MODULE I

Web 2.0: Basics-RIA Rich Internet Applications - Collaborations tools - Understanding websites and web servers: Understanding Internet – Difference between websites and web server- Internet technologies Overview –Understanding the difference between internet and intranet; HTML and CSS: HTML 5.0 , XHTML, CSS 3.

Introduction to Web 2.0

Web 1.0

- Web 1.0 refers to the first stage in the World Wide Web.
- Entirely made up of web pages connected by hyperlinks.
- A set of static web sites that were not yet providing interactive content.
- Used as “Information portal”.

  Examples: Amazon,Yahoo,Personal web pages

Web 2.0

- Web 2.0 is the term used to describe a variety of websites and applications that allow anyone to create and share online information or material they have created.
- It allows people to create, share, collaborate & communicate.
- Allows everyone to produce their content.
- Gives the users the possibility to control their data.
- **Web 2.0** allows groups of people to work on a document or spreadsheet simultaneously.
- In the background a computer keeps track of whom and what changes where and when.
- **Web**-based applications can be accessed from anywhere.
- **Web 2.0 Examples**
  - Web applications(GoogleDocs,Flickr)
  - Video sharing sites(YouTube)
  - Wikis(MediaWiki)
  - Blogs(WordPress)
  - Social networking(Facebook)
  - Micro blogging(Twitter)
  - Hosted services(Google Maps)

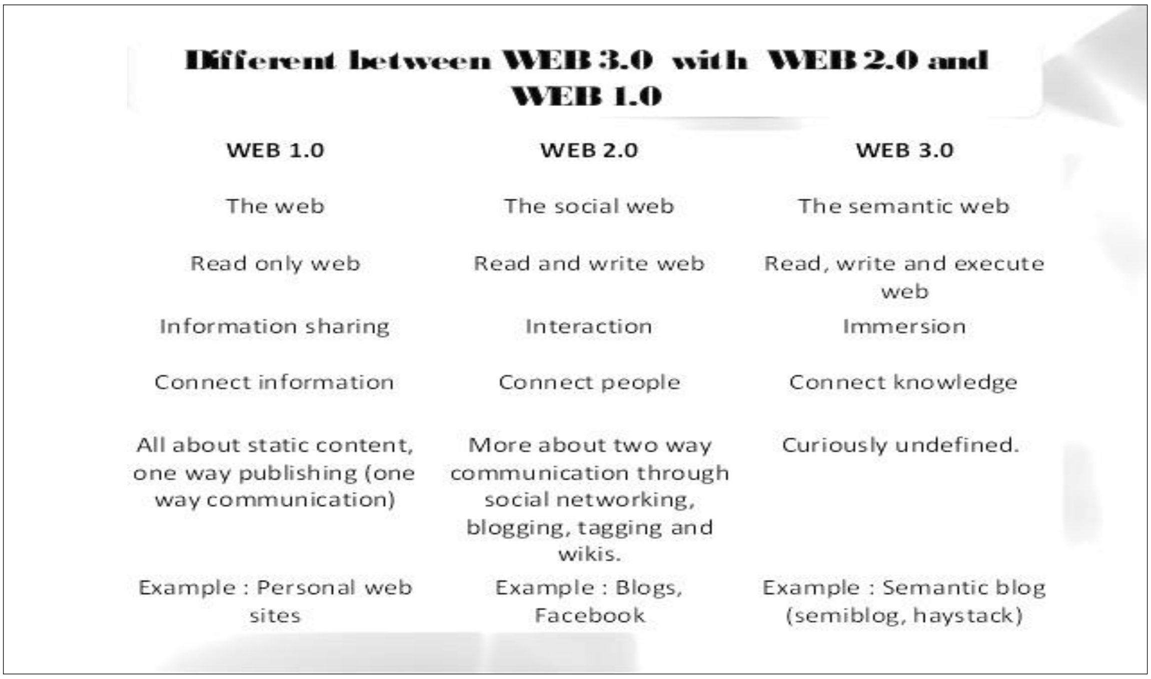
Advantages and disadvantages of Web 2.0

Web 2.0 offers the following pros:

- **Dynamic content.** Web 2.0 showcases dynamic content that users can interact with and modify, unlike the restricted, read-only format of Web 1.0.
- **Increased social networking.** Web 2.0 enables people to participate in discussions, share information with friends and family, and stay in touch with people all around the globe.
- **Ease of use and information sharing.** With Web 2.0, users can easily use, update and share information with a few clicks. Any editing done on the internet can also be tracked.
- **Improved marketability.** Web 2.0 enables business owners to improve user experience by creating responsive websites. They can also promote their products online and increase their marketability through interactive advertisement campaigns.
- **Improved quality of education.** Web 2.0 opens doors to interactive learning and virtual classrooms. For example, students can extend their range of learning by using online calculators while solving math problems.

Web 2.0 offers the following cons:

- **Cybersecurity risks.** The increased online collaboration that comes with Web 2.0 puts users at increased risk of downloading malicious viruses and adware and getting afflicted by various cyber attacks, such as spam and phishing attacks.
- **Information overload.** Web 2.0 is a sea of information that grows incessantly. This can confuse readers and affect the reliability of the content, as the variety and volume of the information are vast.
- **Ethics and credibility.** Critics of Web 2.0 maintain that it makes it too easy for the average person to affect online content, which can impact the credibility, ethics and even legality of web content. The extent of data sharing and gathering also raises concerns about privacy and security.



Different between WEB 3.0 with WEB 2.0 and WEB 1.0

WEB 1.0	WEB 2.0	WEB 3.0
The web	The social web	The semantic web
Read only web	Read and write web	Read, write and execute web
Information sharing	Interaction	Immersion
Connect information	Connect people	Connect knowledge
All about static content, one way publishing (one way communication)	More about two way communication through social networking, blogging, tagging and wikis.	Curiously undefined.
Example : Personal web sites	Example : Blogs, Facebook	Example : Semantic blog (semiblog, haystack)

What is RIA?

- Rich Internet Applications(RIAs) are web applications that have the features and functionality of traditional desktop applications.
- RIAs typically provide a“no-refresh”look to the user interface.
- RIA provides HDu X–HighDefinition UsereXperience.

What is special in RIA?

- RIA works in Web.
- RIA appears–never refreshes.
- RIA reduces network traffic.
- RIA is rich.
- RIA makes it easy.

RIA Tools

- **AdobeFlex**→a highly productive, open source application framework for building and maintaining expressive web applications that deploy consistently on all major browsers, desktops and devices.
- **OpenLaszlo**→a open source platform for the development and delivery of Rich Internet Applications, consists of the LZX programming language and the OpenLaszlo Server.
- **Microsoft Silverlight** → a powerful development tool for creating engaging, interactive user experiences for Web and mobile applications.
- **JavaFX**→a software platform for creating and delivering desktop applications, as well as Rich Internet Applications(RIAs) that can run a cross

a wide variety of devices.

