

# **Unit-I**

**WEBSITE BASICS, HTML 5, CSS 3**

# Contents

Web Essentials: Clients, Servers and Communication – The Internet – Basic Internet protocols – World wide web – HTTP Request Message – HTTP Response Message

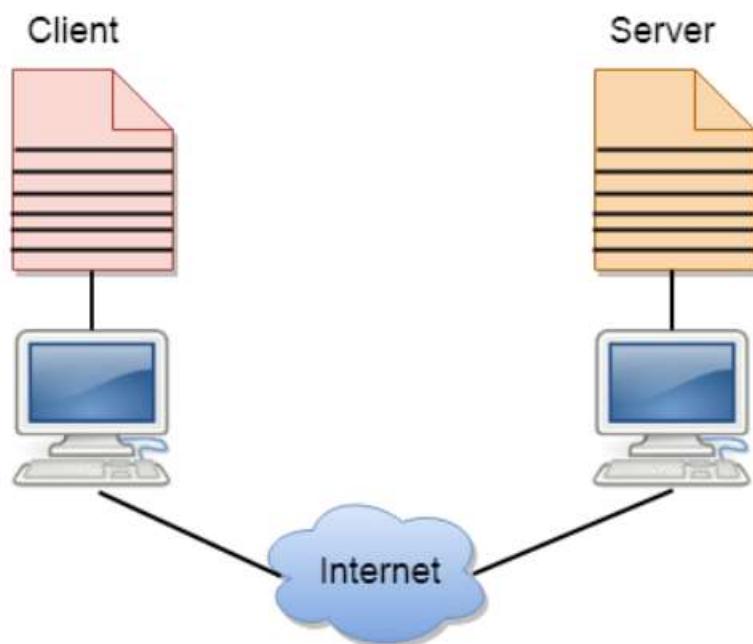
HTML5 – Basic HTML Elements–Tables – Lists – Image – HTML5 control elements –Audio – Video controls

CSS3 – Inline, embedded and external style sheets– Backgrounds – Border Images – Colors – Shadows – Text – Transformations – Transitions – Animations

# Clients, Servers and Communication

## Client and Server model

A client and server networking model is a model in which computers such **as servers provide the network services to the other computers such as clients to perform a user based tasks**. This model is known as client-server networking model.



### Client

A client is a program that runs on the local machine requesting service from the server. A client program is a finite program means that the service started by the user and terminates when the service is completed.

### Server

- A server is a program that runs on the remote machine providing services to the clients.
- When the client requests for a service, then the server opens the door for the incoming requests, but it never initiates the service.
- A server program is an infinite program means that when it starts, it runs infinitely unless the problem arises.

## Advantages of Client-server networks

- **Centralized:** Centralized back-up is possible in client-server networks, i.e., all the data is stored in a server.
- **Security:** These networks are more secure as all the shared resources are centrally administered.
- **Performance:** The use of the dedicated server increases the speed of sharing resources. This increases the performance of the overall system.
- **Scalability:** We can increase the number of clients and servers separately, i.e., the new element can be added, or we can add a new node in a network at any time.

## Disadvantages of Client-Server network

- **Traffic Congestion** is a big problem in Client/Server networks. When a large number of clients send requests to the same server may cause the problem of Traffic congestion.
- It **does not have a robustness** of a network, i.e., when the server is down, then the client requests cannot be met.
- A client/server network is **very decisive**. Sometimes, regular computer hardware does not serve a certain number of clients. In such situations, specific hardware is required at the server side to complete the work.
- Sometimes the resources exist in the server but may not exist in the client. For example, If the application is web, then we cannot take the print out directly on printers without taking out the print view window on the web.

# INTERNET

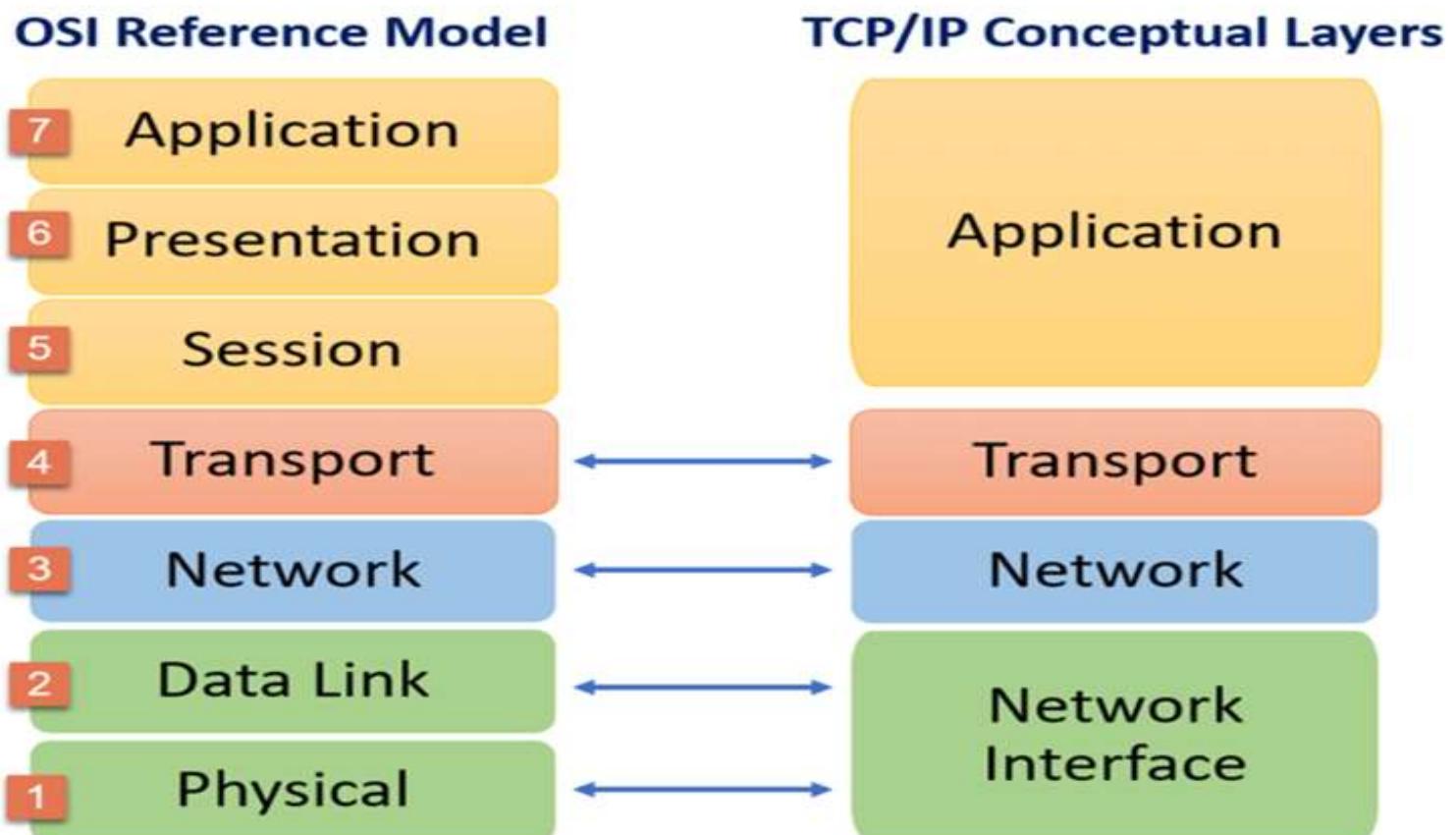
- Global network of computing resources
- Global system of interconnected computer networks that use the Internet protocol suite (TCP/IP) to link devices worldwide.

Some of the basic services available to Internet users are

- **Email** – A fast, easy, and inexpensive way to communicate with other Internet users around the world.
- **Telnet** – Allows a user to log into a remote computer as though it were a local system.
- **FTP** – Allows a user to transfer virtually every kind of file that can be stored on a computer from one Internet-connected computer to another.
- **UseNet news** – A distributed bulletin board that offers a combination news and discussion service on thousands of topics.
- **World Wide Web (WWW)** – A hypertext interface to Internet information resources



# Communication Models



# TCP/IP model

- The TCP/IP model was developed prior to the OSI model.
- The TCP/IP model consists of five layers: the application layer, transport layer, network layer, data link layer and physical layer.
- The first four layers provide physical standards, network interface, internetworking, and transport functions that correspond to the first four layers of the OSI model and these four layers are represented in TCP/IP model by a single layer called the application layer.
- TCP/IP is a hierarchical protocol made up of interactive modules, and each of them provides specific functionality.

# TCP/IP Protocol Suite

OSI Layers	TCP/IP Layers	TCP/IP Protocols				
Application Layer	Application Layer	HTTP	FTP	Telnet	SMTP	DNS
Presentation Layer	Transport Layer	TCP		UDP		
Session Layer	Network Layer	IP				
Transport Layer	Network Interface Layer	Ethernet	Token Ring		Other Link-Layer Protocols	
Network Layer						
Data Link Layer						
Physical Layer						

## Network Access Layer

- A network layer is the lowest layer of the TCP/IP model.
- A network layer is the combination of the Physical layer and Data Link layer defined in the OSI reference model.
- It defines how the data should be sent physically through the network.
- This layer is mainly responsible for the transmission of the data between two devices on the same network.
- The functions carried out by this layer are encapsulating the IP datagram into frames transmitted by the network and mapping of IP addresses into physical addresses.
- The protocols used by this layer are ethernet, token ring, FDDI, X.25, frame relay.

## Internet Layer

- An internet layer is the second layer of the TCP/IP model.
- An internet layer is also known as the network layer.
- The main responsibility of the internet layer is to send the packets from any network, and they arrive at the destination irrespective of the route they take.

# Protocols Used in Internet layer

## IP Protocol

IP protocol is used in this layer, and it is the most significant part of the entire TCP/IP suite.

Following are the responsibilities of this protocol:

- **IP Addressing:** This protocol implements logical host addresses known as IP addresses. The IP addresses are used by the internet and higher layers to identify the device and to provide internetwork routing.
- **Host-to-host communication:** It determines the path through which the data is to be transmitted.
- **Data Encapsulation and Formatting:** An IP protocol accepts the data from the transport layer protocol. An IP protocol ensures that the data is sent and received securely, it encapsulates the data into message known as IP datagram.
- **Fragmentation and Reassembly:** The limit imposed on the size of the IP datagram by data link layer protocol is known as Maximum Transmission unit (MTU). If the size of IP datagram is greater than the MTU unit, then the IP protocol splits the datagram into smaller units so that they can travel over the local network. Fragmentation can be done by the sender or intermediate router. At the receiver side, all the fragments are reassembled to form an original message.
- **Routing:** When IP datagram is sent over the same local network such as LAN, MAN, WAN, it is known as direct delivery. When source and destination are on the distant network, then the IP datagram is sent indirectly. This can be accomplished by routing the IP datagram through various devices such as routers.

## ARP Protocol

- ARP stands for **Address Resolution Protocol**.
- ARP is a network layer protocol which is used to find the physical address from the IP address.
- **The two terms are mainly associated with the ARP Protocol:**
  - **ARP request:** When a sender wants to know the physical address of the device, it broadcasts the ARP request to the network.
  - **ARP reply:** Every device attached to the network will accept the ARP request and process the request, but only recipient recognize the IP address and sends back its physical address in the form of ARP reply. The recipient adds the physical address both to its cache memory and to the datagram header

## ICMP Protocol

- **ICMP** stands for Internet Control Message Protocol.
- It is a mechanism used by the hosts or routers to send notifications regarding datagram problems back to the sender.
- An ICMP protocol mainly uses two terms:
  - **ICMP Test:** ICMP Test is used to test whether the destination is reachable or not.
  - **ICMP Reply:** ICMP Reply is used to check whether the destination device is responding or not.
- The core responsibility of the ICMP protocol is to report the problems, not correct them. The responsibility of the correction lies with the sender.
- ICMP can send the messages only to the source, but not to the intermediate routers because the IP datagram carries the addresses of the source and destination but not of the router that it is passed to.

