

Document Object Model and Open source software

Objectives : At the end of this lesson you shall be able to

- **define DOM**
- **explain DOM methods**
- **explain DOM documents**
- **describe HTML DOM elements**
- **explain HTML DOM events**
- **know Open Source Software**

JavaScript HTML DOM

The HTML DOM model is constructed as a tree of Objects:

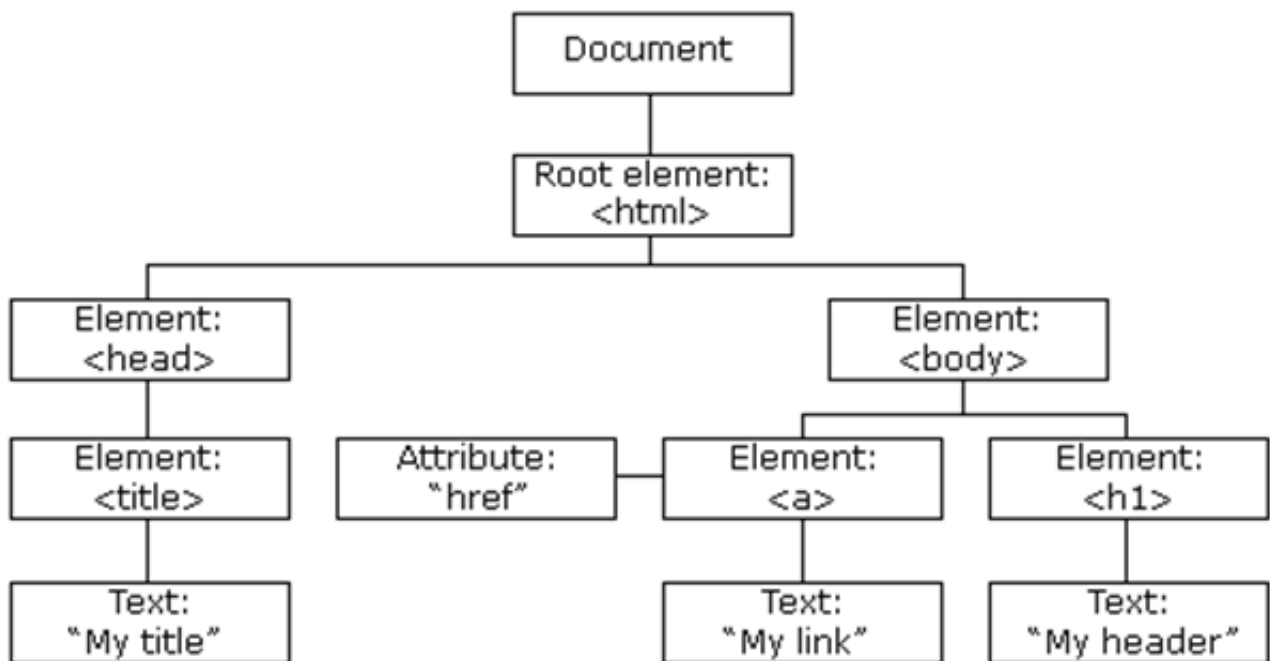
With the HTML DOM, JavaScript can access and modify all the elements of an HTML document.

The HTML DOM Tree of Objects

The HTML DOM (Document Object Model)

With the object model, JavaScript gets all the power it needs to create dynamic HTML:

When a web page is loaded, the browser creates a Document Object Model of the page.



- JavaScript can change all the HTML elements in the page.
- JavaScript can change all the HTML attributes in the page
- JavaScript can change all the CSS styles in the page
- JavaScript can remove existing HTML elements and attributes
- JavaScript can add new HTML elements and attributes
- JavaScript can react to all existing HTML events in the page

DOM

The DOM is a W3C (World Wide Web Consortium) standard.

The DOM defines a standard for accessing documents:

"The W3C Document Object Model (DOM) is a platform and language-neutral interface that allows programs and scripts to dynamically access and update the content, structure, and style of a document."

The W3C DOM standard is separated into 3 different parts:

- Core DOM - standard model for all document types
- XML DOM - standard model for XML documents
- HTML DOM - standard model for HTML documents

What is the HTML DOM?

The HTML DOM is a standard object model and programming interface for HTML. It defines:

- The HTML elements as objects
- The properties of all HTML elements
- The methods to access all HTML elements
- The events for all HTML elements

In other words: The HTML DOM is a standard for how to get, change, add, or delete HTML elements.

HTML DOM Methods

HTML DOM methods are **actions** you can perform (on HTML Elements)

HTML DOM properties are **values** (of HTML Elements) that you can set or change.

The DOM Programming Interface

The HTML DOM can be accessed with JavaScript (and with other programming languages).

In the DOM, all HTML elements are defined as **objects**. The **programming interface** is the properties and methods of each object. A **property** is a value that you can get or set (like changing the content of an HTML element). A **method** is an action you can do (like add or deleting an HTML element).

Example 1

The following example changes the content (the innerHTML) of the <p> element with id="demo":

Finding HTML Elements

Method	Description
document.getElementById()	Find an element by element id
document.getElementsByTagName()	Find elements by tag name
document.getElementsByClassName()	Find elements by class name

```
<html>
<body>
<p id="demo"></p>
<script>
document.getElementById("demo").innerHTML =
"Welcome to JavaScript!";
</script>
</body>
</html>
```

In the example above, getElementById is a method, while innerHTML is a property.

The getElementById Method

The most common way to access an HTML element is to use the id of the element. In the example above the getElementById method used id="demo" to find the element.

The innerHTML Property

The easiest way to get the content of an element is by using the innerHTML property. The innerHTML property is useful for getting or replacing the content of HTML elements.

Note: The innerHTML property can be used to get or change any HTML element, including <html> and <body>.

HTML DOM Document

HTML DOM document object

The document object is the owner of all other objects in your web page. In the HTML DOM object model, the document object represents your web page. If you want to access objects in an HTML page, you always start with accessing the document object.

Below are some examples of how you can use the document object to access and manipulate HTML.

Changing HTML Elements

Method	Description
<code>element.innerHTML=</code>	Change the inner HTML of an element
<code>element.attribute=</code>	Change the attribute of an HTML element
<code>element.setAttribute(attribute,value)</code>	Change the attribute of an HTML element
<code>element.style.property=</code>	Change the style of an HTML element

Adding and Deleting HTML Elements

Method	Description
<code>document.createElement()</code>	Create an HTML element
<code>document.removeChild()</code>	Remove an HTML element
<code>document.appendChild()</code>	Add an HTML element
<code>document.replaceChild()</code>	Replace an HTML element
<code>document.write(text)</code>	Write into the HTML output stream

Adding Events handlers

Method	Description
<code>document.getElementById(id).onclick=function(){code}</code>	Adding event handler code to an onclick event

JavaScript HTML DOM Elements

Finding HTML Elements

Often, with JavaScript, you want to manipulate HTML elements.

