canvas, detecting browser geographical location, displaying web pages in offline mode, increasing the amount of data to be stored on the client computer and providing the functionality of drag and drop like interface on the web page.

RESEARCH METHODOLOGY

This paper provides the syntax and usages of all three types of elements added in HTML5. For the first category of elements since there is no effect on the visual appearance of the web page and so only the syntax and semantic of these elements have been discussed but in case of second and third category of elements screen shot of output of web pages have also been provided in addition to the syntax and semantic of the elements.

Elements having semantic effects only: Several new elements have been added to HTML5 for whom there is no effect on the appearance of content on the web page. But they enhance the meaning of the web page significantly. Following is the list of important semantic elements added to HTML5:

A. header: This tag has been introduced to display the content as a header on the web page. Traditionally HTML provides six different tags for display of content as header on the web page namely H1, H2, H3, H4, H5 and H6. These tags are having effects on the visual appearance of content to be displayed as header. But the size of the content to be displayed in the H5 and H6 tags so much small that generally these two tags are being used to display the content as footer of the web page. In addition to this the whole content to be displayed as header may be enclosed in any one of these tags or combination of these tags. So it always becomes difficult to recognize the actual content which has been displayed as header of the page. In order to solve this problem generally an ID is defined in the CSS code whose name is header and then all combination of header tag is enclosed within a div tag which usages that CSS ID as follows:

```
<div id="header">
  <h1>This is the most important header</h1>
  <h2>This is less important header</h2>
  ...
  ...
</div>
```

This is the most common pattern of writing headers in HTML but it is not necessary that the name of CSS ID will always be declared as header. Its name can be anything as per the choice of the coder. So there is no semantic value from the point of view of header. User agents of web page never infer any meaning from the value of id attribute of div tag. So in order to solve this problem the same code can be written in HTML5 as follows:

```
<header>
  <h1>This is the most important header</h1>
  <h2>This is less important header</h2>
  ...
  ...
</header>
```

Now in this case the design of the header can be controlled from CSS by defining a selector with the name header and writing the same set of rules that were written for the selector whose name was #header. The main thing is that in this case the name of the tag is header and the word "header" is not the value of attribute id of div tag. So the user agents will be able to know the fact that the content written within the tag whose name is "header" contains the header of the web page. This improves the semantic value of the web page significantly.

B. hgroup: The meaning of this tag is header group. It is used to make sub groups of content written in the header section of the page. So this tag is always used as nested of header tag.

C. article: This tag has been used to display an article on the web page. An article can belong to a blog, a news or anything else. But in the previous version of HTML there was no standard tag to identify a paragraph as an article. So user agents were not able to identify the article text available on the web page to

identify them as article. This tag improves the semantic value of a web page from the point of view of article recognition.

D. time: When an article is posted on the web page then a particular date-time is associated with it as the posting date-time of that article. In the previous version of HTML there was no tag available to recognize the posting date-time of an article. The time tag has been introduced in HTML5 to declare the publication date of an article. The format of this tag is as follows:

```
<time datetime="yyyy-mm-ddThh:mm:ss">Display Text</time>
```

The interesting thing about this tag is that it can be used in two formats. When it is used inside the article tag then it implies that the date-time represented by it corresponds to that article. However when it is used outside the article tag then it implies that the date-time represented by it corresponds to the whole web page.

E. nav: Classes and IDs of CSS have been traditionally used to manage the menu of web page in the following format:

```
<div id='nav'>
<ol>
<li><a href=#>Menu1</a></li>
<li><a href=#>Menu2</a></li>
...
...
</ol>
</div>
```

Although it works well from the point of view of functionality but there is no semantic value from the point of view of the fact that these links have been used to manage the navigation bar of the web page. In order to declare the fact that these links have been used to manage the navigation bar of the web page, the nav tag has been introduced by using which the above code can be rewritten as follows:

```
<nav>
<ol>
<li><a href=#>Menu1</a></li>
<li><a href=#>Menu2</a></li>
...
...
</ol>
</nav>
```

F. footer: This tag has been introduced to replace the use of CSS ID with name #footer which has been traditionally used to manage the content of the footer of the web page. Its usage is very similar to the use of header tag.