## Transport Layer

- The transport layer is responsible for the reliability, flow control, and correction of data which is being sent over the network.
- The two protocols used in the transport layer are **User Datagram protocol and Transmission control protocol**.

### User Datagram Protocol (UDP)

- It provides connectionless service and end-to-end delivery of transmission.
- It is an unreliable protocol as it discovers the errors but not specify the error.
- User Datagram Protocol discovers the error, and ICMP protocol reports the error to the sender that user datagram has been damaged.
- **UDP consists of the following fields:**

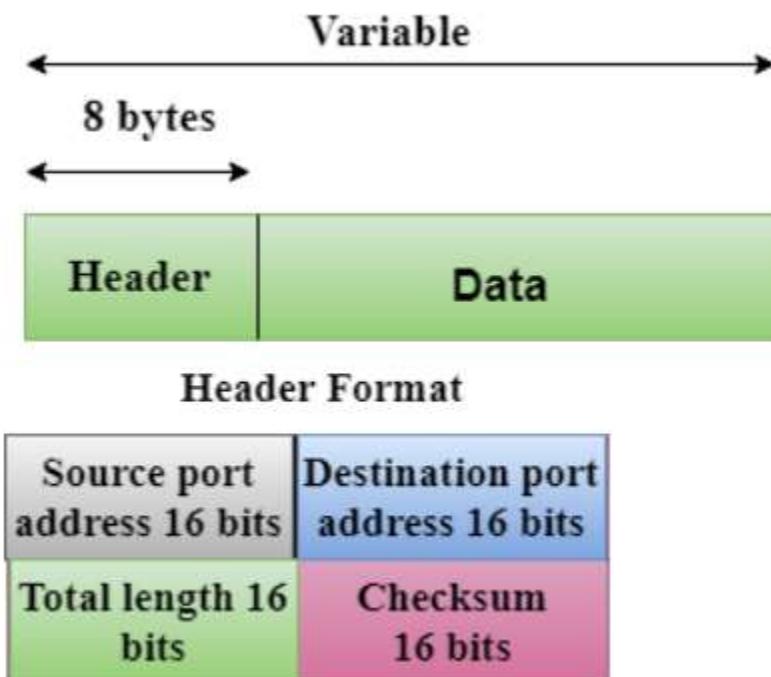
**Source port address:** The source port address is the address of the application program that has created the message.

**Destination port address:** The destination port address is the address of the application program that receives the message.

**Total length:** It defines the total number of bytes of the user datagram in bytes.

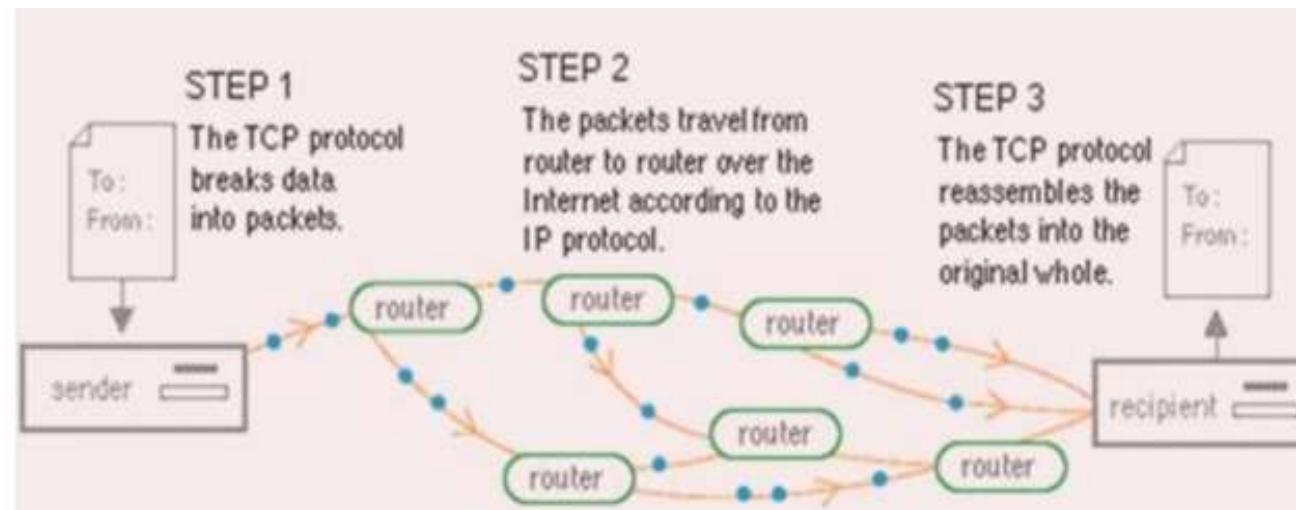
**Checksum:** The checksum is a 16-bit field used in error detection.

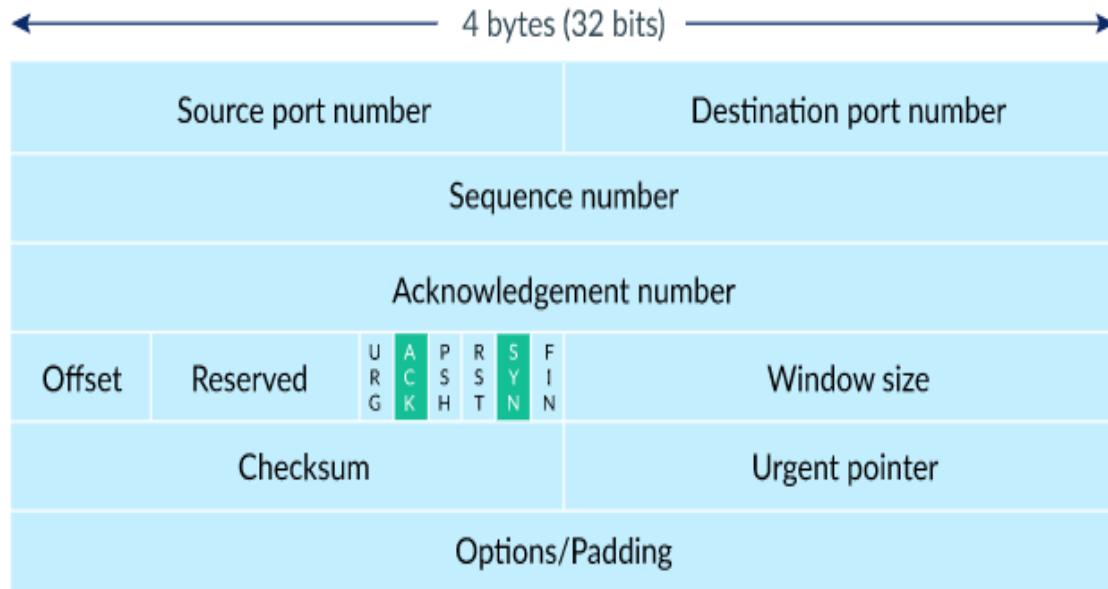
- UDP does not specify which packet is lost. UDP contains only checksum; it does not contain any ID of a data segment.



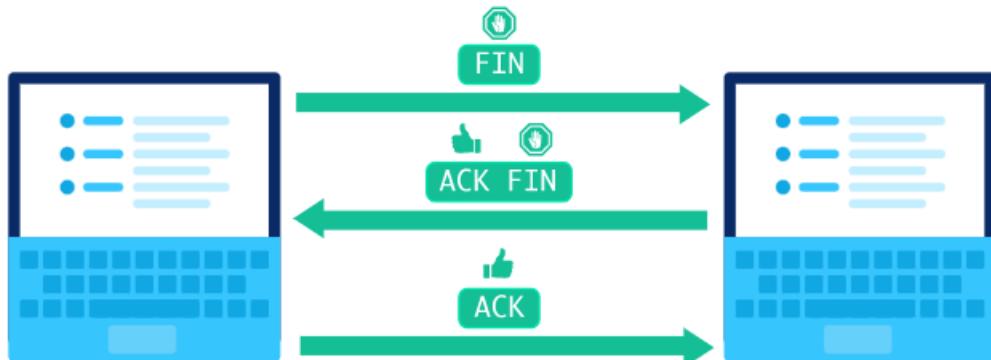
## Transmission Control Protocol (TCP)

- It provides a full transport layer services to applications.
- It creates a virtual circuit between the sender and receiver, and it is active for the duration of the transmission.
- TCP is a reliable protocol as it detects the error and retransmits the damaged frames. Therefore, it ensures all the segments must be received and acknowledged before the transmission is considered to be completed and a virtual circuit is discarded.
- At the sending end, TCP divides the whole message into smaller units known as segment, and each segment contains a sequence number which is required for reordering the frames to form an original message.
- At the receiving end, TCP collects all the segments and reorders them based on sequence numbers.

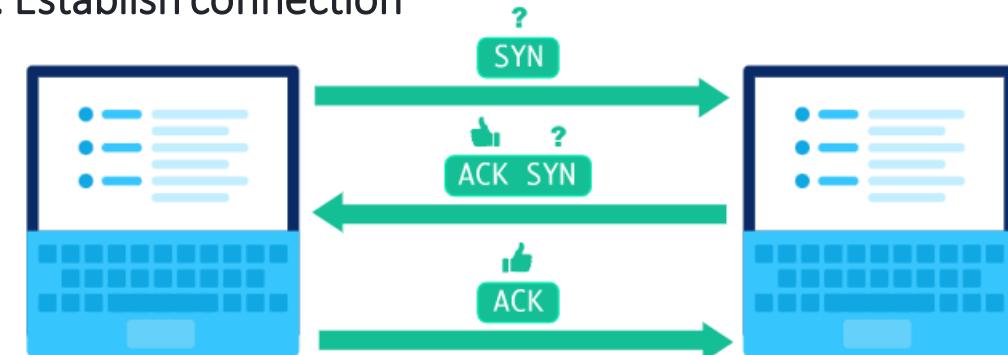




### Step 3: Close the connection



### Step 1: Establish connection



### Step 2: Send packets of data



## Application Layer

- An application layer is the topmost layer in the TCP/IP model.
- It is responsible for handling high-level protocols, issues of representation.
- This layer allows the user to interact with the application.
- When one application layer protocol wants to communicate with another application layer, it forwards its data to the transport layer.
- There is an ambiguity occurs in the application layer. Every application cannot be placed inside the application layer except those who interact with the communication system. For example: text editor cannot be considered in application layer while web browser using **HTTP** protocol to interact with the network where **HTTP** protocol is an application layer protocol.

Following are the main protocols used in the application layer:

- **HTTP:** HTTP stands for Hypertext transfer protocol. This protocol allows us to access the data over the world wide web. It transfers the data in the form of plain text, audio, video. It is known as a Hypertext transfer protocol as it has the efficiency to use in a hypertext environment where there are rapid jumps from one document to another.
- **SNMP:** SNMP stands for Simple Network Management Protocol. It is a framework used for managing the devices on the internet by using the TCP/IP protocol suite.
- **SMTP:** SMTP stands for Simple mail transfer protocol. The TCP/IP protocol that supports the e-mail is known as a Simple mail transfer protocol. This protocol is used to send the data to another e-mail address.
- **DNS:** DNS stands for Domain Name System. An IP address is used to identify the connection of a host to the internet uniquely. But, people prefer to use the names instead of addresses. Therefore, the system that maps the name to the address is known as Domain Name System.
- **TELNET:** It is an abbreviation for Terminal Network. It establishes the connection between the local computer and remote computer in such a way that the local terminal appears to be a terminal at the remote system.
- **FTP:** FTP stands for File Transfer Protocol. FTP is a standard internet protocol used for transmitting the files from one computer to another computer.