To do so, you have to find the elements first. There are a couple of ways to do this:

- Finding HTML elements by id
- Finding HTML elements by tag name
- Finding HTML elements by class name
- Finding HTML elements by HTML object collections

Finding HTML Elements by Id

The easiest way to find HTML elements in the DOM, is by using the element id.

This example finds the element with id="demo":

Example 2

```
var x = document.getElementById("demo");
```

If the element is found, the method will return the element as an object (in x).

If the element is not found, x will contain null.

Finding HTML Elements by Tag Name

This example finds the element with id="main", and then finds all <p> elements inside "main":

Example 3

```
var x = document.getElementById("main");
var y = x.getElementsByTagName("p");
```

Finding HTML Elements by Class Name

If you want to find all HTML elements with the same class name, use this method `getElementsByClassName()`

Example 4

```
document.getElementsByClassName("intro");
```

The example above returns a list of all elements with class="intro".

Note: Finding elements by class name does not work in Internet Explorer 5,6,7, and 8.

Finding HTML Elements by HTML Object Collections

This example finds the form element with id="frm1", in the forms collection, and displays all element values:

Method	Description	DOM
document.anchors	Returns all <a> with a value in the name attribute	1
document.applets	Returns all <applet> elements (Deprecated in HTML5)	1
document.baseURI	Returns the absolute base URI of the document	3
document.body	Returns the <body> element	1
document.cookie	Returns the document's cookie	1
document.doctype	Returns the document's doctype	3
document.documentElement	Returns the <html> element	3
document.documentMode	Returns the mode used by the browser	3
document.documentURI	Returns the URI of the document	3
document.domain	Returns the domain name of the document server	1
document.domConfig	Returns the DOM configuration	3
document.embeds	Returns all <embed> elements	3
document.forms	Returns all <form> elements	1
document.head	Returns the <head> element	3
document.images	Returns all elements	1
document.implementation	Returns the DOM implementation	3
document.inputEncoding	Returns the document's encoding (character set)	3
document.lastModified	Returns the date and time the document was updated	3
document.links	Returns all <area> and <a> elements value in href	1
document.readyState	Returns the (loading) status of the document	3
document.referrer	Returns the URI of the referrer (the linking document)	1
document.scripts	Returns all <script> elements	3
document.strictErrorChecking	Returns if error checking is enforced	3
document.title	Returns the <title> element	1
document.URL	Returns the complete URL of the document	1

Example 5

```
var x = document.getElementById("frm1");
var text = "";
var i;
for (i = 0; i < x.length; i++) {
    text += x.elements[i].value + "<br>";
}
```

document.getElementById("demo").innerHTML = text;

The following HTML objects (and object collections) are also accessible:

- document.anchors
- document.body
- document.documentElement
- document.embeds
- document.forms
- document.head
- document.images
- document.links
- document.scripts
- document.title

The HTML DOM allows JavaScript to change the content of HTML elements.

Changing the HTML Output Stream

JavaScript can create dynamic HTML content. In JavaScript, `document.write()` can be used to write directly to the HTML output stream.

Example 6

```
<!DOCTYPE html>
<html>
<body>
<script>
document.write(Date());
</script>
</body>
</html>
```

Note: Never use `document.write()` after the document is loaded. It will overwrite the document.

Changing HTML Content

The easiest way to modify the content of an HTML element is by using the `innerHTML` property. To change the content of an HTML element, use this syntax.

```
document.getElementById(id).innerHTML = new HTML
```

This example changes the content of a `<p>` element:

Example 7

```
<html>
<body>
<p id="p1">Hello World!</p>
<script>
document.getElementById("p1").innerHTML = "New text!";
</script>
</body>
</html>
```

Example explained:

- The HTML document above contains a `<p>` element with `id="p1"`
- We use the HTML DOM to get the element with `id="p1"`
- A JavaScript changes the content (`innerHTML`) of that element to "New text!"

This example changes the content of an `<h1>` element:

Example 8

```
<!DOCTYPE html>
<html>
<body>
<h1 id="id1">Old Heading</h1>
<script>
var element = document.getElementById("id1");
element.innerHTML = "New Heading";
</script>
</body>
</html>
```

Example explained:

- The HTML document above contains an `<h1>` element with `id="id1"`
- We use the HTML DOM to get the element with `id="id1"`
- A JavaScript changes the content (`innerHTML`) of that element to "New Heading"

Changing the Value of an Attribute

To change the value of an HTML attribute, use this syntax.

```
document.getElementById(id).attribute = new value
```

This example changes the value of the `src` attribute of an `` element.

Example 9

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

<script>
document.getElementById("Image1").src =
"newflower.jpg";
</script>
</body>
</html>
```

Example explained:

- The HTML document above contains an `` element with `id="myImage"`
- We use the HTML DOM to get the element with `id="myImage"`
- A JavaScript changes the `src` attribute of that element from "smiley.gif" to "landscape.jpg"

Changing HTML Style

To change the style of an HTML element, use this syntax.

```
document.getElementById(id).style.property = new style
```

The following example changes the style of a <p> element:

Example 10

```
<html>
<body>
<p id="p2">Hello World!</p>
<script>
document.getElementById("p2").style.color = "green";
</script>
<p>The paragraph above was changed by a script.</p>
</body>
</html>
```

Using Events

The HTML DOM allows you to execute code when an event occurs. Events are generated by the browser when "things happen" to HTML elements.

- An element is clicked on
- The page has loaded
- Input fields are changed

This example changes the style of the HTML element with id="id1", when the user clicks a button.

Example 11

```
<!DOCTYPE html>
<html>
<body>
<h1 id="id1">Heading1</h1>
<button type="button"
onclick="document.getElementById('id1').style.color =
'blue'">
Click Me</button>
</body>
</html>
```

JavaScript HTML DOM Events

HTML DOM allows JavaScript to react to HTML events.

Reacting to Events

A JavaScript can be executed when an event occurs, like when a user clicks on an HTML element. To execute code when a user clicks on an element, add JavaScript code to an HTML event attribute

```
onclick=JavaScript
```

Examples of HTML events:

- When a user clicks the mouse
- When a web page has loaded
- When an image has been loaded
- When the mouse moves over an element
- When an input field is changed
- When an HTML form is submitted
- When a user strokes a key

In this example, the content of the <h1> element is changed when a user clicks on it.

Example 12