Elements having no dependency on JavaScript: In HTML5 three new tags have been introduced for which there is no need of availability of JavaScript support in the browser. These tags have been explained below:

A. audio: Traditionally audio on the web page has been integrated with the help of object tag in the form of flash player. In HTML5 with the help of audio tag, an audio file can be integrated with the backward compatibility feature as follows:

```
<audio controls='controls'>
<source src='test.ogg' type='audio/ogg'>
<source src='test.mp3' type='audio/mp3'>
...
//object tag here
</audio>
```

In this case if the audio tag is supported in the browser then it will first of all try to play the file “test.ogg”. If this file will be played successfully then other source tags as well as the object tag will be skipped. If not then the browser will try to play the file test.mp3. If all files specified in the source tags will be skipped then the audio will be played with the help of object code.

B. video: This tag has been introduced on the same pattern of audio tag and its use is similar to the audio tag. Inside the video tag, a list of source tags is written followed by the object tag. The video file specified by the object tag is played when all files specified by the source tags are skipped by the browser.

C. form: The form tag has been used traditionally to take input from the user on web pages. In HTML5 some new input types have been introduced as well as values of attributes of existing input tag have also been enhanced. For example, for text input, two new attributes have been introduced namely autofocus and placeholder. Traditionally the autofocus feature is implemented using the following JavaScript code:

```
<body onload="document.forms[0].elements[0].focus();">
```

But now the same functionality can be achieved with the help of following code:

```
<input type=text autofocus>
```

Similarly we can put some text in the text box when the text box is empty and the focus of the user is not in this box with the help of following code:

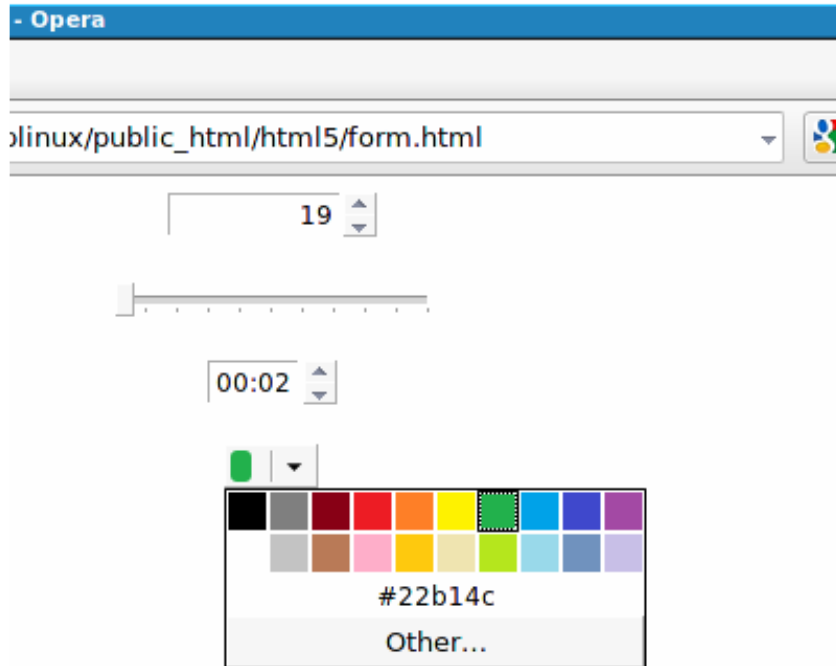
```
<input type=text placeholder="Write your name here">
```

In addition to this some new input types have also been introduced as written below:

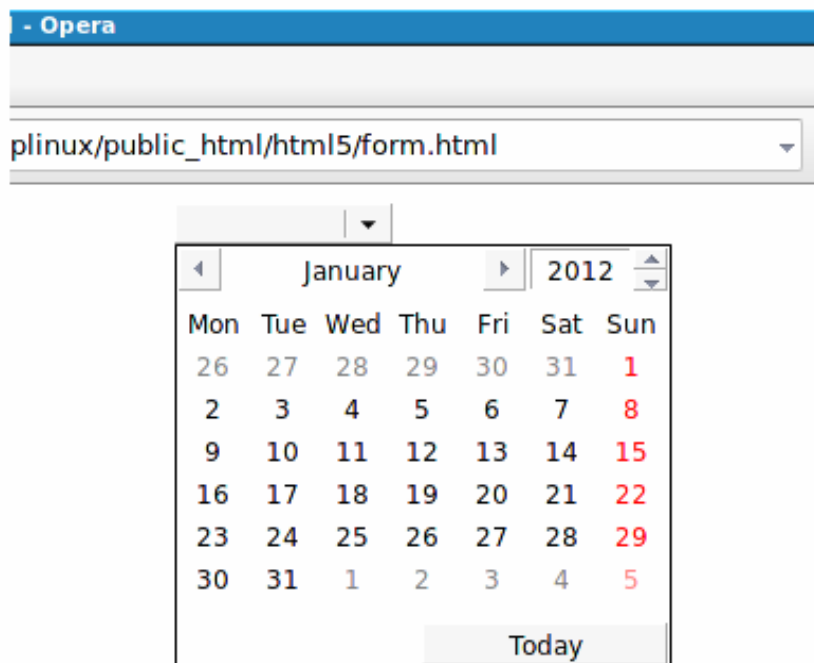
```
<input type=email>  
<input type=url>  
<input type=number min=18 max=60 step=2 value=19>  
<input type=range min=18 max=60 step=2 value=19>  
<input type=date>  
<input type=datetime>  
<input type=month>  
<input type=week>  
<input type=time>  
<input type=search>  
<input type=color>
```

Uses of most of these tags are clear from their name. At this moment the maximum support of these input types is available in the Opera browser. Screen shots of some of these tags are as follows:

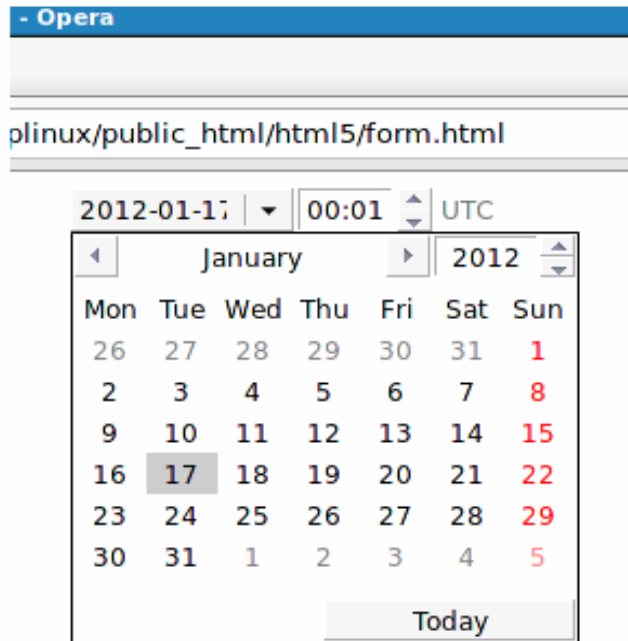
I. Screen shot of number, range, time and color input:



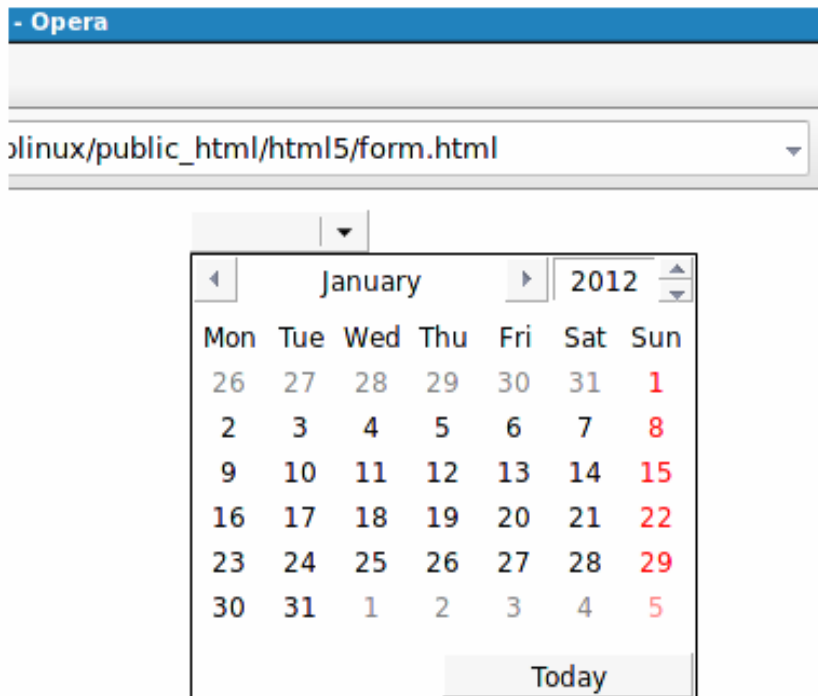
II. Screen shot of date input:



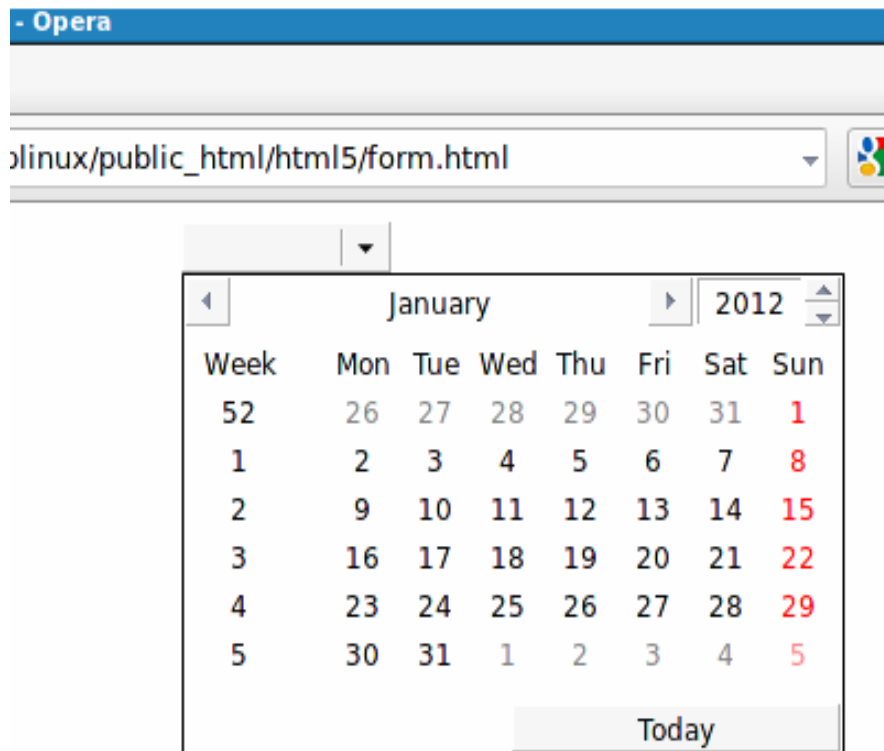
III. Screen shot of datetime input:



IV. Screen shot of month input:



V. Screen shot of week input:



D. Offline Browsing: HTML5 provides a mechanism using which a web page can be viewed in the offline mode also. When for the first time that web page is displayed in the presence of Internet connectivity then all necessary documents are being downloaded on the client computer. The documents to be downloaded for offline browsing are specified in a file known as manifest file. The manifest file is a file which contains a public declaration about the list of files to be available for offline browsing. The name of this file is specified with the help of manifest attribute of html tag as follows:

```
<html manifest="clock.manifest">
```