# World Wide Web

- World Wide Web, which is also known as a Web, is a collection of websites or web pages stored in web servers and connected to local computers through the internet. These websites contain text pages, digital images, audios, videos, etc.
- Users can access the content of these sites from any part of the world over the internet using their devices such as computers, laptops, cell phones, etc.
- The WWW, along with internet, enables the retrieval and display of text and media to your device.

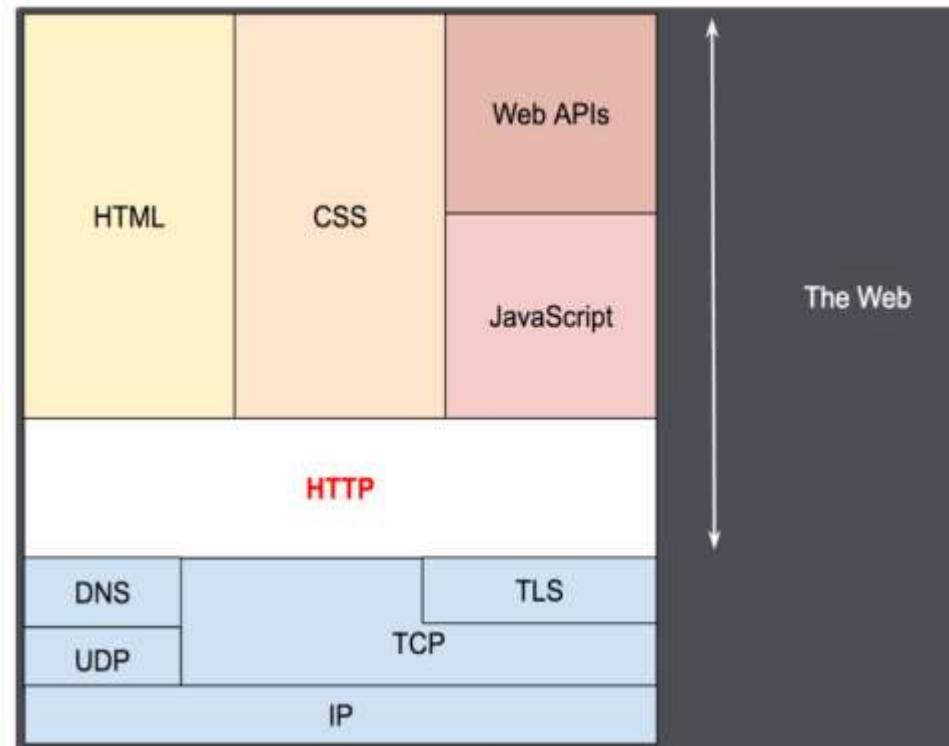
- The building blocks of the Web are web pages which are formatted in HTML and connected by links called "hypertext" or hyperlinks and accessed by HTTP.
- These links are electronic connections that link related pieces of information so that users can access the desired information quickly.
- Hypertext offers the advantage to select a word or phrase from text and thus to access other pages that provide additional information related to that word or phrase.

# HTTP

- HTTP stands for **HyperText Transfer Protocol**.
- It is a protocol used to access the data on the World Wide Web (www).
- The HTTP protocol can be used to transfer the data in the form of plain text, hypertext, audio, video, and so on.
- This protocol is known as HyperText Transfer Protocol because of its efficiency that allows us to use in a hypertext environment where there are rapid jumps from one document to another document.
- HTTP is similar to the FTP as it also transfers the files from one host to another host. But, HTTP is simpler than FTP as HTTP uses only one connection, i.e., no control connection to transfer the files.
- HTTP is used to carry the data in the form of MIME-like format.
- HTTP is similar to SMTP as the data is transferred between client and server. The HTTP differs from the SMTP in the way the messages are sent from the client to the server and from server to the client. SMTP messages are stored and forwarded while HTTP messages are delivered immediately.

## Basic Features

- HTTP is connectionless
- HTTP is media independent
- HTTP is stateless



## HTTP flow

- Open a TCP connection
- Send an HTTP message

```
GET / HTTP/1.1
Host: developer.mozilla.org
Accept-Language: fr
```

- Read the response sent by the server

```
HTTP/1.1 200 OK
Date: Sat, 09 Oct 2010 14:28:02 GMT
Server: Apache
Last-Modified: Tue, 01 Dec 2009 20:18:22 GMT
ETag: "51142bc1-7449-479b075b2891b"
Accept-Ranges: bytes
Content-Length: 29769
Content-Type: text/html
```

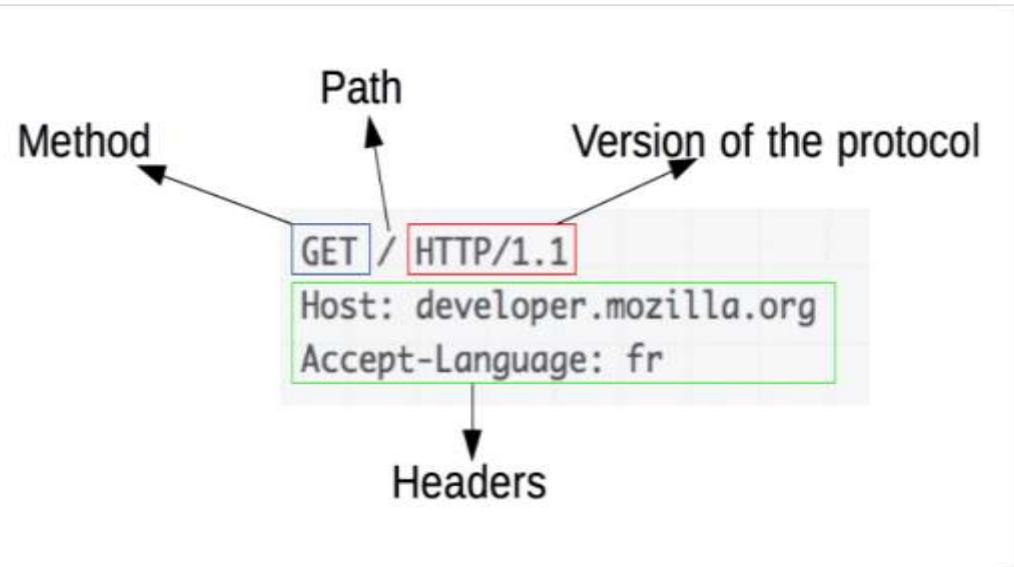
<!DOCTYPE html... (here come the 29769 bytes of the requested web page)

- Close or reuse the connection for further requests.

# HTTP Messages

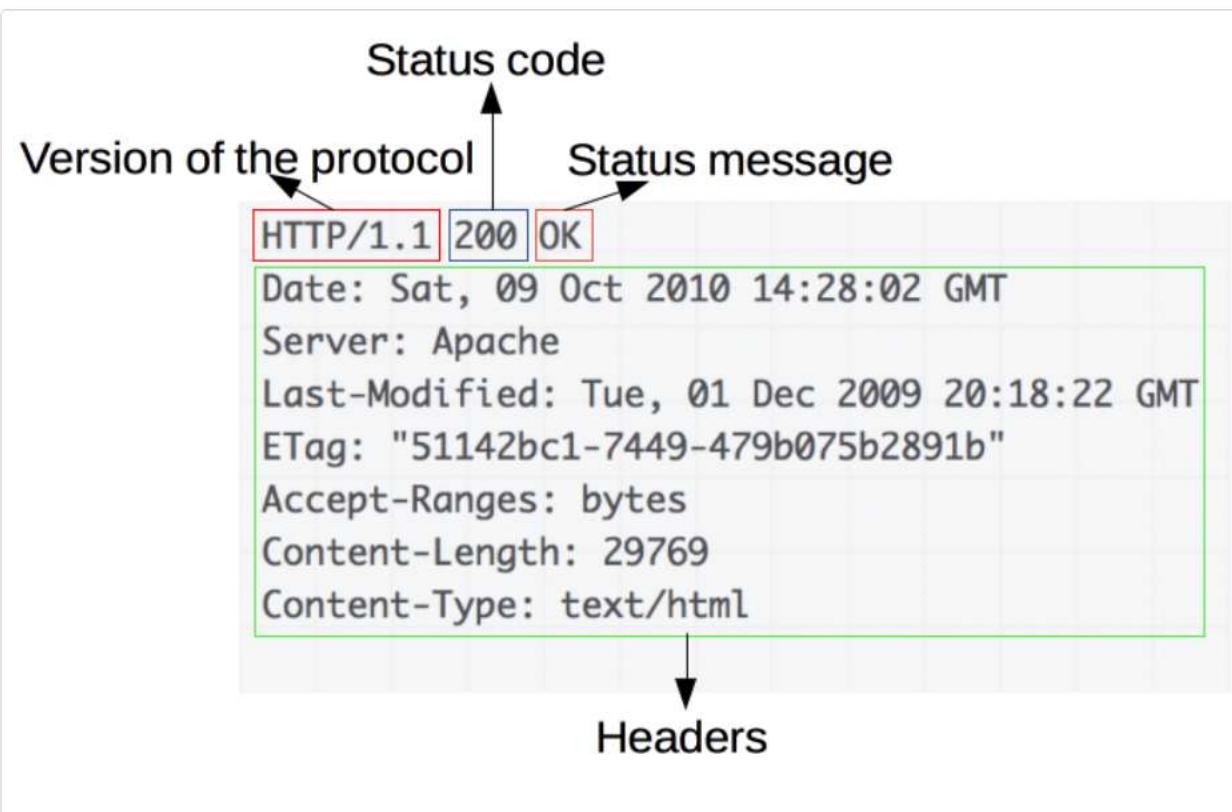
- HTTP messages, as defined in HTTP/1.1 and earlier, are human-readable.
- In HTTP/2, these messages are embedded into a binary structure, a *frame*, allowing optimizations like compression of headers and multiplexing.

# HTTP Requests



- An HTTP method, usually a verb like GET, POST, or a noun like OPTIONS or HEAD that defines the operation the client wants to perform.
- The path of the resource to fetch; the URL of the resource stripped from elements that are obvious from the context, for example without the protocol (`http://`), the domain (here, `developer.mozilla.org`), or the TCP port (here, 80).
- The version of the HTTP protocol.
- Optional headers that convey additional information for the servers

# HTTP Responses



- The version of the HTTP protocol they follow.
- A status code, indicating if the request was successful or not, and why.
- A status message, a non-authoritative short description of the status code.
- HTTP headers, like those for requests.
- Optionally, a body containing the fetched resource.