```
<!DOCTYPE html>
<html>
<body>
<h1 onclick="this.innerHTML = 'Oops!'">Click on this
text!</h1>
</body>
</html>
```

HTML Event Attributes

To assign events to HTML elements you can use event attributes.

Example 13

Assign an onclick event to a button element:

```
<button onclick="displayDate()">Try it</button>
```

In the example above, a function named displayDate will be executed when the button is clicked.

Assign Events Using the HTML DOM

The HTML DOM allows you to assign events to HTML elements using JavaScript.

Example 14

Assign an onclick event to a button element:

```
<script>
document.getElementById("myBtn").onclick =
displayDate;
</script>
```

In the example above, a function named displayDate is assigned to an HTML element with the id="myBtn".

The function will be executed when the button is clicked.

The onload and onunload Events

The onload and onunload events are triggered when the user enters or leaves the page. The onload event can be used to check the visitor's browser type and browser version and load the proper version of the web page based on the information. The onload and onunload events can be used to deal with cookies.

Example 15

```
<body onload="checkCookies()">
```

The onchange Event

The onchange event is often used in combination with validation of input fields. Below is an example of how to use the onchange. The upperCase() function will be called when a user changes the content of an input field.

Example 16

```
<input type="text" id="fname" onchange="upperCase()">
```

The onmouseover and onmouseout Events

The onmouseover and onmouseout events can be used to trigger a function when the user mouses over or out of, an HTML element.

The onmousedown, onmouseup and onclick Events

The onmousedown, onmouseup and onclick events are all parts of a mouse-click. First when a mouse-button is clicked, the onmousedown event is triggered, then, when the mouse-button is released, the onmouseup event is triggered, finally, when the mouse-click is completed, the onclick event is triggered.

onmousedown and onmouseup

Change an image when a user holds down the mouse button.

onload

Display an alert box when the page has finished loading.

onfocus

Change the background-color of an input field when it gets focus.

Mouse Events

Change the color of an element when the cursor moves over it.

DOM Event Listener

The addEventListener() method

Add an event listener that fires when a user clicks a button.

Example 17