- The logos of RIA tools are shown in figure1.3.

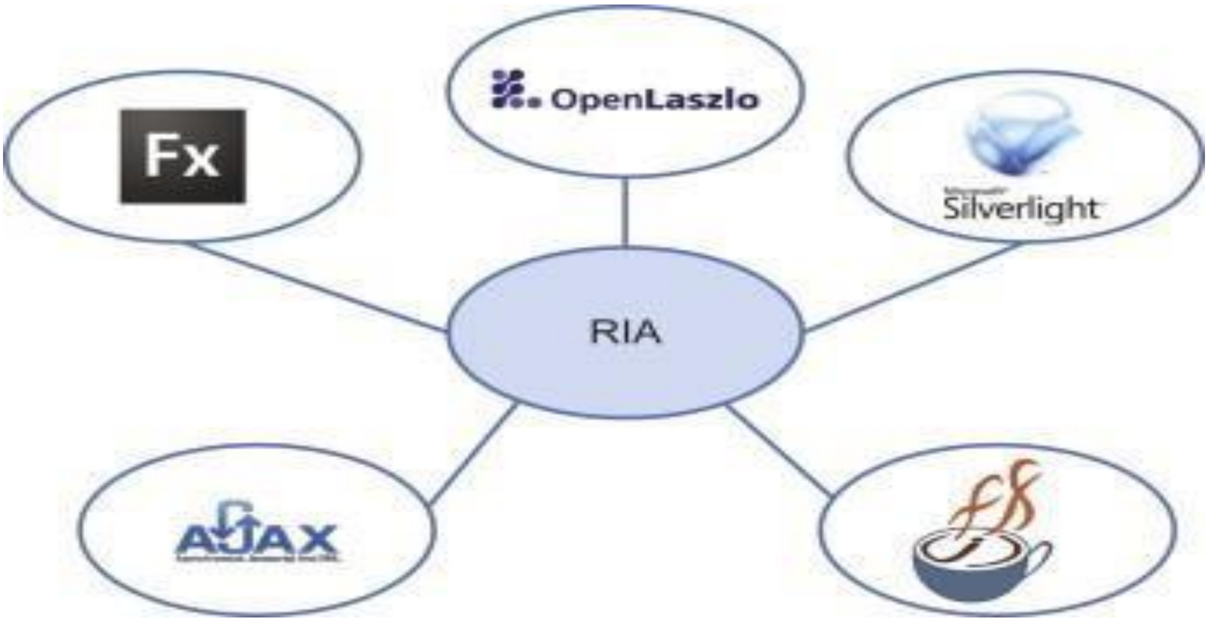


Figure1.3 RIA Tools

Platforms of RIA

- AdobeFlash
- Java
- MicrosoftSilverlight
- AJAX

Characteristics of RIA

- Rich user experience
- Interactive- An RIA can use a wider range of controls that allow greater efficiency and enhance the user experience.
  - Users can interact directly with page elements through editing or drag-and-drop tools.
  - They can also do things like pan a cross a map or other image.
- Responsive
- Low maintenance
- Is plastic(changing,transforming)
- Breaks walled gardens
- **Partial-page updating:** RIAs incorporate additional technologies, such as real-time

streaming, high-performance client-side virtual machines, and local caching mechanisms that reduce latency (wait times) and increase responsiveness.

- **Better feedback:** Because of their ability to change parts of pages without reloading, RIAs can provide the user with fast and accurate feedback, real-time confirmation of actions and choices, and informative and detailed error messages.
- **Consistency of look and feel:** With RIA tools, the user interface and experience with different browsers and operating systems can be more carefully controlled and made consistent.
- **Offline use**: When connectivity is unavailable, it might still be possible to use an RIA if the app is designed to retain its state locally on the client machine.
- **Caching**

  RIA client has the ability of keeping the server information during a period of time improving application performance and UI responsiveness.
- **Security**

  RIAs should be as secure as any other web application, and the framework should be well equipped to enforce limitations appropriately when the user lacks there quired privileges, especially when running with in a constrained environment such as a sandbox.
- **Advanced Communication**

  Sophisticated communications with supporting servers through optimized network protocols can considerably enhance the user experience.
- **Rapid Development**

  An RIA Framework should facilitate rapid development of a rich user experience through hits easy-to-use interfaces in ways that help developers.
- **Improved Features**

  RIA allow programmers to embed various functionalities in graphics-based web pages that look fascinating and engaging like desktop applications.

  RIA provide complex application screens on which various mixed media, including different fonts, vector graphic and bitmap files online conferencing etc. are paused by using different modern development tools.

Business Benefits of RIA

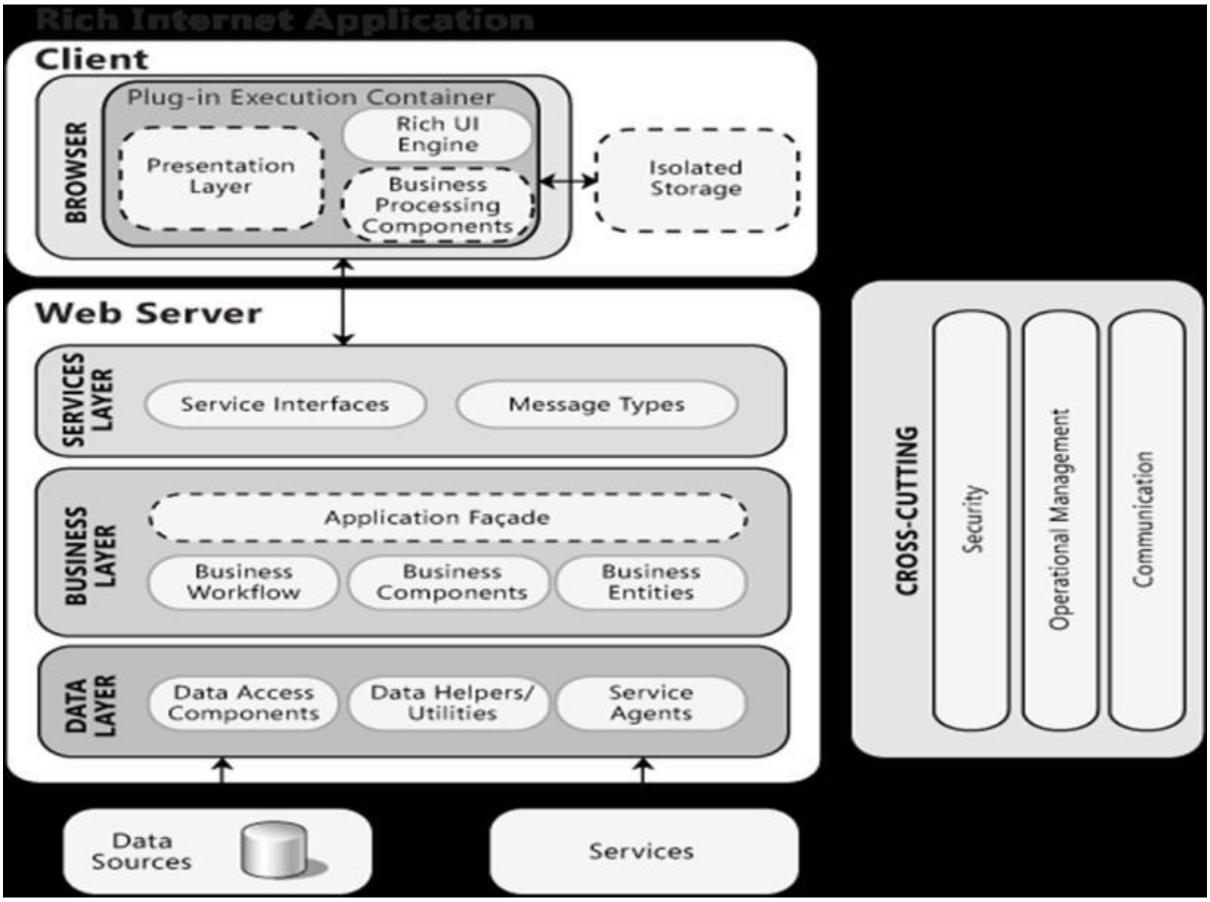
- Enables more transactions
- Retains customers
- Low operational cost
- Improves performance

