

# **BISHOP SCOTT BOYS' SCHOOL**

(Affiliated to CBSE, New Delhi) Affiliation No.: 330726, School Campus: Chainpur, Jaganpura,  
By-Pass, Patna 804453.  
Phone Number: 7061717782, 9798903550. ,  
Web: [www.bishopscottboyschool.com](http://www.bishopscottboyschool.com) Email: [info@bishopscottboyschool.com](mailto:info@bishopscottboyschool.com)



## **ONLINE STUDY MATERIAL**

### **SUBJEC-COMPUTER**

**SESSION-2020-21**

**CLASS-VIII**

## **TOPIC: HTML5-INTRODUCTION (CHAPTER-6)**

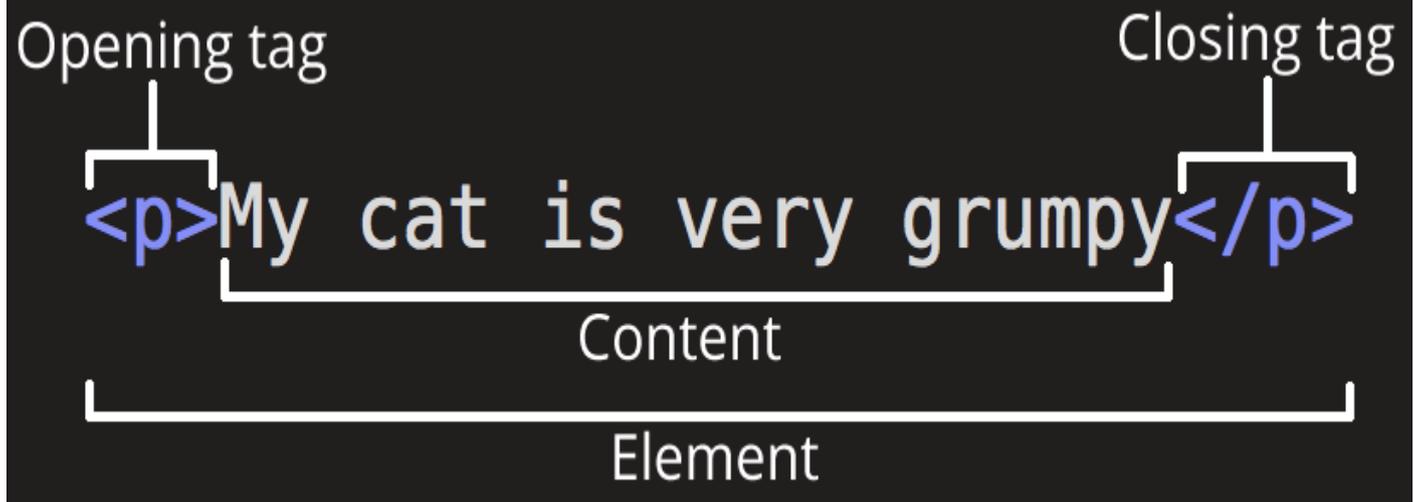
### **DAY-1**

#### **What is HTML?**

HTML (Hypertext Markup Language) is not a programming language. It is a *markup language* that tells web browsers how to structure the web pages you visit. It can be as complicated or as simple as the web developer wants it to be. HTML consists of a series of elements, which you use to enclose, wrap, or *mark up* different parts of content to make it appear or act in a certain way. The enclosing tags can make content into a hyperlink to connect to another page, italicize words, and so on. For example, consider the following line of text:

#### Anatomy of an HTML element

Let's further explore our paragraph element from the previous section:



The anatomy of our element is:

**The opening tag:** This consists of the name of the element (in this example, *p* for paragraph), wrapped in opening and closing angle brackets. This opening tag marks where the element begins or starts to take effect. In this example, it precedes the start of the paragraph text.

**The content:** This is the content of the element. In this example, it is the paragraph text.

**The closing tag:** This is the same as the opening tag, except that it includes a forward slash before the element name. This marks where the element ends. Failing to include a closing tag is a common beginner error that can produce peculiar results.