```
<!DOCTYPE html>
<html>
<body>
<h2>JavaScript addEventListener()</h2>
<p>This example uses the addEventListener() method to attach a click event to a button.</p>
<button id="myBtn">Try it</button>
<p id="demo"></p>
<script>
document.getElementById("myBtn").addEventListener("click", displayDate);
function displayDate() {
document.getElementById("demo").innerHTML = Date();
}
</script>
</body>
</html>
```

Result: (Fig 1)



- The addEventListener() method attaches an event handler to the specified element.
- The addEventListener() method attaches an event handler to an element without overwriting existing event handlers.
- You can add many event handlers to one element.
- You can add many event handlers of the same type to one element, i.e two "click" events.

- You can add event listeners to any DOM object not only HTML elements. i.e the window object.
- The `addEventListener()` method makes it easier to control how the event reacts to bubbling.
- When using the `addEventListener()` method, the JavaScript is separated from the HTML markup, for better readability and allows you to add event listeners even when you do not control the HTML markup.
- You can easily remove an event listener by using the `removeEventListener()` method.

Open Source Software

You can build a website using these popular free and open source website building tools. Nowadays, whether you are an individual entrepreneur or representing a business organisation, a website is a must for personal and professional growth. Organisations are spending lots of money to build attractive websites. The following are some of the open source website building tools that you can use to build your website on your own, without much knowledge about programming or the Internet.

1 WordPress

The official websites for WordPress are <https://wordpress.com> and <https://wordpress.org/>.

2 Kompozer

The official website for Kompozer is <https://www.kompozer.net>.

3 Joomla

The official website for Joomla is <https://www.joomla.org/>.

4 Drupal

The official website for Drupal is <https://www.drupal.org/>.

5 OpenCms

The official website for OpenCms is [http:// www.opencms.org/en/](http://www.opencms.org/en/).

Develop and edit web pages in KompoZer

Objectives : At the end of this lesson you shall be able to

- know introduction to KompoZer
- know develop and edit web pages
- know the use of templates
- explain Publishing web sites
- describe Site Manager
- know preferences
- explain publishing web sites.

Introduction

What is KompoZer?

KompoZer is a complete Web Authoring System which integrates web page development and web file management. It provides a web page editor which has a simple graphical (WYSIWYG - what you see is what you get) interface. With KompoZer, newcomers will quickly and easily be able to produce new web pages. The output code is compliant to a high extent with the latest issues of the appropriate web language specifications

KompoZer incorporates a Site Manager; this gives rapid access to the files on both local machines and remote servers. It can cater for several sites and switch rapidly between them. From within KompoZer pages and associated files may be uploaded to a remote server. KompoZer supports the use of "Styles" through Cascading Style sheets (CSS) both embedded and external. It has an editor which generates CSS code conforming with CSS 2.1 specifications.

Who is KompoZer for?

KompoZer is suitable for anyone wishing to have a modern, free of charge, program for developing small web sites and who would like to learn modern web design techniques such as the use of CSS.

Basics :

Open KompoZer. The main window opens. At the top are a number of toolbars. The topmost is the Menu Bar. This carries a number of items (File, Edit etc) used to make selections. The next is the 'Composition Toolbar' which carries a number of 'Buttons' labelled 'New', 'Open' etc.

To create a new page: On the Composition toolbar Click the 'New' button.

To open an existing page: Assuming that the page is stored on your local disk in HTML format: On the menu Bar click 'File' then 'Open File'. Browse to the file and click 'Open'.

Editing a web page: Your web page - blank or otherwise - is in the large pane in the centre right of the KompoZer

application window. Many editing functions are very similar to those in a word processor. The top four toolbars on the KompoZer application window provide a number of editing functions - to see what any do hover the cursor over an item and a hint will appear.

Saving a Page To save a page: On the Composition toolbar click 'Save'. If it was a new document a dialog window will ask you to enter a title for the page. This will appear in the tab at the top of the page display area. NB this is NOT the file name. Click 'OK'; you will then be offered a normal save window which allows you to browse to a suitable location and name the file. The file extension offered will be HTML.

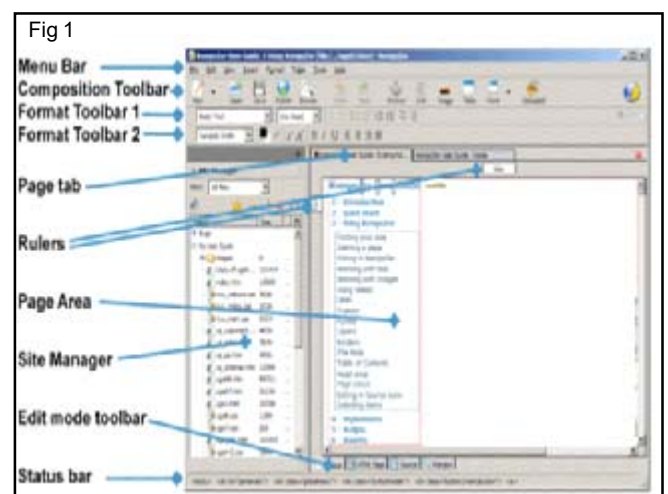
Browsing a page :To see how your page will look in your default browser on the Composition toolbar click 'Browse'.

Help: The help system should be a first resort in case of need. The forum at <http://wysifauthoring.informe.com/> is a place for sharing experiences and obtaining and giving help. Many of the contributors are users of the programs and range from beginners to those with lots of experience.

Develop and edit Web pages

KompoZer Screen Layout

KompoZer screen layout. (Fig 1)



When KompoZer starts the window carries a menu bar across the top (File, Edit, View etc). Below this are three 'Toolbars'. To ensure that everything is visible, on the Menu bar select **View > Show/Hide** and see that each of the following is checked:

- Composition Toolbar,
- Format Toolbar 1
- Format Toolbar 2
- Edit Mode Toolbar
- Status bar
- Rulers
- Site Manager

The three toolbars across the top of the window carry buttons (represented by icons). Hover on any to find its function. If any are greyed out they are not functional in the current mode as they are context sensitive. Across the centre of the window are two panes: the 'Site Manager' on the left and a blank web page on the right. The Site Manager is a powerful tool. Since it is not needed yet it may be closed by clicking on its close button or pressing F9. At the top of the Page area there is a Page tab which carries the name of the page ('Untitled'). KompoZer Help refers to this as the 'Tab Browser' toolbar. If you had several pages open, as shown in the Fig. 1, this tab would allow you to select one them rapidly. At the bottom of the page area is the 'Edit Mode Toolbar' which carries four tabs which select one of four 'Viewing modes' for a page ('Normal', 'HTML Tags', 'Source', 'Preview'.)

At the bottom of the window is the 'Status Bar'. This is a very powerful tool. Once a page is populated, by clicking any item in the page area its structure appears on the status bar. Any class or id applied to an element is shown and any bearing an inline style is indicated in italic type. Hovering reveals the style declaration. Additionally clicking an element marker highlights the element in both normal view and, on changing view, in source view thus simplifying navigation in source view. Note The figure shows the buttons as they appear when KompoZer is first installed. They may be customised to different arrangements. If this has been done some of the following may be difficult to follow.

To restore defaults click **View > Show/Hide > Customize Toolbar > Main Toolbar > Restore Default Set** and repeat similarly for the Format Toolbar. Using KompoZer right-click any toolbar to customise it.

Options for starting a page

There are several ways to start new pages or open existing ones.

To start a new blank page, on the menu bar click File > New. A window headed 'Create a new document or template' appears. Check the boxes 'A blank document' and 'Strict DTD' and clear 'create a XHTML document'.

To open an existing page

- 1 Click the OPEN button to access a normal browse dialogue.
- 2 Click File > Recent Pages to get rapid access to those recently worked on.
- 3 Or use the Site Manager which provides a powerful mini-browser and is very easily set up. The Doctype of an existing page will remain the same as before it was opened. It cannot be changed in KompoZer.

Each page opened starts in a new tab which can be clicked to select a document to work on.

Editing in KompoZer

KompoZer supports all the standard Windows editing commands and shortcut keys. e.g. Copy Ctrl+C, Paste Ctrl+V etc. There are other KompoZer specifics. These are great time savers. In 'HTML Tags' view KompoZer supports drag and drop editing for block items. (Select an item by pressing the Control key while clicking on the Tag.) An extreme time saver is KompoZer's double click response. In several cases, such as links, images and tables, a very useful editing window is opened. KompoZer supports many levels of Undo and Redo, however changes made in 'Source' view cannot be undone after you have changed the view.