Application Benefits:

- Add value to your applications
- Meet your customer expectations
- Simplify & speed up processes
- Get regular and dedicated visitors
- Save cost son Band width
- Increase your productivity

Architecture of RIA

Figure 1.7 Architecture of RIA



A typical Rich Internet Application is decomposed into three layers is shown in Figure 1.7.

• **Presentation layer** - contains UI and presentation logic components

• **Business layer** - contains business logic, business work flow and business entities components

• **Data layer** – Contains data access and service agent components.

• It is common in RIAs to move some of business processing and even the data access code to the client.

• The client may in fact contain some or all of the functionality of the business and data layers, depending on the application scenario.

UI components

• Users can interact with the application
• Format data and render data to users
• Acquire and validate data

Application façade

• To combine multiple business operations into single message-based operation
• The facade can be accessed from the presentation layer using a range of communication technologies.

Data access logic components
• Abstract the logic needed to access the underlying data stores.
• This centralizes data access functionality, makes it easier to configure and maintain
Data helpers/utilities
• For centralizing generic data access functionality (managing db connection and caching data).
• Data source specific helper components can be designed to abstract the complexity of accessing the db.
• Service Agents – Basic mapping between the format of the data exposed by the service and the format your application needs.
• Business components – implement the business logic of the application. Implement business rules and perform business tasks.
• Business work flows – Define and coordinate long running, multistep business processes. They can be implemented using business process management tools.
• Design considerations
• Choose an appropriate technology based on application requirements.(eg . Windows Forms, OBA, WPF)
• Separate presentation logic from interface implementation

- Eases maintenance
- Promotes reusability
- Improves testability

• Identify the presentation tasks and presentation flows

- Helps to design each screen and each step-in multi-screen or wizard
Processes.

• Design to provide a suitable and usable interface

- Features like layout, navigation, choice of controls to maximize
accessibility and usability.

• Extract business rules and other tasks not related to the interface.
• Reuse common presentation logic – (Libraries that contain templates, generalized client-side validation functions and helper classes)
• Loose couple your client from any remote services it uses.
- Use a message-based interface to communicate with services located on separate physical tiers.
• Avoid tight coupling to objects in other layers – Use the abstract base classes or messaging when communicating with other layers of the application.
• Reduce round trips when accessing remote layers – Use coarse grained methods and execute them asynchronously to avoid blocking or freezing the UI.

3.ONLINE COLLABORATION TOOLS

An online collaboration tool is a software platform or an application specifically designed to incorporate individuals into a team. The users of their accounts can **delegate tasks**, **share insights**, **communicate**, and **work in this digital environment**.

The prime aim of using an online collaboration tool is to keep the **members connected** via a **technology-driven medium**. This medium maintains **proper communication** to elevate the efficiency of a team.

Types of Online Collaboration Tools

1. File sharing

File sharing tools offer a common platform for sharing files and editing them. These tools help ensure everyone has the latest version of a document and can collaborate in real-time. Examples include Google Drive and Dropbox, which also offer options for commenting and version history.

2. Project management

Teammates involved in a project can connect and manage task completion with collaboration. They offer features like task assignments, progress tracking, and communication channels. Tools like Trello, Asana, and Monday.com make it easier to visualize project stages and keep everyone on the same page.

3. Time management

Such tools provide a comprehensive structure for day-to-day tasks. They remind users of deadlines for task completion. They remind users of deadlines for task completion and help prioritize tasks to improve productivity and time management. They often include features such as calendars, reminders, and time tracking. Apps like Toggl, Clockify, and Google Calendar can help individuals and teams plan their schedules and avoid missing important tasks.

4. Cloud storage

Cloud storage is a kind of collaboration tool to store and access files on the go. Cloud storage tools provide a secure way to store large amounts of data online, making it accessible from any device with an internet connection. These tools support collaboration by allowing multiple users to access and update files simultaneously.

5. Document collaboration

Such tools offer a digital platform to create and share documents. Such platforms also allow users to share feedback. These platforms also allow users to share feedback, make real-time edits, and collaborate on documents simultaneously, enhancing teamwork and communication. They often include features like commenting, suggesting edits, and tracking changes.

4.Web Server

A "web server" is a computer system that stores and delivers website content (like web pages, images, and videos) to users when requested, while a "website" is a collection of related web pages hosted on a web server, accessible through a unique URL, essentially the online presence of a person or organization; think of a web server as the kitchen where the food is prepared and a website as the restaurant serving the food to customers.

Key points:

- **Function:** A web server is the technical infrastructure that handles requests for web pages and sends them to users, while a website is the actual content and design that users see when they visit a URL.
- **Multiple websites on one server:** A single web server can host multiple different websites.

Example:

- **Web server:**

An Apache server running on a computer, which can host various websites like "google.com", "amazon.com", and "facebook.com".

- **Website:**

The specific online presence of "google.com", which is a collection of web pages accessible through the "google.com" domain.