# HTML



# What is HTML?

- HTML stands for Hyper Text Markup Language
- HTML is the standard markup language for creating Web pages
- HTML describes the structure of a Web page
- HTML consists of a series of elements
- HTML elements tell the browser how to display the content

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>
<h1>My First Heading</h1>
<p>My first paragraph.</p>
</body>
</html>
```



# What is an HTML Element?

An HTML element is defined by a start tag, some content, and an end tag:

`<tagname> Content goes here... </tagname>`

The HTML **element** is everything from the start tag to the end tag:

`<h1>My First Heading</h1>  
<p>My first paragraph.</p>`

Start tag	Element content	End tag
<code>&lt;h1&gt;</code>	My First Heading	<code>&lt;/h1&gt;</code>
<code>&lt;p&gt;</code>	My first paragraph.	<code>&lt;/p&gt;</code>
<code>&lt;br&gt;</code>	<i>none</i>	<i>none</i>

# HTML Page Structure

```
<html>

    <head>

        <title>Page title</title>

    </head>

    <body>

        <h1>This is a heading</h1>

        <p>This is a paragraph.</p>

        <p>This is another paragraph.</p>

    </body>

</html>
```

# HTML History

Year	Version
1989	Tim Berners-Lee invented www
1991	Tim Berners-Lee invented HTML
1993	Dave Raggett drafted HTML+
1995	HTML Working Group defined HTML 2.0
1997	W3C Recommendation: HTML 3.2
1999	W3C Recommendation: HTML 4.01
2000	W3C Recommendation: XHTML 1.0
2008	WHATWG HTML5 First Public Draft
2012	<u>WHATWG HTML5 Living Standard</u>
2014	<u>W3C Recommendation: HTML5</u>
2016	W3C Candidate Recommendation: HTML 5.1
2017	<u>W3C Recommendation: HTML5.1 2nd Edition</u>
2017	<u>W3C Recommendation: HTML5.2</u>

## The <!DOCTYPE> Declaration

The <!DOCTYPE> declaration represents the document type, and helps browsers to display web pages correctly.

It must only appear once, at the top of the page (before any HTML tags).

The <!DOCTYPE> declaration is not case sensitive.

The <!DOCTYPE> declaration for HTML5 is: <!DOCTYPE html>

## HTML Headings

HTML headings are defined with the <h1> to <h6> tags.

<h1> defines the most important heading. <h6> defines the least important heading:

Example

```
<h1>This is heading 1</h1>
<h2>This is heading 2</h2>
<h3>This is heading 3</h3>
```

**This is heading 1**

**This is heading 2**

**This is heading 3**

**This is heading 4**

**This is heading 5**

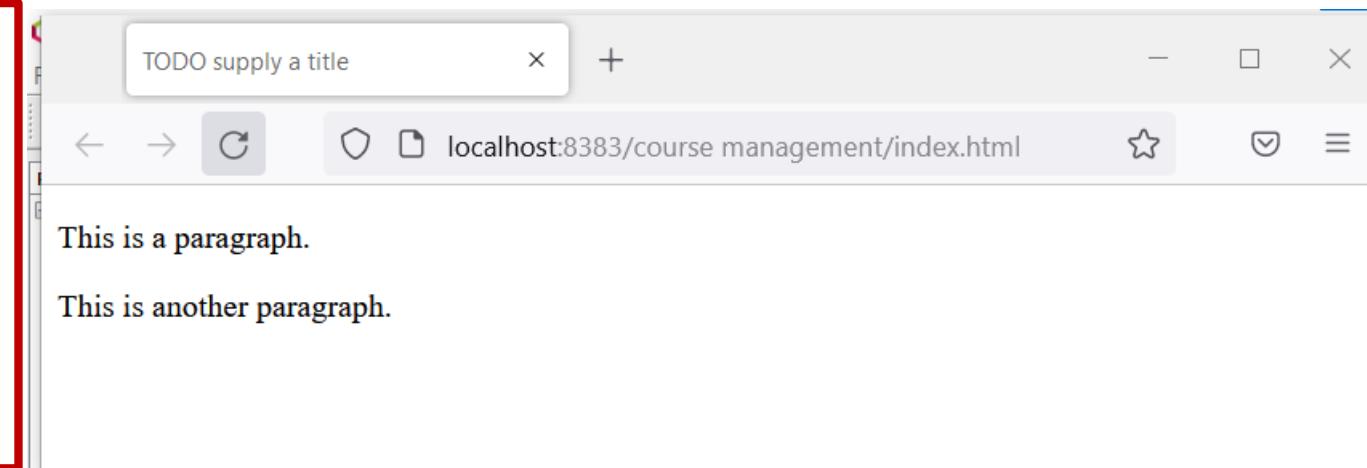
**This is heading 6**

# HTML Paragraphs

HTML paragraphs are defined with the `<p>` tag

Example

```
<p>This is a paragraph.</p>
<p>This is another paragraph.</p>
```



# HTML Links

HTML links are defined with the `<a>` tag

```
<h2>HTML Links</h2>
<p>HTML links are defined with the a tag:</p>

<a href="https://www.google.com">This is a
link</a>
```



Note:

The link's destination is specified in the `href` attribute.

Attributes are used to provide additional information about HTML elements.

# HTML Images

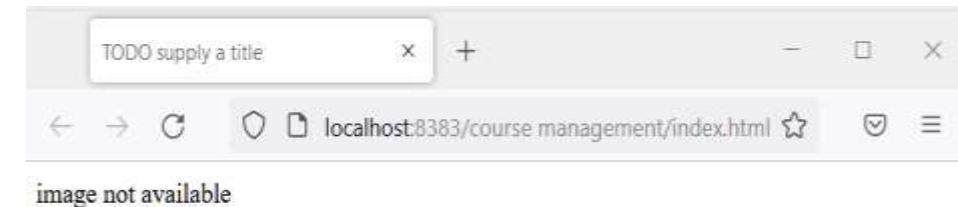
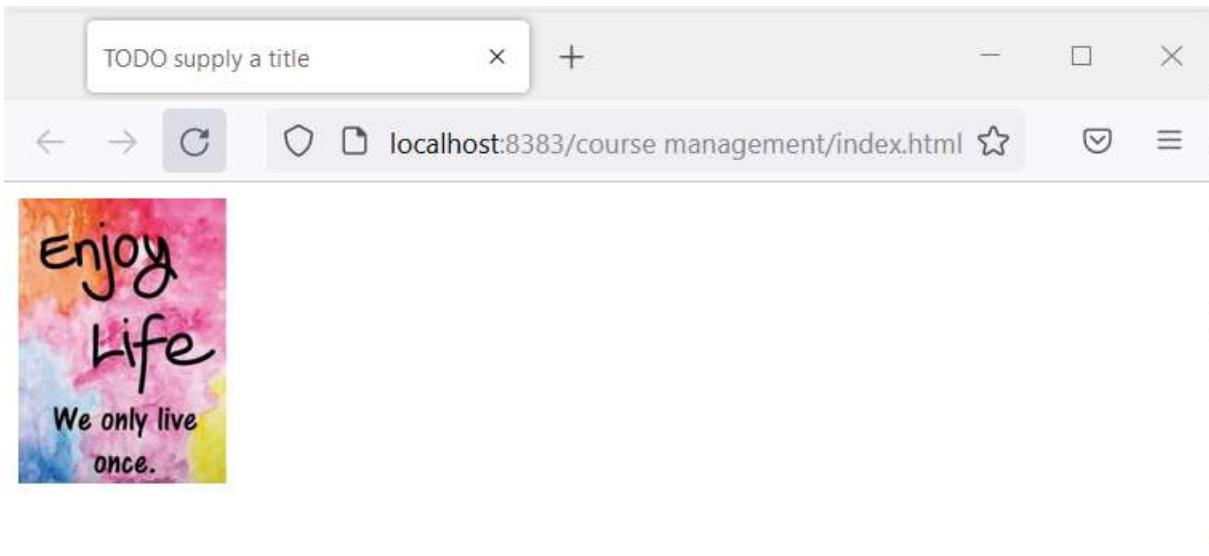
HTML images are defined with the `<img>` tag.

The source file (src), alternative text (alt), width, and height are provided as attributes:

Example

```

```



# HTML Attributes

- All HTML elements can have **attributes**
- Attributes provide **additional information** about elements
- Attributes are always specified in **the start tag**
- Attributes usually come in name/value pairs like: **name="value"**

## The href Attribute

The `<a>` tag defines a hyperlink. The `href` attribute specifies the URL of the page the link goes to:

```
<a href="https://www.google.com">  
Hello Google</a>
```

## The src Attribute

The `<img>` tag is used to embed an image in an HTML page. The `src` attribute specifies the path to the image to be displayed:

```

```

There are two ways to specify the URL in the `src` attribute

1. Absolute URL - Links to an external image that is hosted on another website.  
Example: `src="https://www.ipexpert.com/images/img_girl.jpg"`.
2. Relative URL - Links to an image that is hosted within the website. Here, the URL does not include the domain name.

If the URL begins without a slash, it will be relative to the current page.

Example: `src="img_girl.jpg"`.

If the URL begins with a slash, it will be relative to the domain.

Example: `src="/images/img_girl.jpg"`.

## The width and height Attributes

The <img> tag should also contain the width and height attributes, which specifies the width and height of the image (in pixels):

```

```

## The alt Attribute

The required alt attribute for the <img> tag specifies an alternate text for an image, if the image for some reason cannot be displayed. This can be due to slow connection, or an error in the src attribute, or if the user uses a screen reader.

```

```

## The style Attribute

The style attribute is used to add styles to an element, such as color, font, size, and more.

```
<p style="color:red;">This is a red paragraph.</p>
```

## The lang Attribute

You should always include the lang attribute inside the <html> tag, to declare the language of the Web page. This is meant to assist search engines and browsers.

The following example specifies English as the language:

```
<!DOCTYPE html>
<html lang="en">
<body>
...
</body>
</html>
```

## HTML Horizontal Rules

The `<hr>` tag defines a thematic break in an HTML page, and is most often displayed as a horizontal rule.

The `<hr>` element is used to separate content (or define a change) in an HTML page:

Example:

```
<h1>This is heading 1</h1>
```

```
<p>This is some text.</p>
```

```
<hr>
```

```
<h2>This is heading 2</h2>
```

```
<p>This is some other text.</p>
```

```
<hr>
```

## HTML Line Breaks

The HTML `<br>` element defines a line break.

Use `<br>` if you want a line break (a new line) without starting a new paragraph:

```
<p>This is<br>a paragraph<br>with line breaks.</p>
```

## This is heading 1

This is some text.

---

## This is heading 2

This is some other text.

---

This is  
a paragraph  
with line breaks.

## The HTML <pre> Element

The HTML <pre> element defines preformatted text.

The text inside a <pre> element is displayed in a fixed-width font (usually Courier), and it preserves both spaces and line breaks:

```
<pre>  
  
    My Bonnie lies over the ocean.  
  
    My Bonnie lies over the sea.  
  
    My Bonnie lies over the ocean.  
  
    Oh, bring back my Bonnie to me.  
/>
```

The pre tag preserves both spaces and line breaks:

```
My Bonnie lies over the ocean.  
  
My Bonnie lies over the sea.  
  
My Bonnie lies over the ocean.  
  
Oh, bring back my Bonnie to me.
```

# HTML Styles

## The HTML Style Attribute

Setting the style of an HTML element, can be done with the style attribute.

The HTML style attribute has the following syntax:

```
<tagname style="property:value;">
```

The property is a CSS property. The value is a CSS value.

Example:

```
<body style="background-color:powderblue;">
```

## Try it

Use **background-color** for background color

Use **color** for text colors

Use **font-family** for text fonts

Use **font-size** for text sizes

Use **text-align** for text alignment

# HTML Text Formatting

## HTML Text Formatting Elements

Tag	Description
<u>&lt;b&gt;</u>	Defines bold text
<u>&lt;em&gt;</u>	Defines emphasized text
<u>&lt;i&gt;</u>	Defines a part of text in an alternate voice or mood
<u>&lt;small&gt;</u>	Defines smaller text
<u>&lt;strong&gt;</u>	Defines important text
<u>&lt;sub&gt;</u>	Defines subscripted text
<u>&lt;sup&gt;</u>	Defines superscripted text
<u>&lt;ins&gt;</u>	Defines inserted text
<u>&lt;del&gt;</u>	Defines deleted text
<u>&lt;mark&gt;</u>	Defines marked/highlighted text

# HTML Colors

HTML colors are specified with predefined color names, or with RGB, HEX, HSL, RGBA, or HSLA values.

## Color Names

In HTML, a color can be specified by using a color name:



HTML supports [140 standard color names](#).

## Background Color

```
<h1 style="background-color:DodgerBlue;">Hello World</h1>
```

Hello World

## Text Color

```
<h1 style="color:Tomato;">Hello World</h1>
```

Hello World

## Border Color

```
<h1 style="border:2px solid Tomato;">Hello World</h1>
```

Hello World

## Color Values

rgb(255, 99, 71)

rgba(255, 99, 71, 0.5)

#ff6347

hsla(9, 100%, 64%, 0.5)

hsl(9, 100%, 64%)

# HTML IMAGES

## Image as a Link

To use an image as a link, put the `<img>` tag inside the `<a>` tag:

```
<a href="default.asp">  
    
</a>
```

The most common image file types, which are supported in all browsers (Chrome, Edge, Firefox, Safari, Opera):

Abbreviation	File Format	File Extension
APNG	Animated Portable Network Graphics	.apng
GIF	Graphics Interchange Format	.gif
ICO	Microsoft Icon	.ico, .cur
JPEG	Joint Photographic Expert Group image	.jpg, .jpeg, .jfif, .pjpeg, .pjp
PNG	Portable Network Graphics	.png
SVG	Scalable Vector Graphics	21CS209-Internet Programming <sup>svg</sup>

# HTML Image Tags

<b>Tag</b>	<b>Description</b>
<u>&lt;img&gt;</u>	Defines an image
<u>&lt;map&gt;</u>	Defines an image map
<u>&lt;area&gt;</u>	Defines a clickable area inside an image map
<u>&lt;picture&gt;</u>	Defines a container for multiple image resources

# HTML Image Maps

## Image Maps

The HTML <map> tag defines an image map. An image map is an image with clickable areas. The areas are defined with one or more <area> tags



```


<map name="workmap">
  <area shape="rect" coords="34,44,270,350" alt="Computer"
    href="computer.htm">
  <area shape="rect" coords="290,172,333,250" alt="Phone"
    href="phone.htm">
  <area shape="circle" coords="337,300,44" alt="Coffee" href="coffee.htm">
</map>
```

# Steps to create image maps

## Step 1: Insert the Image

The image is inserted using the `<img>` tag. The only difference from other images is that you must add a `usemap` attribute:

```

```

## Step 2: Create Image Map

Then, add a `<map>` element.

The `<map>` element is used to create an image map, and is linked to the image by using the required `name` attribute:

The name attribute must have the same value as the `<img>`'s `usemap` attribute

## Step 3: add Clickable areas

A clickable area is defined using an `<area>`:

```
<map name="workmap">
```

You must define the shape of the clickable area, and you can choose one of these values:

**rect - defines a rectangular region**

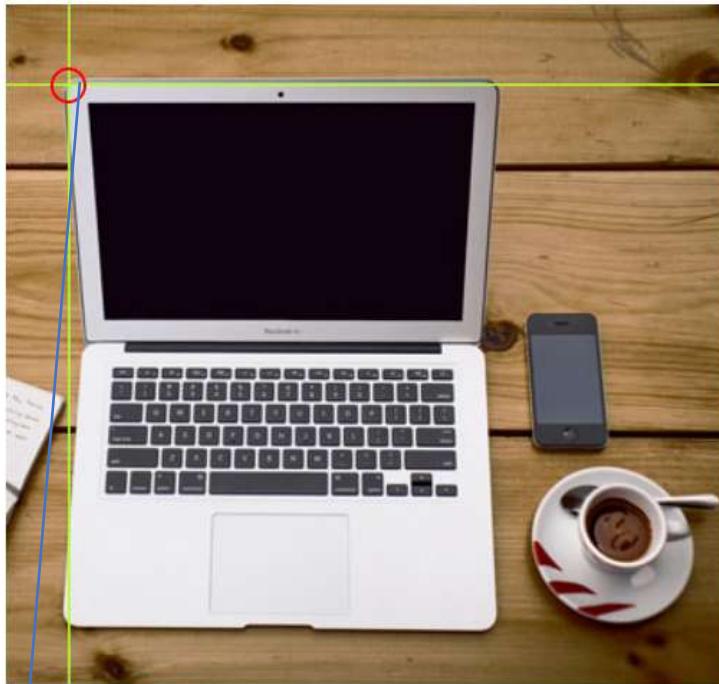
**circle - defines a circular region**

**poly - defines a polygonal region**

**default - defines the entire region**

You must also define some coordinates to be able to place the clickable area onto the image

**Shape="rect"**



(34,44)

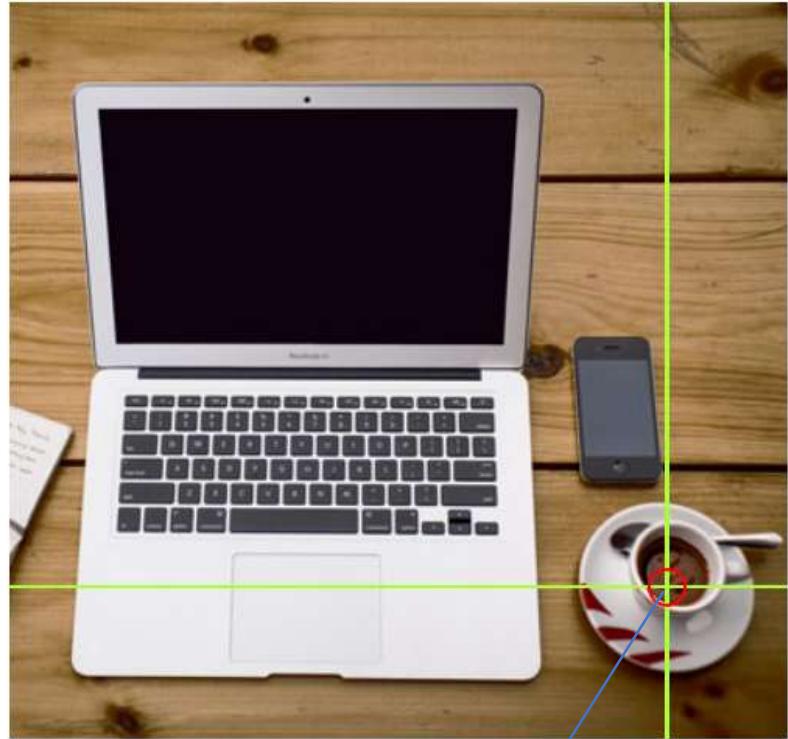
```
<area shape="rect" coords="34,44,270,350" alt="Computer"  
 href="computer.htm">
```



(270,350)



Shape="circle"



(337,300)



Radius=44 pixels

```
<area shape="circle" coords="337, 300, 44" href="coffee.htm">
```

## Shape="poly"

The shape="poly" contains several coordinate points, which creates a shape formed with straight lines (a polygon). This can be used to create any shape.



```
<area shape="poly" coords="140,121,181,116,204,160,204,  
222,191,270,140,329,85,355,58,352,37,322,40,259,103,161  
,128,147" href="croissant.htm">
```



# HTML Background Images

## Background Image on a HTML element

To add a background image on an HTML element, use the HTML style attribute and the CSS background-image property

```
<p>A background image for a p element:</p>
```

```
<p style="background-image: url('img_girl.jpg');">
```

You can specify background images<br>

for any visible HTML element.<br>

In this example, the background image<br>

is specified for a p element.<br>

By default, the background-image<br>

will repeat itself in the direction(s)<br>

where it is smaller than the element<br>

where it is specified. (Try resizing the<br>

browser window to see how the<br>

background image behaves.

```
</p>
```

## Background Image

A background image for a p element:

You can specify background images  
for any visible HTML element.

In this example, the background image  
is specified for a p element.

By default, the background-image  
will repeat itself in the direction(s)

where it is smaller than the element  
where it is specified. (Try resizing the

browser window to see how the  
background image behaves.



## Background Image on a Page

If you want the entire page to have a background image, you must specify the background image on the <body> element

```
<!DOCTYPE html>
<html>
<head>
</head>
<body style=" background-image:
url('img_girl.jpg');">

<h2>Background Image</h2>

<p style="color:Red">By default, the
background image will repeat itself</p>

</body>
</html>
```



# HTML Tables

- HTML tables allow web developers to arrange data into rows and columns.
- A table in HTML consists of table cells inside rows and columns

```
<table>
  <tr>
    <th>Company</th>
    <th>Contact</th>
    <th>Country</th>
  </tr>
  <tr>
    <td>Alfreds Futterkiste</td>
    <td>Maria Anders</td>
    <td>Germany</td>
  </tr>
  <tr>
    <td>Centro comercial Moctezuma</td>
    <td>Francisco Chang</td>
    <td>Mexico</td>
  </tr>
</table>
```

- Each table cell is defined by a `<td>` and a `</td>` tag.
- Each table row starts with a `<tr>` and end with a `</tr>` tag.
- Sometimes you want your cells to be headers, in those cases use the `<th>` tag instead of the `<td>` tag

## HTML Table Borders

```
<style>
table, th, td {
  border: 1px solid black;
}
</style>
```

Firstname	Lastname	Age
Jill	Smith	50
Eve	Jackson	94
John	Doe	80

## Collapsed Table Borders

To avoid having double borders like in the example above, set the CSS border-collapse property to collapse. This will make the borders collapse into a single border:

```
<style>
table, th, td {
  border: 1px solid black;
  border-collapse: collapse;
}
</style>
```

Firstname	Lastname	Age
Jill	Smith	50
Eve	Jackson	94
John	Doe	80

## Style Table Borders

```
table, th, td {  
    border: 1px solid white;  
    border-collapse: collapse;  
}  
  
th, td {  
    background-color: #96D4D4;  
}
```

Firstname	Lastname	Age
Jill	Smith	50
Eve	Jackson	94
John	Doe	80

## Round Table Borders

```
table, th, td {  
    border: 1px solid black;  
    border-radius: 10px;  
}
```

Firstname	Lastname	Age
Jill	Smith	50
Eve	Jackson	94
John	Doe	80

## Dotted Table Borders

```
th, td {  
    border-style: dotted;  
}
```

Firstname	Lastname	Age
Jill	Smith	50
Eve	Jackson	94
John	Doe	80

## HTML Table Sizes

### HTML Table Width

To set the width of a table, add the style attribute to the <table> element:

```
<table style="width:100%">
  <tr>
    <th>Firstname</th>
    <th>Lastname</th>
    <th>Age</th>
  </tr>
  <tr>
    <td>Jill</td>
    <td>Smith</td>
    <td>50</td>
  </tr>
  <tr>
    <td>Eve</td>
    <td>Jackson</td>
    <td>94</td>
  </tr>
</table>
```

Firstname	Lastname	Age
Jill	Smith	50
Eve	Jackson	94
John	Doe	80

## Header for Multiple Columns- **colspan** attribute

```
<table>
  <tr>
    <th colspan="2">Name</th>
    <th>Age</th>
  </tr>
  <tr>
    <td>Jill</td>
    <td>Smith</td>
    <td>50</td>
  </tr>
  <tr>
    <td>Eve</td>
    <td>Jackson</td>
    <td>94</td>
  </tr>
</table>
```

Name		Age
Jill	Smith	50
Eve	Jackson	94

## Table Caption – **Caption tag**

```
<table style="width:100%">
  <caption>Monthly savings</caption>
  <tr>
    <th>Month</th>
    <th>Savings</th>
  </tr>
  <tr>
    <td>January</td>
    <td>$100</td>
  </tr>
  <tr>
    <td>February</td>
    <td>$50</td>
  </tr>
</table>
```

Monthly savings	
Month	Savings
January	\$100
February	\$50

# HTML Table Tags

<b>Tag</b>	<b>Description</b>
<u>&lt;table&gt;</u>	Defines a table
<u>&lt;th&gt;</u>	Defines a header cell in a table
<u>&lt;tr&gt;</u>	Defines a row in a table
<u>&lt;td&gt;</u>	Defines a cell in a table
<u>&lt;caption&gt;</u>	Defines a table caption
<u>&lt;colgroup&gt;</u>	Specifies a group of one or more columns in a table for formatting
<u>&lt;col&gt;</u>	Specifies column properties for each column within a <colgroup> element
<u>&lt;thead&gt;</u>	Groups the header content in a table
<u>&lt;tbody&gt;</u>	Groups the body content in a table
<u>&lt;tfoot&gt;</u>	Groups the footer content in a table

# HTML Lists

HTML lists allow web developers to group a set of related items in lists.

## Example

An unordered HTML list:

- Item
- Item
- Item
- Item

An ordered HTML list:

1. First item
2. Second item
3. Third item
4. Fourth item

## Unordered HTML List

An unordered list starts with the `<ul>` tag. Each list item starts with the `<li>` tag.