Saving files

Go to File > Save as. You are offered a 'Save Page as' window which allows you to browse to the folder you want to use. You will find the file name already completed with your page title. You will probably want to change this to a shorter, all lower case, name. You will find the file extension completed as 'html' you may prefer to, and may alter it to 'htm'.

Printing pages

The 'Print' button allows you to print the current page to a printer. This prints the page view and not the source code.

Working with text

Text typed directly onto the KompoZer page defaults to appearing in the format for the 'Body' element.

HTML defines a small number of elements specifically for text and it is usually preferable to use these. To format text in a standard element format select the text and click the first drop-down box on the format toolbar. This offers a selection of standard text formats. Paragraph is the most appropriate for general text.

Once formatted as a paragraph, when typing in a text area, use of the 'enter' key starts a new block of text i.e. a paragraph. To start a new line within the current paragraph press Shift+Enter; this generates a line break.

Other standard text formats are Heading formats from Heading 1, the largest (for the main heading), to Heading 6, the smallest (for the least significant). Browsers generally render headings in bold type. Text can be edited in any of the viewing modes and KompoZer responds to all the normal windows shortcut commands.

Formatting text

Text can be formatted in a number of ways using a format toolbar. The changes listed in the table can be applied (hover over a tool to discover what it does).

Choose a font , Choose text colour, Choose background colour, Choose highlight colour, Make text smaller, Make text larger, Embolden, Italicise, Underline, Format as a numbered list, Format as a bulleted list, Align left, Align Centre, Align right, Justify, Indent text, Outdent text, Emphasise, Strongly emphasise.

Numbered and Bulleted lists

KompoZer can format a list of items giving each item a sequential number in any of several formats or presenting them bulleted. To start a list from scratch

- 1 Click one of the list buttons (Numbered List or Bulleted list) on the Format toolbar.
- 2 Type the first item.
- 3 Press Enter and type the next item.
- 4 To finish, on the last(blank) item press Enter.

To change existing text into a list

- 1 Select the text required.
- 2 Click one of the list buttons on the Format Toolbar. The text will be changed into a list, a new item starting for each paragraph or other block item encountered.

To add items to a list

- 1 Click at the end of the last item in the list.
- 2 Press Enter and type the new item. Numbering and format will continue from the previous item.

Importing text

Strictly KompoZer does not support importing text from other applications but it is possible to copy and paste text. In normal view content from other web pages may be copied reliably. The result will be rendered according to any styling applied in your document; any reliance on external styles in the original document will be lost.

Text from word processors such as Microsoft Word or OpenOffice.org in rtf or doc format or from text editors such as Windows notepad may also be copied and pasted. When such text is pasted into KompoZer most formatting is lost. Numbered lists will be retained.

Tabs will be rendered as three non-breaking spaces. The contents of tables may be pasted, individual cells will be separated as if by tabs.

Special characters

By special characters we refer to characters which do not have an equivalent keyboard key. HTML uses a system of characters known as 'Unicode'. This includes a large range of characters including all the international currency symbols and many thousands of others, though the fonts supplied on computers will support only a subset. A number of the more common, including accented ones, may be inserted using Insert > Characters and Symbols.

Checking spelling

In any view, other than Source, click on the 'Spell' button.



The spell checker will work sequentially through the page.

To insert an image

- 1 Click the 'Image' button on the Composition toolbar
- 2 The Image properties window opens. Click 'Choose File' and browse and select a file
- 3 Click 'Open'. Leave checked the box 'URL is relative to page location' this will allow you to move the page and image to another location, as you will have to when you upload them to a server. (If you de-select this and move the page, it will try to find the image at the original location.) Note If the box is 'greyed out' this is probably because the page has not been saved.
- 4 In the box labelled 'Alternate text' add a description of the image. (This forms the 'alt' attribute for the image and provides text which will appear in place of the image with user agents (browsers) that cannot display images (screen readers and voice synthesisers). It will also be used by those with visual impairment. The content of this box must be carefully considered so as to be of maximum assistance in such cases.) Note Where the image is purely decorative, and not necessary to understanding the page, alternative text is not required and should be omitted
- 5 In the box labelled 'Tooltip' you may optionally insert a 'Title' attribute for the image. Some browsers will show the text provided when the cursor hovers over the image
- 6 Click OK

Using tables

Tables allow data - images, links, forms, form fields, text, etc. - to be arranged into rows and columns of cells. Inserting tables

- 1 On the Composition Toolbar click the Table button. The 'Insert table' window appears

- 2 Leave the 'Quickly' tab selected and drag out a matrix then click the bottom right cell to define the table arrangement
- 3 The cells appear on the screen with narrow outlines

Note: If later the table border is set to zero these outlines disappear but KompoZer in normal view replaces them with a red outline. This does not appear in Preview or in a browser. Tables have resizing boxes similar to those used with images

Table cell properties

Right-click the table and select 'Table Cell properties'. The Table properties window opens. This has two tabs 'Table' and 'Cells' which allow overall control of several aspects of either the table or individual cells. This includes

- a Alignment of text within cells
- b Wrapping of text
- c Cell spacing - the gap between cells
- d Cell padding - the gap between the edge of the cell and the text within it
- e Size of table and cells
- f Background colour
- g Selection of cells as 'Normal' or 'Header' (Cells which are headings to rows or columns should be selected as 'Header'. Normally this results in them being rendered in bold type.)

Linking text

Linking to another file

To create a link

- 1 Select (highlight) a few words of text
- 2 On the Composition toolbar click on the 'Link' button, alternatively Right-click and select 'Create Link'. The 'Link Properties' window opens
- 3 Click on 'Choose File' and browse to the file that you want to link to
- 4 Click OPEN
- 5 Click OK

Inserting an email address


Instead of linking to a file it is possible to insert an email address. The result will be that, in use, when the link is clicked the email client on the visitor's machine will be opened with the correct address selected.

To do so proceed as under the previous heading. When the Link Properties window opens (or if Image Properties click the Link tab) enter the email address and check the box 'The above is an email address'.

Inserting named Anchors

There is a second type of Anchor element the 'Named anchor'. Such an anchor is extremely useful as it can act as a type of bookmark defining a particular place on a page. Links can jump to such bookmarks.

To insert a named anchor

- 1 Place the cursor at the point you want to mark.
- 2 Click the 'Anchor' button on the Composition toolbar or, on the Menu Bar, select Insert > Named Anchor. The named anchor properties window appears.
- 3 Enter a unique name for the anchor.
- 4 Click OK. In 'Normal' view anchors are marked by a picture of an anchor .

Linking to named anchors

Start as above for linking to another file. When the 'Link Properties' window opens, instead of choosing file use the drop down list. Your anchor name should appear there preceded by a "#". Click it and OK. That's it! If you test your page on a browser when you click the link the view should move to show the position of the anchor.

Linking images

The techniques and possibilities are very similar to those used with text.

To create a link

- 1 Click on the image
- 2 On the Composition toolbar click on the 'Link' button, alternatively Right-click and select 'Create Link'. The 'Image Properties' window opens