The element is the opening tag, followed by content, followed by the closing tag.

## ❖ VIDEO-LINKS

<https://www.youtube.com/watch?v=9gTw2EDkaDQ>

## DAY-2

### Attributes

Elements can also have attributes. Attributes look like this:



Attributes contain extra information about the element that won't appear in the content. In this example, the `class` attribute is an identifying name used to target the element with style information. An attribute should have:

- A space between it and the element name. (For an element with more than one attribute, the attributes should be separated by spaces too.)
- The attribute name, followed by an equal sign.
- An attribute value, wrapped with opening and closing quote marks.

### Active learning: Adding attributes to an element

Another example of an element is `<a>`. This stands for *anchor*. An anchor can make the text it encloses into a hyperlink. Anchors can take a number of attributes, but several are as follows:

- `href`: This attribute's value specifies the web address for the link. For example: `href="https://www.mozilla.org/"`.
- `title`: The `title` attribute specifies extra information about the link, such as a description of the page that is being linked to. For example, `title="The Mozilla homepage"`. This appears as a tooltip when a cursor hovers over the element.
- `target`: The `target` attribute specifies the browsing context used to display the link. For example, `target="_blank"` will display the link in a new tab. If you want to display the linked content in the current tab, just omit this attribute.

Edit the line below in the *Input* area to turn it into a link to your favorite website.

1. Add the `<a>` element.
2. Add the `href` attribute and the `title` attribute.
3. Specify the `target` attribute to open the link in the new tab.

You'll be able to see your changes update live in the *Output* area. You should see a link—that when hovered over—displays the value of the `title` attribute, and when clicked, navigates to the web address in the `href` attribute. Remember that you need to include a space between the element name, and between each attribute.

If you make a mistake, you can always reset it using the *Reset* button. If you get really stuck, press the *Show solution* button to see the answer.

### Boolean attributes

Sometimes you will see attributes written without values. This is entirely acceptable. These are called Boolean attributes. Boolean attributes can only have one value, which is generally the same as the attribute name. For example, consider the `disabled` attribute, which you can assign to form input elements. (You use this to *disable* the form input elements so the user can't make entries. The disabled elements typically have a grayed-out appearance.) For example:

```
<input type="text" disabled="disabled">
```

As shorthand, it is acceptable to write this as follows:

```
<!-- using the disabled attribute prevents the end user from entering text into the input box -->
```

```
<input type="text" disabled>
```

```
<!-- text input is allowed, as it doesn't contain the disabled attribute -->
```

```
<input type="text">
```

For reference, the example above also includes a non-disabled form input element. The HTML from the example above produces this result:

## Omitting quotes around attribute values

If you look at code for a lot of other sites, you might come across a number of strange markup styles, including attribute values without quotes. This is permitted in certain circumstances, but it can also break your markup in other circumstances. For example, if we revisit our link example from earlier, we could write a basic version with *only* the `href` attribute, like this:

```
<a href=https://www.mozilla.org/>favorite website</a>
```

However, as soon as we add the `title` attribute in this way, there are problems:

```
<a href=https://www.mozilla.org/ title=The Mozilla homepage>favorite website</a>
```

As written above, the browser misinterprets the markup, mistaking the `title` attribute for three attributes: a title attribute with the value *The*, and two Boolean attributes, `Mozilla` and `homepage`. Obviously, this is not intended! It will cause errors or unexpected behavior, as you can see in the live example below. Try hovering over the link to view the title text! Always include the attribute quotes. It avoids such problems, and results in more readable code.