Web Sites

In a Rich Internet Application (RIA), a "website" refers to the collection of web pages that users interact with, while a "web server" is the computer system that stores and delivers those web pages upon request, allowing for dynamic content and interactions that feel more like a desktop application, unlike traditional static websites; essentially, the web server is the engine behind the RIA, processing user actions and updating the displayed content in real-time

Types of Web Sites There are many types of Web sites, each catering to a particular type of content or use. Hence, few illustrative but not exhaustive cases are given below:

1) Blog (Web Log): site generally used to post online diaries which may include discussion forums.
2) Social Networking Site: where users could communicate with one another and share media, such as pictures, videos, music and blogs with other users. These may include games and Web applications.
3) Wiki Site: which users collaboratively edit (such as Wikipedia and Wikihow).
4) Web Portal: that provides a starting point or a gateway to other resources on the Internet.
5) Search Engine Site: a site that provides general information and is intended as a gateway or lookup for other sites like Google, Yahoo, Bing search engines.
6) Education Site: where teachers, students, or administrators can post information about current events at or involving their school.

EXAMPLE:

- Imagine a library:
    - **Web Server:** The library building itself, where all the books (web pages) are stored and organized.
    - **Website:** A specific collection of books on a particular topic (like science fiction) within the library.

How it works:

- When you type a website address in your browser, your computer sends a request to the web server where that website is hosted.
- The web server then retrieves the requested web pages from its storage and sends them back to your browser, allowing you to see the website.

WEBSITE	**WEBSERVER**
It is collection of webpages where one can get the information they want and perform various things.	The Web server accepts, approves, and responds to the request made by the web browser for a web document or service.
SIt includes content, information, images, videos, hyperlinks to visit different web pages, etc.	The web server is responsible for connecting websites and web browsers.
Its component includes CMS, web hosting service, testimonial page, email campaigns, etc.	The web server is a software or a system which maintain the web applications, generate response and accept clients data.
Its main function is to provide essential and helpful information to users or visitors and turn them into loyal customers.	Components of web server architecture- hardware, operating system software, and Web server software.
SIt uses different web pages to display information.	The web server gets HTTP requests and sends HTTP responses.
It provides good web navigation and make it easy for the user to have access to	There exist three types of processing models for web servers i.e Process-based, Thread based, and Hybrid.

information that in turn increase user experience.	
It is more difficult to create and maintain website as compared to web browser.	Web servers provide an area to store and organize the pages of the website.
Websites are mostly created using HTML, CSS and javascript.	The web server can be a remote machine placed on the other side of your network or even on the other end of the globe, or it is your very own personal computer at home.
Its type includes E-commerce Websites, Landing Page Websites, Magazine Websites, etc.	An example of a Web Server is Apache Server.

Difference between Internet and Intranet

INTERNET	**INTRANET**
Simultaneously link computers on different network / global network	Owned by local or private organisations / companies
Support multiple users	Users are limited
Unsafe, not protected	Protected and secured
It’s a public network with more traffic	A private network and traffic is less
Can transfer unlimited data	Can transfer only limited data
Can be widely accessed and used	Company or organisation employees or admin with access to login details can only use this
More data or information can be accessed or availed	Data or information accessible over intranet will be limited and specific to the company records or details

HTML 5

HTML is the main markup language for describing the structure of web pages.

HTML stands for HyperText Markup Language. HTML is the basic building block of World Wide Web.

Hypertext is text displayed on a computer or other electronic device with references to other text that theuser can immediately access, usually by a mouse click or key press.

Apart from text, hypertext may contain tables, lists, forms, images, and other presentational elements. It isan easy-to-use and flexible format to share information over the Internet.

Markup languages use sets of markup tags to characterize text elements within a document, which givesinstructions to the web browsers on how the document should appear.

HTML Tags and Elements

HTML is written in the form of HTML elements consisting of markup tags. These markup tags are the fundamental characteristic of HTML. Every markup tag is composed of a keyword, surroundedby angle brackets, such as <html>, <head>, <body>, <title>, <p>, and so on.

HTML tags normally come in pairs like <html> and </html>. The first tag in a pair is often called theopening tag (or start tag), and the second tag is called the closing tag (or end tag).

An opening tag and a closing tag are identical, except a slash (/) after the opening angle bracket ofthe closing tag, to tell the browser that the command has been completed.

Inserting Images into Web Pages

Images enhance visual appearance of the web pages by making them more interesting and colorful.

The <img> tag is used to insert images in the HTML documents. It is an empty element and contains attributes only. The syntax of the <img> tag can be given with:

```
<img src="url" alt="some_text">
```

The following example inserts three images on the web page:

Example

Try this code »

```
<img src="kites.jpg" alt="Flying Kites">
<img src="sky.jpg" alt="Cloudy Sky">
<img src="balloons.jpg" alt="Balloons">
```

Each image must carry at least two attributes: the src attribute, and an alt attribute.

The src attribute tells the browser where to find the image. Its value is the URL of the image file.

Whereas, the alt attribute provides an alternative text for the image, if it is unavailable or cannot bedisplayed for some reason. Its value should be a meaningful substitute for the image.

HTML Tables

Creating Tables in HTML

HTML table allows you to arrange data into rows and columns. They are commonly used to displaytabular data like product listings, customer's details, financial reports, and so on.

You can create a table using the <table> element. Inside the <table> element, you can use the <tr> elements to create rows, and to create columns inside a row you can use the <td>

elements.You can also define a cell as a header for a group of table cells using the <th> element.

The following example demonstrates the most basic structure of a table.

```
<table>
   <tr>
      <th>No.</th>
      <th>Name</th>
      <th>Age</th>
   </tr>
   <tr>
      <td>1</td>
      <td>Peter Parker</td>
      <td>16</td>
   </tr>
   <tr>
      <td>2</td>
      <td>Clark Kent</td>
      <td>34</td>
      </tr>
      </table>
```

Tables do not have any borders by default. You can use the CSS border property to add borders tothe tables. Also, table cells are sized just large enough to fit the contents by default. To add more space around the content in the table cells you can use the CSS padding property.

Defining a Table Header, Body, and Footer

HTML provides a series of tags <thead>, <tbody>, and <tfoot> that helps you to create morestructured table, by defining header, body and footer regions, respectively.

The following example demonstrates the use of these elements.

Example

```
<table>
   <thead>
      <tr>
         <th>Items</th>
         <th>Expenditure</th>
      </tr>
      </thread>
      <tbody>
      <tr>
         <td>Stationary</td>
         <td>2,000</td>
      </tr>
      <tr>
         <td>Furniture</td>
         <td>10,000</td>
         </tr>
```

```
        </tbody>
        <tfoot>
        <tr>
        <th>Total</th>
        <td>12,000</t
    </tfoot>
</table>
```

HTML Lists

HTML lists are used to present list of information in well formed and semantic way. There are threedifferent types of list in HTML and each one has a specific purpose and meaning.

- **Unordered list** — Used to create a list of related items, in no particular order.
- **Ordered list** — Used to create a list of related items, in a specific order.
- **Description list** — Used to create a list of terms and their descriptions.