- 3 Click on 'Choose File' and browse to the file that you want to link to. (The box 'URL is relative to page location' is checked. This means that if you move your page to a new folder you should move the image to a corresponding new folder. If you clear this box the absolute address of the image on the hard drive is given in full. If you move your page now it will look there for the image. As you start to organise a web site you will find that this is not a good arrangement and potentially disastrous when you upload the page to a server.)
- 4 Click OPEN
- 5 Click OK

Editing Links

To change the file to which a link refers, in Normal, Tags or Preview mode double-click on the link. The 'Link properties' window opens (for an image the 'Image Properties' window opens - click the Link tab). Edit the link.

To remove the link delete the link reference in the box.

Frames

If you open a frame document, you get a message 'This page can't be edited for an unknown reason' but it displays the frame content rather beautifully. Then you can do nothing with it except click on the 'Source' tab. You then see the source code and the system will seem to lock up. Actually it doesn't lock and you can load another page and revert to normal operation.

This is not a great limitation. Though the code for frames takes a little getting used to it is usually quite short and can easily be produced using a text editor. Once established it probably rarely needs to be altered. You can use KompoZer to develop the pages that go into the frames.

Forms

To set up a form

- 1 Click the form button.
- 2 In the Form properties window give the form a name of your choosing
- 3 Complete the Action box with the correct URL and select the appropriate method
- 4 'Encoding' and 'Target Frame' will frequently not be required but, if they are, select 'More Properties' and complete the boxes
- 5 Click OK
- 6 On the form place any headings, paragraphs and images ensuring that there is a placeholder for any controls needed. (If blank placeholders are needed it is probably sensible to put some dummy text in now and delete it later.)
- 7 Where controls are needed click the corresponding placeholder and using the drop down box beside the Form button select the required control
- 8 Give each control a unique name
- 9 Each control has specific information which needs to be entered. Enter it into the box in the window which appears

Table of contents

If you have a long document with sections headed using heading formats Heading 1, Heading 2 etc KompoZer can generate a Table of Contents (ToC) automatically. The table reflects the structure of the page, the content of the headings forming the text of the table.

Inserting a Table of Contents

- 1 Place the cursor where the table is required
- 2 Click Insert > Table of Contents > Insert
- 3 The 'Table of Contents' window appears
- 4 In the column headed 'Tag' select the tag for each level

e.g. against level 1 select h3 and against level 2 select h4 and for all the others select '--'

- 5 If, instead of using headings you wish to use classed paragraphs or a div, instead of selecting a heading tag select 'p' or 'div' and in the box in the column headed 'Class' enter the class required. (It is, of course, also possible to select headings by allocating a class.)
- 6 If you wish the contents to be numbered check the box 'Number all entries ...'
- 7 Click OK

The Table of Contents will be created.

To update a ToC

After changes have been made to a page. There is no need to place the cursor.

- 1 Click Insert > Table of Contents > Update
- 2 The 'Table of Contents' window appears showing the selections previously made
- 3 If desired, changes may be made to the selections
- 4 To update the ToC click OK

To delete a ToC

There is no need to place the cursor.

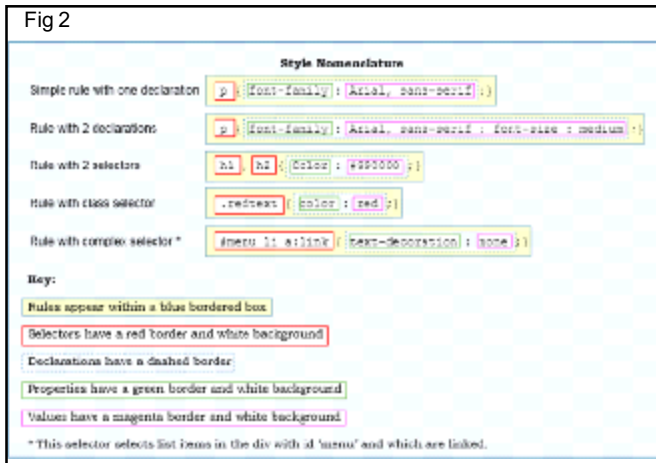
- 1 Click Insert > Table of Contents > Remove

Style and stylesheets

Styles specify how particular elements on a page appear on the screen, in print or whatever. Style may typically define aspects of presentation as the font face, size and variant, the font colour, the background colour, whether an element is to be aligned right, centre or left, whether spaced away from other elements, surrounded by a border and, if so, what type or colour. Elements may be given an absolute position relative to the page (which allows elements to overlap).

Elements such as paragraphs, tables and images are considered to exist within boxes or blocks and the sizes of these boxes may be specified.

Classes As well as allowing you to specify the style of elements it is possible to define styles and apply them selectively to only some elements. This is done through 'Classes' - a 'Class' is just a style that can be applied as and when you choose. 'Classes' are applied to 'Tags' (a marker attached to an element). The element to which the class is applied appears in the format defined by the class. Other similar elements without the class applied appear in the default format i.e. either the default specified by the browser or the style that the user has defined for the corresponding element. On the status bar KompoZer shows classes along with the tag to which they are applied. Fig 2 shows Style Nomenclature.

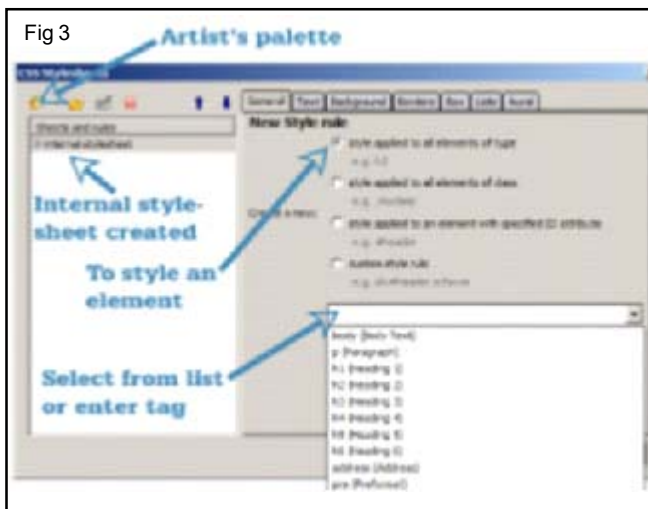


Creating styles

Internal styles: AsKompoZer includes a CSS editor called CaScadeS which allows you to create stylesheets and define rules. Before you define a rule you must have a stylesheet but if you are working on a page which has none CaScadeS will create one for you.

Creating a style rule for an element.

Css Stylesheet window (Fig 3)



To create a style rule for an element

- 1 Click the CaScadeS button on the Composition toolbar. The CSS Stylesheets window opens.
- 2 Click on the artist's palette button. In the 'Sheets and rules' pane you will see an internal stylesheet has been created for you.
- 3 To create a rule click 'Style applied to all elements of type'
- 4 Beside the blank box click the drop down arrow. You will see listed a number of common elements. To create a style for one of these click it alternatively enter the tag for any other element.
- 5 Click 'Create Style rule'

- 6 You are now presented with a window headed 'Selector' followed by the tag for the element. The window actually lists the style declarations for that element, but of course that is now blank.

To define how you want elements to look

- 1 Select in turn as required the tabs for 'Text', 'Background', 'Border' etc and specify exactly how you wish that element to appear. The next section amplifies some details of how to do this.
- 2 Return to the general tab to see the full declarations that you have set for the Selector.
- 3 If you click the 'General' tab you will see all the declarations for the rule. You can edit these here but it is better to leave the job to CaScadeS because if you make any errors the declaration will be deleted
- 4 When you are satisfied click 'OK'.

Link to an existing stylesheet

If you have a stylesheet that you created for another page or intend to use right across your site you can link your page to that.

To link to an existing external style sheet

- 1 Click the CaScadeS button on the Composition toolbar. The CSS Stylesheets window opens
- 2 Click the dropdown arrow beside the artist's palette button and select 'Linked stylesheet'
- 3 Click 'Choose file' and browse to the file that you need
- 4 Click 'Open'
- 5 Click 'Create stylesheet'
- 6 Click 'OK' You can now close CaScadeS, but, of course you can work on the stylesheet in the usual way.

Saving stylesheets

Once you edit a stylesheet in KompoZer its name is marked by a red symbol indicating that it has not been saved. When you close CaScadeS changed sheets are saved immediately.

Removing styles

CaScadeS allows you to remove styles in a similar way to adding them. In the 'Sheets and rules' pane select the rule you want to remove and click 'Remove'. Similarly you can remove a stylesheet. Select the sheet and click 'Remove'. If you select an internal stylesheet it is deleted from the file completely.

Templates

Templates are basically pages having some content (e.g. a letter head) which can be re-used to create other pages which will have the same underlying page structure and, often, the same graphical layout.

Templates are not altered in use and can be used over and over again. The simplest template is probably a blank sheet which links to a stylesheet for use throughout a site. More common is a page which has a banner and perhaps a menu to appear on every page. Last might be a complete page layout for use on all, or many, pages of a site but which includes areas for customising individually. Templates may be considered as having two parts - the fixed part or 'boilerplate' which remains the same for every page and the 'editable part' which changes.