```
<h2>An unordered HTML list</h2>
```

```
<ul>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>
```

### An unordered HTML list

- Coffee
- Tea
- Milk

The CSS `list-style-type` property is used to define the style of the list item marker. It can have one of the following values

Value	Description
disc	Sets the list item marker to a bullet (default)
circle	Sets the list item marker to a circle
square	Sets the list item marker to a square
none	The list items will not be marked

## Example - Disc

```
<h2>Unordered List with Disc Bullets</h2>
```

```
<ul style="list-style-type:disc;">
<li>Coffee</li>
<li>Tea</li>
<li>Milk</li>
</ul>
```

### Unordered List with Disc Bullets

- Coffee
- Tea
- Milk

## Example - Circle

```
<h2>Unordered List with Circle Bullets</h2>
```

```
<ul style="list-style-type:circle;">
<li>Coffee</li>
<li>Tea</li>
<li>Milk</li>
</ul>
```

### Unordered List with Circle Bullets

- Coffee
- Tea
- Milk

## Ordered HTML List

An ordered list starts with the `<ol>` tag. Each list item starts with the `<li>` tag. The list items will be marked with numbers by default:

```
<h2>An ordered HTML list</h2>
```

```
<ol>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
```

### An ordered HTML list

- 1. Coffee
- 2. Tea
- 3. Milk

## Ordered HTML List - The Type Attribute

The type attribute of the `<ol>` tag, defines the type of the list item marker

Type	Description
<code>type="1"</code>	The list items will be numbered with numbers (default)
<code>type="A"</code>	The list items will be numbered with uppercase letters
<code>type="a"</code>	The list items will be numbered with lowercase letters
<code>type="I"</code>	The list items will be numbered with uppercase roman numbers
<code>type="i"</code>	The list items will be numbered with lowercase roman numbers

## **Example: Uppercase Letters**

<h2>Ordered List with Letters</h2>

```
<ol type="A">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
```

### **Ordered List with Letters**

- A. Coffee
- B. Tea
- C. Milk

## **Example: Uppercase Roman Numbers**

<h2>Ordered List with Roman Numbers</h2>

```
<ol type="I">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
```

### **Ordered List with Roman Numbers**

- I. Coffee
- II. Tea
- III. Milk

## HTML Description Lists

HTML also supports description lists.

A description list is a list of terms, with a description of each term.

The `<dl>` tag defines the description list, the `<dt>` tag defines the term (name), and the `<dd>` tag describes each term

```
<h2>A Description List</h2>
```

```
<dl>
  <dt>Coffee</dt>
  <dd>- black hot drink</dd>
  <dt>Milk</dt>
  <dd>- white cold drink</dd>
</dl>
```

### A Description List

```
Coffee
  - black hot drink
Milk
  - white cold drink
```

### HTML List Tags

Tag	Description
<code>&lt;ul&gt;</code>	Defines an unordered list
<code>&lt;ol&gt;</code>	Defines an ordered list
<code>&lt;li&gt;</code>	Defines a list item
<code>&lt;dl&gt;</code>	Defines a description list
<code>&lt;dt&gt;</code>	Defines a term in a description list
<code>&lt;dd&gt;</code>	Describes the term in a description list

# Block-level Elements

A block-level element always starts on a new line, and the browsers automatically add some space (a margin) before and after the element.

A block-level element always takes up the full width available (stretches out to the left and right as far as it can).

Two commonly used block elements are: <p> and <div>.

- The <p> element defines a paragraph in an HTML document.
- The <div> element defines a division or a section in an HTML document.

<address>	<article>	<aside>	<blockquote>	<canvas>	<dd>	<div>	<dl>
<dt>	<fieldset>	<figcaption>	<figure>	<footer>	<form>	<h1>-<h6>	<header>
<hr>	<li>	<main>	<nav>	<noscript>	<ol>	<p>	<pre>
<section>	<table>	<tfoot>	<ul>	<video>			

# Inline Elements

An inline element does not start on a new line.

An inline element only takes up as much width as necessary.

Example: This is a `<span>` element inside a paragraph.

<code>&lt;a&gt;</code>	<code>&lt;abbr&gt;</code>	<code>&lt;acronym&gt;</code>	<code>&lt;b&gt;</code>	<code>&lt;bdo&gt;</code>	<code>&lt;big&gt;</code>	<code>&lt;br&gt;</code>	<code>&lt;button&gt;</code>
<code>&lt;cite&gt;</code>	<code>&lt;code&gt;</code>	<code>&lt;dfn&gt;</code>	<code>&lt;em&gt;</code>	<code>&lt;i&gt;</code>	<code>&lt;img&gt;</code>	<code>&lt;input&gt;</code>	<code>&lt;kbd&gt;</code>
<code>&lt;label&gt;</code>	<code>&lt;map&gt;</code>	<code>&lt;object&gt;</code>	<code>&lt;output&gt;</code>	<code>&lt;q&gt;</code>	<code>&lt;samp&gt;</code>	<code>&lt;script&gt;</code>	<code>&lt;select&gt;</code>
<code>&lt;small&gt;</code>	<code>&lt;span&gt;</code>	<code>&lt;strong&gt;</code>	<code>&lt;sub&gt;</code>	<code>&lt;sup&gt;</code>	<code>&lt;textarea&gt;</code>	<code>&lt;time&gt;</code>	<code>&lt;tt&gt;</code>
<code>&lt;var&gt;</code>							

# HTML Iframes

An HTML iframe is used to display a web page within a web page.

The HTML <iframe> tag specifies an inline frame.

An inline frame is used to embed another document within the current HTML document.

## Syntax

```
<iframe src="url" title="description"></iframe>
```

## Iframe - Set Height and Width

Use the height and width attributes to specify the size of the iframe.

The height and width are specified in pixels by default:

```
<iframe src="demo_iframe.htm" height="200" width="300" title="Iframe Example"></iframe>
```

# HTML Forms

An HTML form is used to collect user input. The user input is most often sent to a server for processing.

First name:

Last name:

---

## The `<form>` Element

The HTML `<form>` element is used to create an HTML form for user input:

```
<form>
  .
  .
  </form>
```

# HTML Form Attributes

## The Action Attribute

The action attribute defines the action to be performed when the form is submitted.

Usually, the form data is sent to a file on the server when the user clicks on the submit button.

```
<form action="/action_page.php">
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname"
value="John"><br>
  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname"
value="Doe"><br><br>
  <input type="submit" value="Submit">
</form>
```

## The Target Attribute

The target attribute specifies where to display the response that is received after submitting the form.

The target attribute can have one of the following values:

Value	Description
_blank	The response is displayed in a new window or tab
_self	The response is displayed in the current window
_parent	The response is displayed in the parent frame
_top	The response is displayed in the full body of the window
framename	The response is displayed in a named iframe

## The Method Attribute

The method attribute specifies the HTTP method to be used when submitting the form data.

The form-data can be sent as URL variables (with method="get") or as HTTP post transaction (with method="post").

The default HTTP method when submitting form data is GET.

```
<h2>The method Attribute</h2>
<p>This form will be submitted using the GET method:</p>
<form action="/action_page.php" target="_blank" method="get">
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname" value="John"><br>
  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname"
  value="Doe"><br><br>
  <input type="submit" value="Submit">
</form>
<p>After you submit, notice that the form values is visible in the
address bar of the new browser tab.</p>
```

### The method Attribute

This form will be submitted using the GET method:

First name:

Last name:

After you submit, notice that the form values is visible in the address bar of the new browser tab.

### Submitted Form Data

Your input was received as:

The server has processed your input and returned this answer.

# The HTML <form> Elements

- `<input>`
- `<label>`
- `<select>`
- `<textarea>`
- `<button>`
- `<fieldset>`
- `<legend>`
- `<datalist>`
- `<output>`
- `<option>`
- `<optgroup>`

## The `<label>` Element

The `<label>` element defines a label for several form elements.

The `<label>` element is useful for screen-reader users, because the screen-reader will read out loud the label when the user focus on the input element.

The **for attribute** of the `<label>` tag should be equal to the `id` attribute of the `<input>` element to bind them together.

## The <input> Element

One of the most used form element is the <input> element.

The <input> element can be displayed in several ways, depending on the type attribute.

- `<input type="button">`
- `<input type="checkbox">`
- `<input type="color">`
- `<input type="date">`
- `<input type="datetime-local">`
- `<input type="email">`
- `<input type="file">`
- `<input type="hidden">`
- `<input type="image">`
- `<input type="month">`
- `<input type="number">`
- `<input type="password">`
- `<input type="radio">`
- `<input type="range">`
- `<input type="reset">`
- `<input type="search">`
- `<input type="submit">`
- `<input type="tel">`
- `<input type="text">`
- `<input type="time">`
- `<input type="url">`
- `<input type="week">`

```
<form action="/action_page.php">
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname" value="John"><br>
  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname" value="Doe"><br><br>
  <input type="submit" value="Submit">
  <input type="reset">
</form>
```

First name:

Last name:

## Input Type Checkbox

```
<form>
  <input type="checkbox" id="vehicle1" name="vehicle1" value="Bike">
  <label for="vehicle1"> I have a bike</label><br>
  <input type="checkbox" id="vehicle2" name="vehicle2" value="Car">
  <label for="vehicle2"> I have a car</label><br>
  <input type="checkbox" id="vehicle3" name="vehicle3" value="Boat">
  <label for="vehicle3"> I have a boat</label>
</form>
```

□ I have a bike  
□ I have a car  
□ I have a boat

## Input Type Color

```
<h2>Show a Color Picker</h2>
<p>The <strong>input type="color"</strong> is used for input fields that should contain a color.</p>
<form action="/action_page.php">
  <label for="favcolor">Select your favorite color:</label>
  <input type="color" id="favcolor" name="favcolor" value="#ff0000">
  <input type="submit" value="Submit">
</form>
```

### Show a Color Picker

The `input type="color"` is used for input fields that should contain a color.

Select your favorite color:  Submit

Note: `type="color"` is not supported in Internet Explorer 11 or Safari 9.1 (or earlier).

<p><b>Note:</b> `type="color"` is not supported in Internet Explorer 11 or Safari 9.1 (or earlier).</p>

## Input Type Date

<h2>Date Field</h2>

<p>The <strong>input type="date"</strong> is used for input fields that should contain a date.</p>

```
<form action="/action_page.php">
  <label for="birthday">Birthday:</label>
  <input type="date" id="birthday" name="birthday">
  <input type="submit" value="Submit">
</form>
```

<p><strong>Note:</strong> type="date" is not supported in Internet Explorer 11 or prior Safari 14.1.</p>

## Date Field

The **input type="date"** is used for input fields that should contain a date.

Birthday:

**Note:** type="date" is not supported in Internet Explorer 11 or prior Safari 14.1.

## Input Type Datetime-local

### Local Date Field

The **input type="datetime-local"** specifies a date and time input field, with no time zone.

Birthday (date and time):

Note: type="datetime-local": March, 2022 ▾

Mo	Tu	We	Th	Fr	Sa	Su
28	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31	1	2	3
4	5	6	7	8	9	10

Safari 14.1.

## Input Type Email

### Email Field

The **input type="email"** is used for input fields that should contain an e-mail address:

Enter your email:

Please include an '@' in the email address. 'Hello' is missing an '@'.

## Input Type File

### File upload

Show a file-select field which allows a file to be chosen for upload:

Select a file:  No file chosen

## Input Type Number

```
<form>
  <label for="quantity">Quantity (between 1 and 5):</label>
  <input type="number" id="quantity" name="quantity" min="1" max="5">
</form>
```

Quantity (between 1 and 5):

## Input Type Range

```
<form>
  <label for="vol">Volume (between 0 and 50):</label>
  <input type="range" id="vol" name="vol" min="0" max="50">
</form>
```

Volume (between 0 and 50): 

## The <select> Element

The <select> element defines a drop-down list