HTML Unordered Lists

An unordered list created using the <ul> element, and each list item starts with the <li> element.The list items in unordered lists are marked with bullets. Here's an example:

Example,

```
<ul>
    <li>Chocolate Cake</li>
    <li>Black Forest Cake</li>
    <li>Pineapple Cake</li>
</ul>
```

— The output of the above example will look something like this:

- Chocolate Cake
- Black Forest Cake
- Pineapple Cake

You can also change the bullet type in your unordered list using the CSS list-style-type property. Thefollowing style rule changes the type of bullet from the default *disc* to *square*:
Example,

```
ul {
    list-style-type: square;
}
```

Please check out the tutorial on CSS lists to learn about styling HTML lists in details.

HTML Ordered Lists

An ordered list created using the <ol> element, and each list item starts with the <li> element.Ordered lists are used when the order of the list's items is important.

The list items in an ordered list are marked with numbers. Here's an example:

```
    <li>Fasten your seatbelt</li>
    <li>Starts the car's engine</li>
    <li>Look around and go</li>
</ol>
```

— The output of the above example will look something like this:

1. Fasten your seatbelt
2. Starts the car's engine
3. Look around and go

HTML5 Image

HTML Images Syntax

In HTML, images are defined with the `<img>` tag.

The `<img>` tag is empty, it contains attributes only, and does not have a closing tag.

The `src` attribute specifies the URL (web address) of the image:

```
<img src="url">
```

EXAMPLE

```
<!DOCTYPE html>

<html>

<body>



<h2>HTML Image</h2>

<img src="html5.gif" alt="HTML5 Icon" style="width:128px;height:128px;">



</body>

</body>

</html>
```

HTML Form

HTML Forms are required to collect different kinds of user inputs, such as contact details likename, email address, phone numbers, or details like credit card information, etc.

Forms contain special elements called **controls** like ***input box, check boxes, radio-buttons, submit buttons***, etc. Users generally complete a form by modifying its controls e.g. entering text, selectingitems, etc. and submitting this form to a web server for further processing.

The <form> tag is used to create an HTML form. Here's a simple example of a login form:

```
<form>
   <label>Username: <input type="text"></label>
   <label>Password: <input type="password"></label>
   <input type="submit" value="Submit">
</form>
```

The following section describes different types of controls that you can use in your form.

Input Element

This is the most commonly used element within HTML forms.

It allows you to specify various types of user input fields, depending on the type attribute. An inputelement can be of type *text field*, *password field*, *checkbox*, *radio button*, *submit button*, *reset button*, *file select box*, as well as several new input types introduced in HTML5.

The most frequently used input types are described below.

Text Fields

Text fields are one line areas that allow the user to input text.

Single-line text input controls are created using an <input> element, whose type attribute has a valueof text. Here's an example of a single-line text input used to take username:

```
<form>
   <label for="username">Username:</label>
   <input type="text" name="username" id="username">
</form>
```

Password Field

Password fields are similar to text fields. The only difference is; characters in a password field aremasked, i.e. they are shown as asterisks or dots. This is to prevent someone else from reading the password on the screen. This is also a single-line text input controls created using an <input> element whose type attribute has a value of password.

```
<form>
   <label for="user-pwd">Password:</label>
   <input type="password" name="user-password" id="user-pwd">
</form>
```

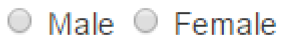
— The output of the above example will look something like this:

Radio Buttons

Radio buttons are used to let the user select exactly one option from a pre-defined set of options. Itis created using an <input> element whose type attribute has a value of radio. For example,

```
<form>
   <input type="radio" name="gender" id="male">
   <label for="male">Male</label>
   <input type="radio" name="gender" id="female">
   <label for="female">Female</label>
</form>
```

— The output of the above example will look something like this:

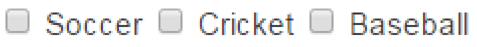


Checkboxes

Checkboxes allows the user to select one or more option from a pre-defined set of options. It is created using an <input> element whose type attribute has a value of checkbox. For example,

```
<form>
   <input type="checkbox" name="sports" id="soccer">
   <label for="soccer">Soccer</label>
   <input type="checkbox" name="sports" id="baseball">
   <label for="baseball">Baseball</label>
</form>
```

— The output of the above example will look something like this:



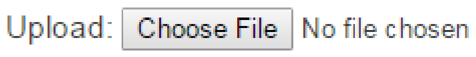
File Select box

The file fields allow a user to browse for a local file and send it as an attachment with the form data.Web browsers such as Google Chrome and Firefox render a file select input field with a Browse button that enables the user to navigate the local hard drive and select a file.

This is also created using an <input> element, whose type attribute value is set to file. For Example,

```
<form>
   <label for="file-select">Upload:</label>
   <input type="file" name="upload" id="file-select">
</form>
```

— The output of the above example will look something like this:



Text area

Text area is a multiple-line text input control that allows a user to enter more than one line of text.Multi-line text input controls are created using an <textarea> element. For example,

```
<form>
   <label for="address">Address:</label>
   <textarea rows="3" cols="30" name="address" id="address"></textarea>
</form>
```

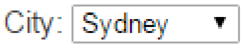
— The output of the above example will look something like this:

Select Boxes

A select box is a dropdown list of options that allows user to select one or more option from a pull-down list of options. Select box is created using the <select> element and <option> element.

```
<form>

    <label for="city">City:</label>
    <select name="city" id="city">
        <option value="sydney">Sydney</option>
        <option value="melbourne">Melbourne</option>
        <option value="cromwell">Cromwell</option>
    </select>
</form>
```

— The output of the above example will look something like this:



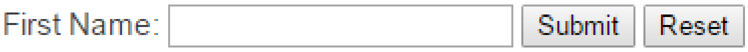
Submit and Reset Buttons

A submit button is used to send the form data to a web server. When submit button is clicked theform data is sent to the file specified in the form's action attribute to process the submitted data.