Create a template from a page

A pre-existing document may be transformed into a template

- 1 Click Format > Page Title and Properties
- 2 Check the box 'This page is a template'.
- 3 Click OK.
- 4 Click File > Save as. The file type 'HTML Template' will be completed.
- 5 Name and save the file as normal

To save a template

- 1 Click File > Save or File > Save As.
- 2 The extension 'mzt' will be selected automatically.

To make blocks editable

- 1 In turn, select each block that you wish to make editable.
- 2 In HTML tags view select the block by clicking its tag.
- 3 On the status bar right-click the corresponding highlighted tag.
- 4 Click Templates > Make editable.
- 5 In the 'Insert an editable area' window give the block a recognisable name. Now check the options boxes if required.
- 6 Click OK.

To make a flow selection editable

- 1 In turn select (highlight) each section of text that you wish to make editable.
- 2 Click Insert > Templates > Insert editable area.
- 3 In the 'Insert an editable area' window give the block a recognisable name.
- 4 Leave checked the option 'Flow of text'.
- 5 Check the options boxes if required as described above. Note The option 'Area is moveable' is inappropriate for flow areas).
- 6 Click OK

To create a page using Template

- 1 Click File > New > A new document based on a template > Choose File.
- 2 Select the Template (Note templates have the file extension 'mzt')
- 3 Click 'Create'.

To use the page

- 1 Click in turn in each editable areas.
- 2 Select and delete the sample text and replace it with new text.
 - 2a If the editable area was repeatable a small square appears within the label, hovering turns it red and clicking makes a copy. Copies have small circles which act as delete buttons.
 - 2b If any area was optional a small circle with an x appears within the label. Hovering turns it red and clicking deletes it. The same figure shows this for the 'Other languages'. Because of the limitation described it has not been possible to fill in all the editable areas.
- 3 When all editable boxes have been completed detach the page from the template by clicking Edit > Detach from template. The page now assumes its final appearance.
- 4 Save the page in the normal way. The last figure shows the final result. The areas which could not be edited earlier have been completed. Now it is possible to edit any item and as a workaround the frozen repeatable items may be added.

Editing templates

Templates which have already been saved may be altered after opening using menu commands File > Open File and selecting 'Files of Type' then 'HTML Templates'.

Site Manager

The site manager allows you to navigate your site or between sites easily. To toggle the Site Manager on or off either press F9 or use View > Show/Hide > Site Manager. Site Manager can deal with sites irrespective of whether they reside on a local machine or on a remote server. In the latter case, if you are on a dial-up network, Site Manager will dial and make the connection for you. Since generally you will set up a site on a local machine and later 'publish' to a remote server we will deal first with setting up on a local machine. Site Manager provides a directory tree view of a site similar to the view with Windows Explorer. It however lists only directories which you have specifically set up as 'Sites'. You can set up many sites, they appear in Site Manager irrespective of where they appear in a normal directory tree.

Setting Preferences

You can set up a number of features in KompoZer according to personal preferences. Several of the options are grouped under the Tools > Options menu. In addition you can customise toolbars via the View > Show/Hide menu.

Defaults

The defaults set by KompoZer will generally be found satisfactory. (Fig 4)

Fig 4

Menu selection	Default
Tools > Options > General	
Maximum number of pages	10
Retain original source formatting	Checked
Reformat HTML Source	Cleared
Save images when saving pages	Cleared
Always show publish dialog	Cleared
Maintain table layout	Checked
Use CSS styles	Checked
Always open a document in a new tab	Checked
Tools > Options > Fonts	
Allow documents to use other fonts for others see text	Checked
Tools > Options > New Page settings	
Author	Blank (see below)
Reader's default colors	Checked
Background image	Blank (see below)
Language	Blank (see below)
Writing direction	No direction specified
Character set	ISO-8859-1
Tools > Options > Advanced	
Set up Proxies	Direct connection
Markup - Language	HTML 4
Markup - DTD	Strict
Return in paragraph always creates new	Checked
Underline misspelled words	Checked
Output the following characters	HTML4
Special characters	Only & < > and no break space
Don't encode '>' outside attribute	Cleared
Don't encode special characters	Cleared

Editing preferences KompoZer can be customised in several ways through the Menu selection Tools > Options mechanism. All the options may be set at any time. All take effect immediately except for 'New page settings' which do not apply to any existing page.

Publishing

Introduction Publishing a site means transferring the site, i.e. the pages, images and stylesheets involved, to a web server from which they may be accessed, usually but not necessarily, by the public. This process is called 'Uploading'. Prior to publishing there are a few checks which should be carried out.

Setting up your site

While setting up Site Manager you may already have configured the 'remote' site, if not, either proceed as detailed there, go directly to Publish Settings via Edit > Publishing Site settings.

Confirmation of correct publication Enter the following details:

- In the 'Site names' box enter the name that you want to know the site by.

- HTTP address (URL) of your site. From your ISP (see hints).
- Publishing address - This is the ftp address to which you will publish.
- User name - From your ISP.
- Password - From your ISP.
- If you wish to, check 'Save Password'. If you have several sites set up and you have one site that you always or usually publish to you may wish to click on the name of this site then 'Set as default'. This simplifies uploading. Click OK.

Uploading

Open the page that you want to upload. An easy way to do this is from the Site Manager.

- 1 Click the 'Publish' button.
- 2 On the 'Publish Page' window on the 'Publish' tab, if it is not your default, in the 'Site name' box select the site to which you want to publish. The 'Page title' and 'File name' should already be completed.
- 3 If the page is to be uploaded to a sub-directory, rather than the root directory, enter the name of a sub-directory and any of the other data if required.

Note: This directory must exist. KompoZer cannot create it.

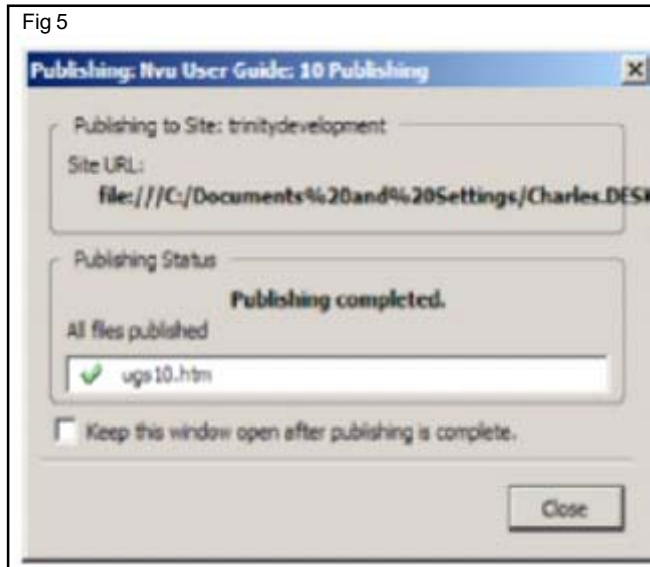
- 4 If it is the first time to upload the page, and if it includes images or uses external style sheets, check the box 'Include images and other files'. (If it is not the first time and these other files have not changed the box may be left unchecked.) The files will be placed in the same directory as the page. If you want them to go in a sub-directory check the box 'Use this site sub-directory' and name the directory. In this case the directory will be created if needed.
- 5 You should not need to refer to the 'Settings' tab as the data should be collected via the Site name you have selected but you may view the data and change if you wish.
- 6 Click 'Publish'. A 'Publishing' window will appear and uploading will commence. (If you are on a dial-up connection this will be connected.)
- 7 Within a short time you should receive confirmation of correct publication similar to the first figure above.
- 8 One possible source of problems occurs if you are prevented from accessing the site by a firewall. In this case you may receive a 'Publishing failed' message similar to that shown in the second figure.

The 'Troubleshooting' button takes you to the KompoZer help system but this is short of aid in this area at present.

Other possible problems include

- Some required files are missing
- File or directory names incorrect e.g. Wrong case

Once you have published a page, if you need to publish it again, your settings (e.g. subdirectories) should be remembered by KompoZer. You will not see steps 2 to 4 again unless changes have been made to the page. Fig 5 and Fig 6 shows publishing window.



Concepts of Animation and Multimedia files in JavaScript

Objectives : At the end of this lesson you shall be able to

- know animation settings in JavaScript
- explain multimedia in JavaScript.

Animation

Styling the Elements

To make an animation possible, the animated element must be animated relative to a "parent container".

The container element should be created with style = "position: relative".

The animation element should be created with style = "position: absolute".

Example