```
<label for="cars">Choose a car:</label>
<select id="cars" name="cars">
  <option value="volvo">Volvo</option>
  <option value="saab">Saab</option>
  <option value="fiat">Fiat</option>
  <option value="audi">Audi</option>
</select>
```

```
<option value="fiat" selected>Fiat</option>
```

## The select Element

The select element defines a drop-down list:

Choose a car:

Volvo  
Saab  
Fiat  
Audi

### Visible Values:

Use the size attribute to specify the number of visible values

Choose a car:

## Allow Multiple Selections:

Use the multiple attribute to allow the user to select more than one value:

```
<label for="cars">Choose a car:</label>
<select id="cars" name="cars" size="4" multiple>
  <option value="volvo">Volvo</option>
  <option value="saab">Saab</option>
  <option value="fiat">Fiat</option>
  <option value="audi">Audi</option>
</select>
```

## Allow Multiple Selections

Use the multiple attribute to allow the user to select more than one value.

Choose a car:



Volvo  
Saab  
Fiat  
Audi

Submit

Hold down the Ctrl (windows) / Command (Mac) button to select multiple options.

## The <textarea> Element

The <textarea> element defines a multi-line input field (a text area)

```
<textarea name="message" rows="10" cols="30">  
The cat was playing in the garden.  
</textarea>
```

## Textarea

The textarea element defines a multi-line input field.

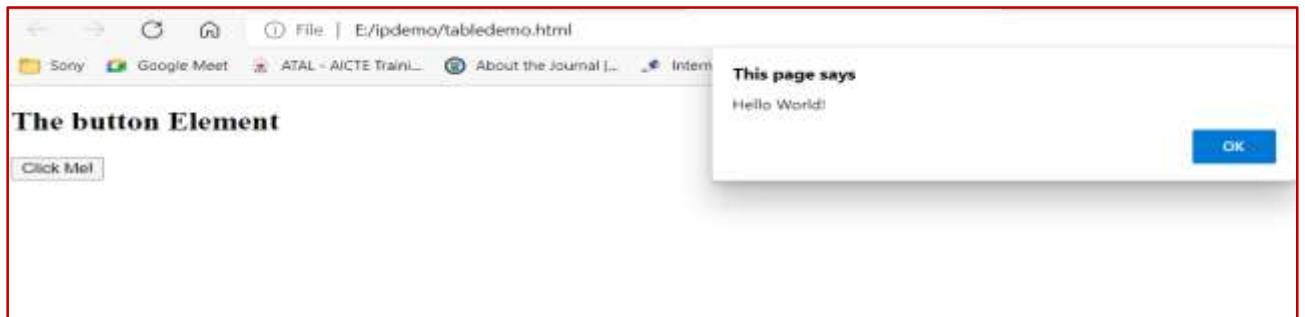
The cat was playing in the  
garden.

Submit

## The <button> Element

The <button> element defines a clickable button:

```
<button type="button" onclick="alert('Hello World!')>Click Me!</button>
```



## The <fieldset> and <legend> Elements

The <fieldset> element is used to group related data in a form.

The <legend> element defines a caption for the <fieldset> element.

```
<form action="/action_page.php">
  <fieldset>
    <legend>Personalia:</legend>
    <label for="fname">First name:</label><br>
    <input type="text" id="fname" name="fname" value="John"><br>
    <label for="lname">Last name:</label><br>
    <input type="text" id="lname" name="lname" value="Doe"><br><br>
    <input type="submit" value="Submit">
  </fieldset>
</form>
```

Personal Details:

First name:  
John

Last name:  
Doe

Submit

# Design the Form

## Fast Forms

**Full Name**

**Date of Birth**

**Weekly Work Capacity**

**Profile Picture**  **No fil...hosen**

**Gender**  Male  
 Female  
 Other  
 I'd rather not specify

**Food Choices**  Vegan  
 Vegetarian  
 Non Vegetarian  
 I'd rather not specify

**Primary Language**

**Your Email**

**Your Phone Number**

**Your Website**

**Your Password**

**Confirm Password**

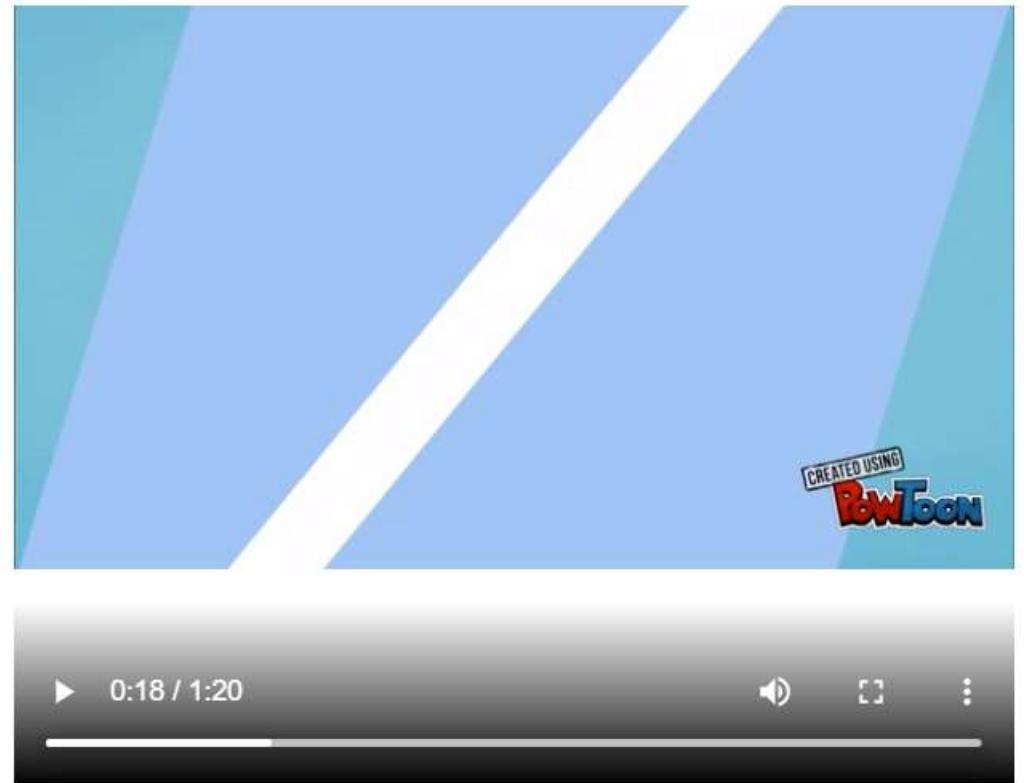
**Terms and Conditions**

# HTML Video

## The HTML <video> Element

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

```
<video width="500" height="500" controls>
  <source src="CSVideo.mp4" type="video/mp4">
    Your browser does not support the video tag.
</video>
```



# HTML Audio

## The HTML <audio> Element

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

```
<audio controls autoplay>
```

```
  <source src="musicAR.mp3" type="audio/mpeg">
Your browser does not support the audio element.
</audio>
```



# CSS3

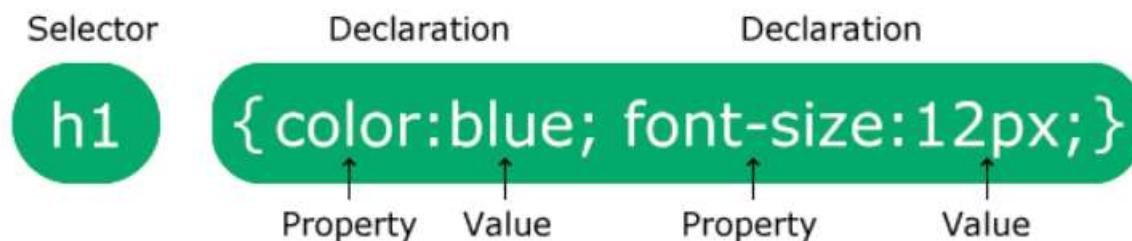


## **CASCADING STYLE SHEETS**

# What is CSS?

- CSS stands for Cascading Style Sheets
- CSS describes how HTML elements are to be displayed on screen, paper, or in other media
- CSS saves a lot of work. It can control the layout of multiple web pages all at once
- External stylesheets are stored in CSS files

# CSS Syntax



- The selector points to the HTML element you want to style.
- The declaration block contains one or more declarations separated by semicolons.
- Each declaration includes a CSS property name and a value, separated by a colon.
- Multiple CSS declarations are separated with semicolons, and declaration blocks are surrounded by curly braces.

# CSS Selectors

A CSS selector selects the HTML element(s) you want to style.

## CSS Selectors

CSS selectors are used to "find" (or select) the HTML elements you want to style.

We can divide CSS selectors into five categories:

- Simple selectors (select elements based on name, id, class)
- Combinator selectors (select elements based on a specific relationship between them)
- Pseudo-class selectors (select elements based on a certain state)
- Pseudo-elements selectors (select and style a part of an element)
- Attribute selectors (select elements based on an attribute or attribute value)

## The CSS element Selector

The element selector selects HTML elements based on the element name.

```
<style>
p {
    text-align: center;
    color: red;
}
</style>
```

```
<p>Every paragraph will be affected by the style.</p>
<p id="para1">Me too!</p>
<p>And me!</p>
```

Every paragraph will be affected by the style.

Me too!

And me!

## The CSS id Selector

The id selector uses the id attribute of an HTML element to select a specific element. The id of an element is unique within a page, so the id selector is used to select one unique element!

To select an element with a specific id, write a hash (#) character, followed by the id of the element.

```
<style>
#para1 {
    text-align: center;
    color: red;
}
</style>
```

```
<p id="para1">Hello World!</p>
<p>This paragraph is not affected by the style.</p>
```

Hello World!

This paragraph is not affected by the style.

# The CSS class Selector

The class selector selects HTML elements with a specific class attribute.

To select elements with a specific class, write a period (.) character, followed by the class name.

```
<style>  
.center {  
    text-align: center;  
    color: red;  
}  
</style>
```

```
<h1 class="center">Red and center-aligned heading</h1>  
<p class="center">Red and center-aligned paragraph.</p>
```

**Red and center-aligned heading**

Red and center-aligned paragraph.

```
<style>
p.center {
    text-align: center;
    color: red;
}
</style>
```

```
<h1 class="center">This heading will not be affected</h1>
<p class="center">This paragraph will be red and center-aligned.</p>
```

**This heading will not be affected**

This paragraph will be red and center-aligned.

```
<style>
p.center {
    text-align: center;
    color: red;
}
p.large {
    font-size: 300%;
}
</style>
```

```
<h1 class="center">This heading will not be affected</h1>
<p class="center">This paragraph will be red and center-aligned.</p>
<p class="center large">This paragraph will be red, center-aligned, and in a large font-size.</p>
```

**This heading will not be affected**

This paragraph will be red and center-aligned.

**This paragraph will be red, center-aligned, and in a large font-size.**

# The CSS Universal Selector

The universal selector (\*) selects all HTML elements on the page.

```
<style>
* {
    text-align: center;
    color: blue;
}
</style>
```

```
<h1>Hello world!</h1>
```

```
<p>Every element on the page will be affected by the style.</p>
<p id="para1">Me too!</p>
<p>And me!</p>
```

**Hello world!**

Every element on the page will be affected by the style.

Me too!

And me!

## The CSS Grouping Selector

The grouping selector selects all the HTML elements with the same style definitions.  
Look at the following CSS code (the h1, h2, and p elements have the same style definitions):

```
<style>  
h1, h2, p {  
    text-align: center;  
    color: red;  
}  
</style>
```

```
<h1>Hello World!</h1>  
<h2>Smaller heading!</h2>  
<p>This is a paragraph.</p>
```

Hello World!

Smaller heading!

This is a paragraph.

# Three Ways to Insert CSS

There are three ways of inserting a style sheet:

- External CSS
- Internal CSS
- Inline CSS

## External CSS

With an external style sheet, you can change the look of an entire website by changing just one file!

Each HTML page must include a reference to the external style sheet file inside the `<link>` element, inside the head section.