A reset button resets all the forms control to default values. Try out the following example by typingyour name in the text field, and click on submit button to see it in action.

```
<form action="action.php" method="post">
    <label for="first-name">First Name:</label>
    <input type="text" name="first-name" id="first-name">
 <input type="submit"
```

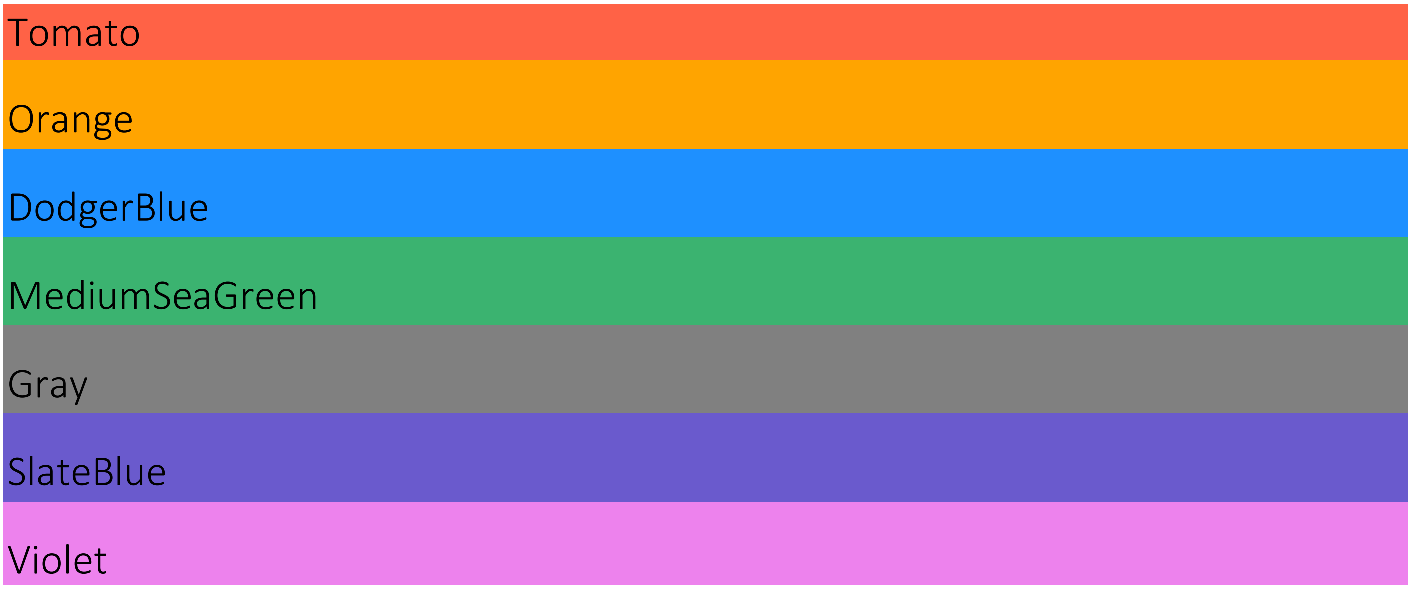
```
    <
    value="Submit">
    <input type="reset" value="Reset">
</form>
```



HTML5 Colors

```
<!DOCTYPE html>

<html>

<body>


<h1 style="background-color:Tomato;">Tomato</h1>

<h1 style="background-color:Orange;">Orange</h1>

<h1 style= "background
color:DodgerBlue;">DodgerBlue</h1>

 <h1 style="background-color:Gray;">Gray</h1>

<h1 style="background-color:SlateBlue;">SlateBlue</h1>

<h1 style="background-color:Violet;">Violet</h1>

<h1 style="background-color:LightGray;">LightGray</h1>
</body>

</html>
```

OUTPUT



```
Tomato
Orange
DodgerBlue
MediumSeaGreen
Gray
SlateBlue
Violet
```

HTML5 Audio

Embedding Audio in HTML Document

Inserting audio onto a web page was not easy before, because web browsers did not have a uniformstandard for defining embedded media files like audio.

Using the HTML5 audio Element

The newly introduced HTML5 `<audio>` element provides a standard way to embed audio in webpages. However, the audio element is relatively new but it works in most of the modern web browsers.

The following example simply inserts an audio into the HTML5 document, using the browserdefault set of controls, with one source defined by the `src` attribute.

Example

```
<audio controls="controls" src="media/birds.mp3">
   Your browser does not support the HTML5 Audio element.
</audio>
```

An audio, using the browser default set of controls, with alternative sources.

```
<audio controls="controls">
   <source src="media/birds.mp3" type="audio/mpeg">
   <source src="media/birds.ogg" type="audio/ogg">
   Your browser does not support the HTML5 Audio element.
</audio>
```

HTML5 Video

Embedding Video in HTML Document

Inserting video onto a web page was not relatively easy, because web browsers did not have auniform standard for defining embedded media files like video.

Using the HTML5 video Element

The newly introduced HTML5 `<video>` element provides a standard way to embed video in webpages. However, the video element is relatively new, but it works in most of the modern web browsers.

The following example simply inserts a video into the HTML document, using the browser defaultset of controls, with one source defined by the `src` attribute.

```
<video controls="controls" src="media/shuttle.mp4">
```

```
  Your browser does not support the HTML5 Video element.
</video>
```

A video, using the browser default set of controls, with alternative sources.

```
<video controls="controls">
  <source src="media/shuttle.mp4" type="video/mp4">
  <source src="media/shuttle.ogv" type="video/ogg">
  Your browser does not support the HTML5 Video element.
</video>
```

New HTML5 Elements

The most interesting new HTML5 elements are:

New **semantic elements** like <header>, <footer>, <article>, and <section>.

New **attributes of form elements** like number, date, time, calendar, and range.

New **graphic elements**: <svg> and <canvas>.

New **multimedia elements**: <audio> and <video>.

What are Semantic Elements?

A semantic element clearly describes its meaning to both the browser and the developer.

Examples of **non-semantic** elements: <div> and <span> - Tells nothing about its content.

Examples of **semantic** elements: <form>, <table>, and <article> - Clearly defines its content.

New Semantic Elements in HTML5

Many web sites contain HTML code like:

<div id="nav"> <div class="header"> <div id="footer">
to indicate navigation, header, and footer.

HTML5 offers new semantic elements to define different parts of a web page:

- <article>
- <aside>
- <details>
- <figcaption>
- <figure>
- <footer>
- <header>
- <main>
- <mark>

- <nav>
- <section>
- <summary>
- <time>

HTML5 <section> Element

The <section> element defines a section in a document.

According to W3C's HTML5 documentation: "A section is a thematic grouping of content, typically with a heading."

A home page could normally be split into sections for introduction, content, and contact information.

```
<section>
  <h1>WWF</h1>
  <p>The World Wide Fund for Nature (WWF) is... </p>
</section>
```

HTML5 <article> Element

The <article> element specifies independent, self-contained content.

An article should make sense on its own, and it should be possible to read it independently from the rest of the web site.

Examples of where an <article> element can be used:

- Forum post
- Blog post
- Newspaper article

The <header> element should be used as a container for introductory content.

You can have several <header> elements in one document.

The following example defines a header for an article:

```
<article>
  <header>
    <h1>What Does WWF Do?</h1>
    <p>WWF's mission:</p>
  </header>
  <p>WWF's mission is to stop the degradation of our planet's natural environment,
  and build a future in which humans live in harmony with nature.</p>
</article>
```

HTML5 <footer> Element

The <footer> element specifies a footer for a document or section.