```
<!Doctype html>
<html>
<style>
#myContainer{
width: 400px;
height: 400px;
position: relative;
background: pink;
}
#myAnimation {
width: 50px;
height: 50px;
position: absolute;
background: green;
}
</style>
<body>
<h1>My First JavaScript Animation</h1>
<div id="myContainer">
<div id="myAnimation"></div>
</div>
</body>
</html>
?
```

The Animation Code

JavaScript animations are done by programming gradual changes in an element's style. The changes are called by a timer. When the timer interval is small, the animation looks continuous. The basic code is:

Example

```
var id = setInterval(frame, 5);
function frame() {
if (/* test for finished */) {
clearInterval(id);
} else {
/* code to change the element style */
}
}
```

Create the Animation Using JavaScript

Example

```
<style>`
#myContainer{
width: 400px;
height: 400px;
position: relative;
background: pink;
}
#myAnimation {
width: 50px;
height: 50px;
position: absolute;
background-color: green;
}
</style>
<body>
<p>
<button onclick="myMove()">Click Me</button>
</p>
```

```

<div id="myContainer">
<div id="myAnimation"></div>
</div>
<script>
function myMove(){
var elem = document.getElementById("myAnimation");
var pos = 0;
var id = setInterval(frame, 10);
function frame(){
if (pos == 350) {
clearInterval(id);
} else {
pos++;
elem.style.top = pos + 'px';
elem.style.left = pos + 'px';
}
}
}
</script>
</body>
</html>

```

Multimedia files

What is Multimedia?

Multimedia comes in many different formats. It can be almost anything you can hear or see. Web pages often contain multimedia elements of different types and formats.

Examples: Images, music, sound, videos, records, films, animations and more.

Multimedia Formats

Multimedia elements (like audio or video) are stored in media files. The most common way to discover the type of a file, is to look at the file extension. Multimedia files have formats and different extensions like: .swf, .wav, .mp3, .mp4, .mpg, .wmv, and .avi.

Playing Videos in HTML

To show a video in HTML, use the <video> element:

Example

```

<video width="320" height="240" controls>
<source src="movie.mp4" type="video/mp4">
<source src="movie.ogg" type="video/ogg">
Your browser does not support the video tag.
</video>

```

How it Works

The controls attribute adds video controls, like play, pause, and volume. It is a good idea to always include width and height attributes. If height and width are not set, the page might flicker while the video loads. The <source> element allows you to specify alternative video files which the browser may choose from. The browser will use the first recognized format. The text between the <video> and </video> tags will only be displayed in browsers that do not support the <video> element.

HTML <video> Autoplay

To start a video automatically use the autoplay attribute:

Example

```

<video width="320" height="240" autoplay>
<source src="movie.mp4" type="video/mp4">
<source src="movie.ogg" type="video/ogg">
Your browser does not support the video tag.
</video>

```

Note: Autoplay attribute does not work in mobile devices like iPad and iPhone

HTML Video - Media Types

File Format	Media Type
MP4	video/mp4
WebM	video/webm
Ogg	video/ogg

HTML Video - Methods, Properties, and Events

HTML5 defines DOM methods, properties, and events for the <video> element. This allows you to load, play, and pause videos, as well as setting duration and volume. There are also DOM events that can notify you when a video begins to play, is paused, etc.

HTML5 Video Tags

Tag	Description
<video>	Defines a video or movie
<source>	Defines multiple media resources for media elements, such as <video> and <audio>
<track>	Defines text tracks in media players

Audio on the Web

The HTML5 <audio> element specifies a standard way to embed audio in a web page.

The HTML <audio> Element

To play an audio file in HTML, use the <audio> element:

Example

```
<audio controls>
```

```
<source src="horse.ogg" type="audio/ogg">
```

```
<source src="horse.mp3" type="audio/mpeg">
```

Your browser does not support the audio element.

```
</audio>
```

HTML Audio - How It Works

The controls attribute adds audio controls, like play, pause, and volume. The <source> element allows you to specify alternative audio files which the browser may choose from. The browser will use the first recognized format. The text between the <audio> and </audio> tags will only be displayed in browsers that do not support the <audio> element.

HTML Audio - Media Types

File Format	Media Type
MP3	audio/mpeg
OGG	audio/ogg
WAV	audio/wav

HTML Audio - Methods, Properties, and Events

HTML5 defines DOM methods, properties, and events for the <audio> element. This allows you to load, play, and pause audios, as well as set duration and volume. There are also DOM events that can notify you when an audio begins to play, is paused, etc.

HTML5 Audio Tags

Tag	Description
<audio>	Defines sound content
<source>	Defines multiple media resources for media elements, such as <video> and <audio>