Then the content of the file clock.manifest can be written as follows:

```
CACHE MANIFEST
clock.html
clock.css
clock.js
```

Elements having dependency on JavaScript: Several tags in HTML5 have been introduced whose performance depends upon the JavaScript as described below:

A. canvas: This tag is used to draw an image on the web page using the logic represented in the JavaScript language. So drawing of image has become more convenient task for programmers. Several functions have been defined to control the movement of cursor position along with the functions to control the background and foreground colors to be used in the canvas. In addition to this, functions have also been provided to

control the color gradient on the canvas. Even the existing images can also be integrated in the canvas area. Output of some of the canvas can be viewed at the URL http://www.glug4muz.org/html5/test_canvas.html.

B. geolocation: A new object has been introduced in HTML5 to know the current geographical location of browser. The name of this object is geolocation which is available in the navigator global object. In order to know the current location of browser we need to call the `getCurrentPosition` function of this object as follows:

```
navigator.geolocation.getCurrentPosition(show_position, show_error);
```

where `show_position` is the address of the function to be called in case of success and `show_error` is the function to be called in error. Then inside the `show_position` function we can pass an argument to know the current position as follows:

```
function show_position(position)
{
  //position.coords.latitude represents current latitude
  //position.coords.longitude represents current longitude.
}
```

C. localStorage: Cookies have been traditionally used to store data on the client system. But the amount of data to be stored using them is very low. In addition to this Cookies are included in every HTTP request which makes the performance of application slow. So in order to remove these difficulties, in HTML5 a new object namely localStorage has been introduced. By using the `setItem` and `getItem` functions of this object we can save and retrieve values on client computer.

D. Drag and Drop: HTML5 provides a facility to define an element draggable with mouse pointer. An attribute has been defined with name `draggable` which can be used with any element. If the value of this attribute is true then the corresponding element can be dragged with mouse pointer. Corresponding to this attribute, some events have also been defined such as `ondragstart`, `ondrop` and `ondragover`. In addition to this an object with name `dataTransfer` has also been defined in the event object which can be used to set/get the nature of data to be dragged with the mouse pointer. The methods available with this object are `setData` and `getData`. An example of this functionality can be viewed at the URL http://www.glug4muz.org/html5/drag_text.html. This feature is not available in the Opera browser. But it is working fine in Google Chrome.

PROPOSED WORK

This paper presents a comprehensive analysis of new features introduced in HTML5. The newly introduced elements have been analyzed from the point of view of their implementation in various browsers available in the market as well as from the point of view of their role in the output and semantic of a web page. In addition to this the newly introduced elements have also been analyzed from the point of view of their dependency on the JavaScript language.

CONCLUSION

HTML5 is a step to implement the concept of semantic web. Attempts have been taken to make the HTML source code more meaningful. In addition to this several new tags as well as attributes have been introduced to perform the trivial work that was done with the help of Javascript code. But all features introduced in the HTML5 are not JavaScript independent till now. Several newly introduced tags are still dependent on JavaScript. Similarly since all browsers are still not compatible with HTML5 and so at this moment it is still mandatory to check the support of HTML5 in the browser before its actual implementation in the program. The support of a HTML5 tag in a browser can be easily done with help of an open source JavaScript library namely Modernizr which is available at <http://www.modernizr.com/>. In future all elements of HTML5 should be made JavaScript independent.

REFERENCES

- Mark Pilgrim, (2010). “HTML5 Up and Running”
- Brian P. Hogan, (2011). “HTML5 and CSS3: Develop with Tomorrow's Standard Today”
- Ivan Bayross, (2011). “HTML5 and CSS3 Made Simple”
- Rick Rogers, (2011). “Augmented Reality with HTML5”, Linux Journal
- Gobe Donnini, (2012). “Is HTML5 a New Battleground for Browsers”, Search Engine Journal