## Single or double quotes?

In this article you will also notice that the attributes are wrapped in double quotes. However, you might see single quotes in some HTML code. This is a matter of style. You can feel free to choose which one you prefer. Both of these lines are equivalent:

```
<a href="http://www.example.com">A link to my example.</a>
```

```
<a href='http://www.example.com'>A link to my example.</a>
```

Make sure you don't mix single quotes and double quotes. This example (below) shows a kind of mixing quotes that will go wrong:

```
<a href="http://www.example.com">A link to my example.</a>
```

However, if you use one type of quote, you can include the other type of quote *inside* your attribute values:

```
<a href="http://www.example.com" title="Isn't this fun?">A link to my example.</a>
```

To use quote marks inside other quote marks of the same type (single quote or double quote), use [HTML entities](#). For example, this will break:

```
<a href='http://www.example.com' title='Isn't this fun?'>A link to my example.</a>
```

Instead, you need to do this:

```
<a href='http://www.example.com' title='Isn&#39;t this fun?'>A link to my example.</a>
```

## Anatomy of an HTML document

Individual HTML elements aren't very useful on their own. Next, let's examine how individual elements combine to form an entire HTML page:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>My test page</title>
  </head>
  <body>
    <p>This is my page</p>
  </body>
</html>
```

Here we have:

1. `<!DOCTYPE html>`: The doctype. When HTML was young (1991-1992), doctypes were meant to act as links to a set of rules that the HTML page had to follow to be considered good HTML. Doctypes used to look something like this:

2. `<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"`

```
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

More recently, the doctype is a historical artifact that needs to be included for everything else to work right. `<!DOCTYPE html>` is the shortest string of characters that counts as a valid doctype. That is all you need to know!

3. `<html></html>`: The `<html>` element. This element wraps all the content on the page. It is sometimes known as the root element.
4. `<head></head>`: The `<head>` element. This element acts as a container for everything you want to include on the HTML page, **that isn't the content** the page will show to viewers. This includes keywords and a page description that would appear in search results, CSS to style content, character set declarations, and more. You'll learn more about this in the next article of the series.
5. `<meta charset="utf-8">`: This element specifies the character set for your document to UTF-8, which includes most characters from the vast majority of human written languages. With this setting, the page can now handle any textual content it might contain. There is no reason not to set this, and it can help avoid some problems later.
6. `<title></title>`: The `<title>` element. This sets the title of the page, which is the title that appears in the browser tab the page is loaded in. The page title is also used to describe the page when it is bookmarked.
7. `<body></body>`: The `<body>` element. This contains *all* the content that displays on the page, including text, images, videos, games, playable audio tracks, or whatever else.

### ❖ VIDEO-LINKS

<https://www.youtube.com/watch?v=9gTw2EDkaDQ>

## HTML - Images

Images are very important in a web page. It is therefore important to use them in the right way. Insert images using the tag `<img/>`.

```
HTML 
```

### Demo



## HTML - img src

"src" is shortcut for "source". This attribute is used to indicate the image location. As explained in the [tutorial about links](#) you can use any URL to point to the file.

Local source	Description
<code>src="image.jpg"</code>	the picture is located on the same level as .html file
<code>src="../image.jpg"</code>	the picture is located in previous level as .html file.
<code>src="../img/image.jpg"</code>	the picture is located in the "img" folder in a previous level as .html file

You may also address the full URL. For example:

```
HTML src="http://www.tutorialehtml.com/assets_tutorials/img/image.jpg"
```

The URL pattern is not recommended because in case you change the domain you will have to change as well the addresses of all image files. Or if the image is stored on another domain just use `src="http://www.domain.com/image.jpg"`.

The fact that storing the image on an alternative server, presents a great advantage if the space provided for your domain is a limited one.

# HTML - "alternative" attribute for images

Attribute "alt" is used to display text instead of image in case your browser for some reason can not display the picture or when a user has the "ShowImage" option unselected.

```
HTML 
```

## Demo



# HTML - the height and width of an image

To determine the height and width of an image is used and "height" attributes "width". Logical, no!

```
HTML 
```

## Demo



# HTML - horizontal and vertical alignment of an image

For this we will use the attributes align and valign. The options offered by these attributes are:

## 1. Align (horizontal)

- right
- left
- center

## 2. Valign (vertical)

- top
- bottom
- center

```
HTML
```

```
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.</p>
```

```
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.</p>
```

```
<p>Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.</p>
```

## Demo

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.



Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.

## HTML - Using images as links

This is only an introduction, we will talk more about this in the next tutorial. The images are very useful as links and can be formed as follows

```
HTML<a href="http://www.tutorialehtml.com/" title="HTML link image example">
```

```

```

```
</a>
```

## Demo



The image above will send to the home page. You can change it to your start page and here you have a picture with a link to your start page

# HTML - Thumbnails

Thumbnails are miniature version (less kilobytes) of images that have a much higher quality. To create a thumbnail, save the image in a lower quality than the original one. then link this small image to the original high quality image.

```
HTML <a href="/assets_tutorials/img/image.jpg" title="HTML link image thumbnail example">  
  
      
  
</a>
```

## Demo



## VIDEO LINK:

<https://www.youtube.com/watch?v=9gTw2EDkaDQ>

## DAY-4

# HTML - Audio

Some time ago it was quite complicated to insert music or sounds on a web page. Now, that problem is solved, adding sounds being quite simple.

```
HTML <embed height="60" type="audio/midi" width="144" src="audio.mp3" volume="60" loop="false"  
autostart="false" />
```

It is recommended that both height and width to be directly proportional to avoid problems.

To hide the player the value of the hidden attribute (which is false at the moment) will be replaced by true. This thing is done if you are absolutely sure that the user does not wish to stop the sound.

## Controlling and manipulating the audio player

Let's have a look on the above example:

```
HTML <embed height="60" type="audio/midi" width="144" src="audio.mp3" volume="60" loop="false"  
autostart="false" />
```

- **autostart** - sets whether the sound will start immediately after the web page is loaded. Can have the true or false value.
- **loop** - sets whether the sound will be replayed forever. Can have the true or false value.
- **volume** - can have any value from 1 to 100.

You can play a little with these attributes to better memorize and understand them, and also be aware that a higher volume may be annoying for the user.

## HTML 5 Audio tag

With the introduction of HTML5, audio tag has spread very quickly and it is very well supported by current browsers. Audio tag is used to add sound and music in the html page. At the moment the **audio** tag, supports 3 types of audio files:

- mp3 - MIME-type audio/mpeg
- wav - MIME-type audio/wav
- ogg - MIME-type audio/ogg

**HTML** <audio controls>

```
<source src="/assets_tutorials/media/Loreena_Mckennitt_Snow_56bit.mp3" type="audio/mpeg">
```

```
<source src="/assets_tutorials/media/Loreena_Mckennitt_Snow_56bit.ogg" type="audio/ogg">
```

Your browser does not support the audio tag.

</audio>

### Demo

Browser	MP3	Wav	Ogg
Internet Explorer			
Chrome			
Firefox			
Safari			
Opera			

Here you can find a simple yet powerfull generator for the [audio html5 tag](#)

A video file can be inserted in an HTML page in two ways. The first way would be using the <embed/> tag. This tag does not need an ending/finishing tag. It works, mostly, the same as a tag for a photo.

```
HTML <embed src="example.mpeg" autostart="false" height="30" width="144" />
```

Also, you can insert a video file using a link.

```
HTML <a href="example.mpeg" title="Download movie">film name</a>
```

## Supported extensions for the embed tag

These are:

- .swf - made by Macromedia Flash
- .wmv - Microsoft Windows Media Video
- .mov - Quick Time Movie, belongs to Apple
- .mpeg - created by Moving Pictures Expert Group.

The most used are .mpeg and .swf, because of the compact format.

## Attributes of the embed tag

- **autostart** - sets whether the file will run automatically after the page is loaded. Can have the true or false value
- **hidden** - sets whether the buttons are hidden or not. Can have the true or false value.
- **volume** - can have any value from 0 to 100
- **loop** - sets whether the file will be replayed after it is finished. Can have the true or false value.
- **playcount** - this establishes how many times the file will be replayed. For example playcount="2" means it will be replayed two times and after it will stop.