A <footer> element should contain information about its containing element.

A footer typically contains the author of the document, copyright information, links to terms of use, contact information, etc.

You may have several <footer> elements in one document.

```
<footer>
  <p>Posted by: Hege Refsnes</p>
  <p>Contact information: <a href="mailto:someone@example.com">
  someone@example.com</a>.</p>
</footer>
```

HTML5 <figure> and <fig caption> Elements

The purpose of a figure caption is to add a visual explanation to an image.

In HTML5, an image and a caption can be grouped together in a <figure> element:

Example

```
<figure>
  <img src="pic_trulli.jpg" alt="Trulli">
  <fig caption>Fig1. - Trulli, Puglia, Italy.</fig caption>
</figure>
```

Semantic Elements in HTML5

Below is an alphabetical list of the new semantic elements in HTML5.

The links go to our complete HTML5 Reference.

Tag	Description
<article>	Defines an article
<aside>	Defines content aside from the page content
<details>	Defines additional details that the user can view or hide
<fig caption>	Defines a caption for a <figure> element
<figure>	Specifies self-contained content, like illustrations, diagrams, photos, code listings, etc.

<footer>	Defines a footer for a document or section
<header>	Specifies a header for a document or section
<main>	Specifies the main content of a document
<mark>	Defines marked/highlighted text
<nav>	Defines navigation links
<section>	Defines a section in a document
<summary>	Defines a visible heading for a <details> element
<time>	Defines a date/time

What Is CSS3 And Why Is It Used?

To help build highly interactive online pages, CSS3 is invariably used due to its importance in providing greater options in the design process. When marketing products and services, web design plays a vital part; a site should be created in a manner that will draw potential customers to explore and revisit a website more often. Many **web design firm**s are developing and enhancing websites through the use of CSS3 as this is a great form of web development. This article will help define CSS3 and will point out its advantages.

Definition

The acronym CSS stands for Cascading Style Sheets which is used to augment the functionality, versatility. and efficient performance of site content. It allows for the creation of content-rich websites that do not require much weight or codes; this translates into more interactive graphics and animation, superior user-interface, and significantly more organization and rapid download time.

It is used with HTML to create content structure, with CSS3 being used to format structured content. It is responsible for font properties, colors, text alignments, graphics, background images, tables and other components. This tool provides extra capabilities such as absolute, fixed and relative positioning of various elements. The increasing popularity of CSS3 when used by **web design firm**s stimulates major browsers such as Google Chrome, Firefox, Safari, and IE9 to adopt and embrace this programming language.

Advantages

Although CSS3 is not the only web development solution, it does allow provide greater advantages for several reasons.

- **Customization** – A web page can be customized and alterations created in the design by simply changing a modular file.
- **Bandwidth Requirements** – It decreases server bandwidth requirements, giving rapid download time when a site is accessed with desktop or hand-held devices, providing an improved user experience.
- **Consistency** – It delivers consistent and accurate positioning of navigational elements on the website.
- **Appealing** – It makes the site more appealing with adding videos and graphics easier.
- **Viewing** – It allows online videos to be viewed without the use of third-party plug-ins.
- **Visibility** – It delivers the opportunity to improve brand visibility by designing effective online pages.
- **Cost Effective** – It is cost-effective, time-saving, and supported by most browsers.

Since the introduction of CSS3, there is greater control of the presentation of content and various elements on a website; however it is not really responsible for overall design as it only specifies the structure and content presentation of certain web pages.

Cascading Style Sheets (CSS) are files with styling rules that govern how your website is presented on screen. CSS rules can be applied to your website's HTML files in various ways. You can use an **external stylesheet**, an **internal stylesheet**, or an **inline style**. Each method has advantages that suit particular uses.

An **external stylesheet** is a standalone .css file that is linked from a web page. The advantage of external stylesheets is that it can be created once and the rules applied to multiple web pages. Should you need to make widespread changes to your site design, you can make a single change in the stylesheet and it will be applied to all linked pages, saving time and effort.

An **internal stylesheet** holds CSS rules for the page in the **head** section of the HTML file. The rules only apply to that page, but you can configure CSS classes and IDs that can be used to style multiple elements in the page code. Again, a single change to the CSS rule will apply to all tagged elements on the page.

Inline styles relate to a specific HTML tag, using a **style** attribute with a CSS rule to style a specific page element. They're useful for quick, permanent changes, but are less flexible than external and internal stylesheets as each inline style you create must be separately edited should you decide to make a design change.

Using external CSS stylesheets

An HTML page styled by an external CSS stylesheet must reference the .css file in the document head. Once created, the CSS file must be uploaded to your server and linked in the HTML file with code such as:

```
<link href="style.css" rel="stylesheet" type="text/css">
```

You can name your stylesheet whatever you wish, but it should have a .css file extension.

Using internal CSS stylesheets

Rather than linking an external .css file, HTML files using an internal stylesheet include a set of rules in their **head** section. CSS rules are wrapped in <style> tags, like this:

```
<head>
<style type="text/css">
```

```
h1 {
        color:#fff
        margin-left: 20px;
   }
```

```
    p {
        font-family: Arial, Helvetica, Sans Serif;
      }



</style>
</head>
```

Using inline styles

Inline styles are applied directly to an element in your HTML code. They use the style attribute, followed by regular CSS properties.

For example:

```
<h1 style="color:red;margin-left:20px;">Today’s Update</h1>
```

Rule Cascading

Cascade and inheritance

Conflicting rules

CSS stands for Cascading Style Sheets, and that first word *cascading* is incredibly important to understand — the way that the cascade behaves is key to understanding CSS.
At some point, we will find that the CSS have created two rules which could potentially apply to the same element. The cascade, and the closely-related concept of specificity, are mechanisms that control which rule applies when there is such a conflict. Which rule is styling your element may not be the one you expect, so you need to understand how these mechanisms work.
Also significant here is the concept of inheritance, which means that some CSS properties by default inherit values set on the current element's parent element, and some don't. This can also cause some behavior that you might not expect.

CASCADE:

Stylesheets cascade — at a very simple level this means that the order of CSS rules matter; when two rules apply that have equal specificity the one that comes last in the CSS is the one that will be used.

EXAMPLE
In the below example, we have two rules that could apply to the `h1`. The `h1` ends up being colored blue — these rules have an identical selector and therefore carry the same specificity, so the last one in the source order wins.

```
h1 {

    color: red;


 h1 {
    color: blue;

}

    <h1>This is my heading.</h1>
```

Inheritance

Inheritance also needs to be understood in this context — some CSS property values set on parent elements are inherited by their child elements, and some aren't.

For example, if you set a color and font-family on an element, every element inside it will also be styled with that color and font, unless you've applied different color and font values directly to them.

Some properties do not inherit — for example if you set a width of 50% on an element, all of its descendants do not get a width of 50% of their parent's width. If this was the case, CSS would be very frustrating to use!

```
body {
   color: blue;
}

span {
   color: black;
}
```

```
 p>As the body has been set to have a color of blue this is inherited through thedescendants.</p>

<p>We can change the color by targetting the element with a selector,such as this

<span>span</span>.</p>
```

With CSS3 you can create two types of shadows: text-shadow (adds shadow to text) and box-shadow (adds shadow to other elements).

CSS3 Text Shadow

The text-shadow property can take up to four values:

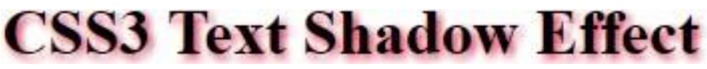
- the horizontal shadow
- the vertical shadow
- the blur effect
- the color

Examples:

- Normal text shadow

```
• h1 {
• text-shadow: 2px 2px 5px crimson;
```

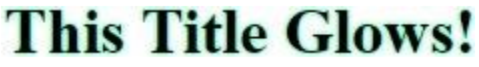
```
}
```


CSS3 Text Shadow Effect

- Glowing text effect

```
• h1 {
• text-shadow: 0 0 4px #00FF9C;
```

```
}
```


This Title Glows!

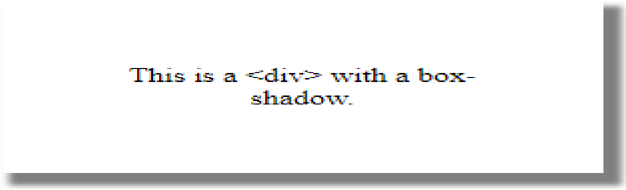
CSS3 Box Shadow

The box-shadow property can take up to six values:

- the horizontal shadow
- the vertical shadow
- the blur effect
- the spreading
- the color

Examples:

```
.first-div {
        box-shadow: 1px 1px 5px 3px grey;
}
```



This is a <div> with a box-shadow.

CSS Animations

CSS allows animation of HTML elements without using JavaScript or Flash!

What are CSS Animations?

An animation lets an element gradually change from one style to another.

You can change as many CSS properties you want, as many times you want.

To use CSS animation, you must first specify some keyframes for the animation.

Keyframes hold what styles the element will have at certain times.

The @keyframes Rule

When you specify CSS styles inside the @keyframes rule, the animation will gradually change from the current style to the new style at certain times.

To get an animation to work, you must bind the animation to an element.

The following example binds the "example" animation to the <div> element. The animation will last for 4 seconds, and it will gradually change the background-color of the <div> element from "red" to "yellow":

Example

```
/* The animation code */
@keyframes example
```

```
  to {background-color: yellow;}
}

/* The element to apply the animation to */
div {
  width: 100px;
  height: 100px;
  background-color: red;
  animation-name: example;
  animation-duration: 4s;
}
```

Note: The animation-duration property defines how long time an animation should take to complete. If the animation-duration property is not specified, no animation will occur, because the default value is 0s (0 seconds).

In the example above we have specified when the style will change by using the keywords "from" and "to" (which represents 0% (start) and 100% (complete)).

It is also possible to use percent. By using percent, you can add as many style changes as you like.

The following example will change the background-color of the <div> element when the animation is 25% complete, 50% complete, and again when the animation is 100% complete:

Example

```
/* The animation code */
@keyframes example {
  0%   {background-color: red;}
  25%  {background-color: yellow;}
  50%  {background-color: blue;}
  100% {background-color: green;}
}

/* The element to apply the animation to */
div {
  width: 100px;
  height: 100px;
  background-color: red;
  animation-name: example;
  animation-duration: 4s;
}
```

CSS Animation Properties

The following table lists the @keyframes rule and all the CSS animation properties:

Property	Description
@keyframes	Specifies the animation code
animation	A shorthand property for setting all the animation properties

animation-direction	Specifies whether an animation should be played forwards, backwards or in alternate cycles
animation-duration	Specifies how long time an animation should take to complete one cycle
animation-fill-mode	Specifies a style for the element when the animation is not playing (before it starts, after it ends, or both)
animation-iteration-count	Specifies the number of times an animation should be played
animation-name	Specifies the name of the @keyframes animation
animation-play-state	Specifies whether the animation is running or paused
animation-timing-function	Specifies the speed curve of the animation

CSS Transitions

CSS Transitions is a module of CSS that lets you create gradual transitions between the values of specific CSS properties. The behavior of these transitions can be controlled by specifying their timing function, duration, and otherattributes.

Properties

- transition
- transition-delay
- transition-duration
- transition-property
- transition-timing-function

The **transition** CSS property is a shorthand property for transition-property, transition-duration, transition-timing-function,and transition-delay.

CSS transition Property

Example

Hover over a <div> element to gradually change the width from 100px to 300px:

```
div {
  width: 100px;
  transition: width 2s;
}

div:hover {
  width: 300px;
}
```

Definition and Usage

The transition property is a shorthand property for:

- transition-property
- transition-duration
- transition-timing-function

Property Values

Value	Description
transition-property	Specifies the name of the CSS property the transition effect is for
transition-duration	Specifies how many seconds or milliseconds the transition effect takes to complete
transition-timing-function	Specifies the speed curve of the transition effect
transition-delay	Defines when the transition effect will start
initial	Sets this property to its default value. Read about *initial*
inherit	Inherits this property from its parent element. Read about *inherit*

Example

When an <input type="text"> gets focus, gradually change the width from

100px to 250px:

```
input[type=text] {
  width: 100px;
  transition: width .35s ease-in-out;
}

input[type=text]:focus {
}
```

OUTPUT

The width Property

Set the width of the input field to 100 pixels. However, when the input field gets focus, make it250 pixels wide:

Search:

CSS background-color

The background-color property specifies the background color of an element.

Example

The background color of a page is set like this:

```
body {
   background-color: lightblue;
}
```

CSS background-image

The background-image property specifies an image to use as the background of an element.By default, the

image is repeated so it covers the entire element.

Example

The background image for a page can be set like this:

```
body {
   background-image: url("paper.gif");
}
```

CSS background - Shorthand property

To shorten the code, it is also possible to specify all the background properties in onesingle property. This is called a shorthand property.

The shorthand property for background is background.

Use the shorthand property to set all the background properties in one declaration:

```
body {
  background: #ffffff url("img_tree.png") no-repeat right top;
}
```

CSS Border - Shorthand Property

As you can see from the examples above, there are many properties to consider whendealing with borders.

To shorten the code, it is also possible to specify all the individual border properties inone property.

The border property is a shorthand property for the following individual border properties:

- border-width
- border-style (required)
- border-color

Example

```
p {
  border: 5px solid red;
}
```

Result:

Some text