## Inserting a YouTube video using embed tag

You can do that very easy because YouTube provides you with the HTML code.

```
HTML <object width="425" height="344" data="https://www.youtube.com/embed/N4koEPJ0bjU"></object>
```

```
HTML <embed src="https://www.youtube.com/embed/N4koEPJ0bjU allowfullscreen="true" width="425" height="344">
```

```
HTML <iframe width="425" height="344" src="https://www.youtube.com/embed/N4koEPJ0bjU" frameborder="0" allowfullscreen></iframe>
```

Note that the video has a fixed width and height and it will not be responsive. To make the YouTube video responsive, you can wrap it into a container like this:

```
HTML <div style="position:relative; padding-bottom:56.25%; padding-top:30px; height:0; overflow:hidden;">
```

```
<iframe width="425" height="344" src="https://www.youtube.com/embed/N4koEPJ0bjU" frameborder="0" allowfullscreen
```

```
style="position:absolute; top:0; left:0; width:100%; height:100%;">
```

```
</iframe>
```

```
</div>
```

## Demo

YOu can also take all style to a class or use JavaScript for cleaner markup, but this is not the subject of this tutorial.

You will also find out that embed YouTube scripts are not XHTML valid. But don't worry, give it a search and you will find plenty of tools to generate valid XHTML codes for videos on youtube.

You can also find here a good tutorial about [inserting YouTube videos](#) into a CMS like WordPress, Joomla or Drupal, PowerPoint presentations, Gmail and some other cool things

# HTML5 video tag

With the introduction of HTML5, the video tag spread out very quickly with good support by modern browsers. If after reading all this you are still not comfortable in creating your own player using the HTML5 video tag, you can use a simple yet powerful generator, to [create a video tag for the player](#).

Video tag is used to add a video to the html page. At the moment the **HTML5 video tag**, supports 3 types of video files:

- **mp4** - MIME-type video/mp4
- **webm** - MIME-type video/webm
- **ogg** - MIME-type video/ogg

```
HTML<video width="320" height="240" controls>
```

```
<source src="/assets_tutorials/media/Shاون-the-Sheep-The-Movie-Official-Trailer.mp4" type="video/mp4">
```

```
<source src="/assets_tutorials/media/Shاون-the-Sheep-The-Movie-Official-Trailer.ogg" type="video/ogg">
```

```
Your browser does not support the video tag.
```

```
</video>
```

## Demo

At the moment the video tag is supported very well by major modern browsers.

Browser	MP4	WebM	Ogg
Internet Explorer			

Chrome			
Firefox			
Safari			
Opera	(Opera 25+)		

**VIDOE LINK:**

<https://www.youtube.com/watch?v=9gTw2EDkaDQ>

**DAY-5**

**A. Write one line code to do the following.**

1. Give the title of webpage as "HTML5 – SUMMARY".
2. Use bgcolor to set the background colour of the whole webpage.
3. Change the font colour of a text to blue and type as "Old Bookman Style".
4. Set the audio file "guitar.mp3" to play automatically on webpage.
5. Set the image "school.jpg" on right side and with border 5.

**B. What happens when?**

1. "Data" keyword in front of custom attribute is not written.
2. HTML file saved with extension .txt is opened in any browser.
3. <font> tag is not closed after use.
4. <source> child element is not included in <picture> tag.
5. <!DOCTYPE html> is not written as first line of html5 document.

**C. True/ False.**

1. HTML5 is not backward compatible.
2. Attributes can be created with or without values.
3. We can write HTML code only in Notepad.

4. <audio> tag can work without controls attribute.

5. <img> cannot be used to insert animated images

**1. Answer the following questions.**

1. Give three important points to remember for writing HTML5 code.
2. Give two different ways of inserting image in HTML5. Give a small code to support your answer.
3. What are standard attributes? Name any three.
4. What is backward compatibility? Is HTML5 backward compatible?
5. What are structural elements? Name any three.