```
<head>
<link rel="stylesheet" href="mystyle.css">
</head>

<h1>This is a heading</h1>
<p>This is a
paragraph.</p>
```

### mystyle.css

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

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

This is a heading

This is a paragraph.

## Internal CSS

An internal style sheet may be used if one single HTML page has a unique style. The internal style is defined inside the `<style>` element, inside the head section.

```
<style>
body {
    background-color: linen;
}

h1 {
    color: maroon;
    margin-left: 40px;
}
</style>
```

```
<h1>This is a heading</h1>
<p>This is a paragraph.</p>
```

**This is a heading**

This is a paragraph.

## Inline CSS

An inline style may be used to apply a unique style for a single element.

To use inline styles, add the style attribute to the relevant element. The style attribute can contain any CSS property.

```
<h1 style="color:blue;text-align:center;">This is a heading</h1>  
<p style="color:red;">This is a paragraph.</p>
```

This is a heading

This is a paragraph.

## Cascading Order

What style will be used when there is more than one style specified for an HTML element?  
All the styles in a page will "cascade" into a new "virtual" style sheet by the following rules,  
where number one has the highest priority:

- 1.Inline style (inside an HTML element)
- 2.Internal style sheets
- 3.External Style sheets has the least Priority
- 4.Browser default

```
<head>
<link rel="stylesheet" type="text/css" href="mystyle.css">
<style>
h1 {
  color: orange;
}
</style></head>
<h1>This is a heading</h1>
<p>The style of this document is a combination of an external
stylesheet, and internal style</p>
```

```
mystyle.css
h1 {
  color: navy;
}
```

This is a heading

The style of this document is a combination of an external stylesheet, and internal style

# Pseudo-classes

- A Pseudo class in CSS is used to define the special state of an element. It can be combined with a CSS selector to add an effect to existing elements based on their states.
- For Example, changing the style of an element when the user hovers over it, or when a link is visited.

## Syntax:

```
selector: pseudo-class{  
    property: value;  
}
```

**:hover Pseudo-class:** This pseudo-class is used to add special effect to an element when our mouse pointer is over it

```
<div class="box">  
    My color changes if you hover over me!  
</div>
```

```
.box{  
    background-color: yellow;  
    width: 300px;  
    height: 200px;  
    margin: auto;  
    font-size: 40px;  
    text-align: center;  
}  
  
.box:hover{  
    background-color: orange;  
}
```

My color  
changes if you  
hover over  
me!

**:active Pseudo-class:** This pseudo-class is used to select an element which is activated when the user clicks on it.

```
<div class="box">  
    My color changes for a moment if you click me!  
</div>
```

```
.box{  
    background-color: yellow;  
    width: 300px;  
    height: 200px;  
    margin: auto;  
    font-size: 40px;  
    text-align: center;  
}
```

My color  
changes for a  
moment if you  
click me!

```
.box:active{  
    background-color: orange;  
}
```

**:focus Pseudo-class:** This pseudo-class is used to select an element which is currently focused by the user. It works on user input elements used in forms and is triggered as soon as the user clicks on it.

```
<form>
  <label for="username">Username:</label>
  <input type="text" name="username"
    placeholder="Enter your username" />
  <br>

  <label for="emailid">Email-Id:</label>
  <input type="email" name="emailid"
    placeholder="Enter your email-id" />

  <label for="Password">Password:</label>
  <input type="password" name="Password"
    placeholder="Enter your password" />
</form>
```

```
form{
  width: 300px;
  height: 200px;
  margin: 0 auto;
  text-align: center;
  line-height: 2rem
}

label{
  width: 30%;
}

input{
  background-color: default;
  float: right;
}

input:focus{
  background-color: grey;
}
```

Username:

Email-Id:

Password:

**:visited Pseudo-class:** This pseudo-class is used to select the links which have been already visited by the user.

```
<p>
  <a href="https://www.google.com/" target="_blank">
    My color changes once you vist this link
  </a>
</p>
```

```
a:visited{
  color: red;
}
```

# CSS Backgrounds

The CSS background properties are used to add background effects for elements.

- `background-color`
- `background-image`
- `background-repeat`
- `background-attachment`
- `background-position`
- `background` (shorthand property)

## CSS background-color

With CSS, a color is most often specified by:

- a valid color name - like "red"
- a HEX value - like "#ff0000"
- an RGB value - like "rgb(255,0,0)"

## CSS background-color example!

```
<style>
h1 {
    background-color: green;
}

div {
    background-color: lightblue;
}

p {
    background-color: yellow;
}
</style>
```

```
<body>

<h1>CSS background-color
example!</h1>
<div>
This is a text inside a div element.
<p>This paragraph has its own
background color.</p>
We are still in the div element.
</div>

</body>
```

This is a text inside a div element.

This paragraph has its own background color.

We are still in the div element.

## Opacity / Transparency

The opacity property specifies the opacity/transparency of an element. It can take a value from 0.0 - 1.0. The lower value, the more transparent:

```
<style>
div {
    background-color: green;
}

div.first {
    opacity: 0.1;
}

div.second {
    opacity: 0.3;
}

div.third {
    opacity: 0.6;
}
</style>
```

<p>When using the opacity property to add transparency to the background of an element, all of its child elements become transparent as well. This can make the text inside a fully transparent element hard to read:</p>

```
<div class="first">
    <h1>opacity 0.1</h1>
</div>
```

When using the opacity property to add transparency to the background of an element, all of its child elements become transparent as well. This can make the text inside a fully transparent element hard to read:

opacity 0.1

```
<div class="second">
    <h1>opacity 0.3</h1>
</div>
```

opacity 0.3

```
<div class="third">
    <h1>opacity 0.6</h1>
</div>
```

opacity 0.6

```
<div>
    <h1>opacity 1 (default)</h1>
</div>
```

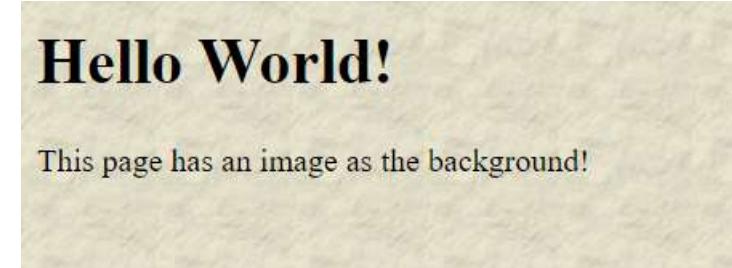
opacity 1 (default)

## CSS background-image

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

```
<style>  
body {  
    background-image: url("paper.gif");  
}  
</style>  
<h1>Hello World!</h1>
```

```
<p>This page has an image as the background!</p>
```



## CSS background-repeat

By default, the background-image property repeats an image both horizontally and vertically.

Some images should be repeated only horizontally or vertically, or they will look strange, like this:

```
<style>
body {
    background-image: url("gradient_bg.png");
    background-repeat: repeat-x;
}
</style>
<h1>Hello World!</h1>
<p>Here, a background image is repeated only horizontally!</p>
```

**Hello World!**

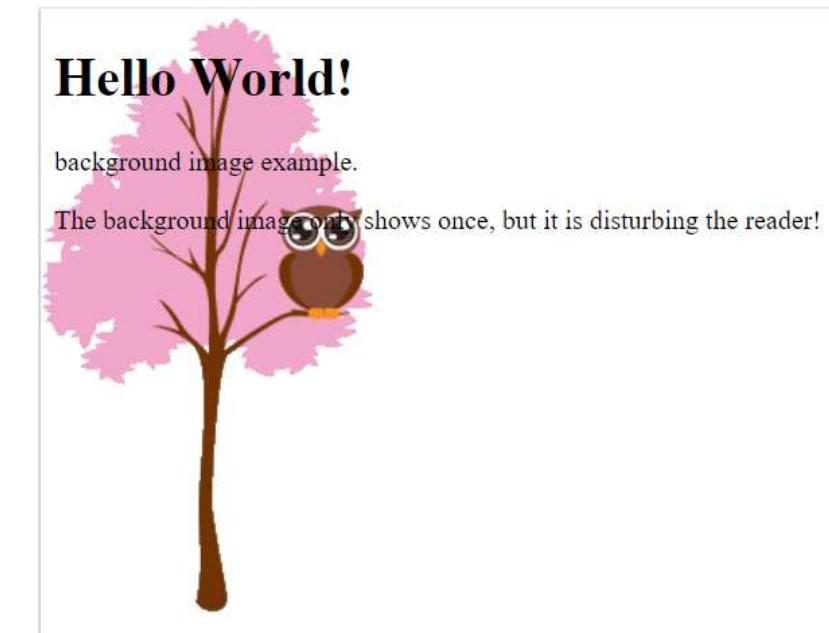
Here, a background image is repeated only horizontally!

## CSS background-repeat: no-repeat

Showing the background image only once is also specified by the background-repeat property:

```
<style>  
body {  
background-image: url("img_tree.png");  
background-repeat: no-repeat;  
}  
</style>
```

```
<h1>Hello World!</h1>  
<p>background image example.</p>  
<p>The background image only shows once, but it is disturbing the reader!</p>
```



## CSS background-position

```
<style>
body {
background-image: url("img_tree.png");
background-repeat: no-repeat;
background-position: right top;
margin-right: 200px;
}
</style>
```

```
<h1>Hello World!</h1>
<p>Here, the background image is only shown once. In
addition it is positioned away from the text.</p>
<p>In this example we have also added a margin on the
right side, so that the background image will not disturb
the text.</p>
```

## Hello World!

Here, the background image is only shown once. In addition it is positioned away from the text.

In this example we have also added a margin on the right side, so that the background image will not disturb the text.



## CSS background-attachment

The background-attachment property specifies whether the background image should scroll or be fixed (will not scroll with the rest of the page)

```
body {  
    background-  
    image: url("img_tree.png");  
    background-repeat: no-repeat;  
    background-position: right top;  
    background-attachment: scroll;  
}
```

## The background-attachment Property

The background-attachment property specifies whether the background image should scroll or be fixed (will not scroll with the rest of the page).

**Tip:** If you do not see any scrollbars, try to resize the browser window.

The background-image scrolls. Try to scroll down the page.

The background-image scrolls. Try to scroll down the page.

The background-image scrolls. Try to scroll down the page.

The background-image scrolls. Try to scroll down the page.

The background-image scrolls. Try to scroll down the page.

The background-image scrolls. Try to scroll down the page.

The background-image scrolls. Try to scroll down the page.

The background-image scrolls. Try to scroll down the page.

The background-image scrolls. Try to scroll down the page.

The background-image scrolls. Try to scroll down the page.

The background-image scrolls. Try to scroll down the page.



## CSS background - Shorthand property

```
<style>
body {
  background: #ffffff url("img_tree.png") no-repeat right
top;
  margin-right: 200px;
}
```

### The background Property

The background property is a shorthand property for specifying all the background properties in one declaration.

Here, the background image is only shown once, and it is also positioned in the top-right corner.

We have also added a right margin, so that the text will not write over the background image.

### The background Property

The background property is a shorthand property for specifying all the background properties in one declaration.

Here, the background image is only shown once, and it is also positioned in the top-right corner.

We have also added a right margin, so that the text will not write over the background image.



# CSS Borders

The CSS border properties allow you to specify the style, width, and color of an element's border.

## **border-style property.**

- dotted - Defines a dotted border
- dashed - Defines a dashed border
- solid - Defines a solid border
- double - Defines a double border
- groove - Defines a 3D grooved border. The effect depends on the border-color value
- ridge - Defines a 3D ridged border. The effect depends on the border-color value
- inset - Defines a 3D inset border. The effect depends on the border-color value
- outset - Defines a 3D outset border. The effect depends on the border-color value
- none - Defines no border
- hidden - Defines a hidden border

The border-style property can have from one to four values (for the top border, right border, bottom border, and the left border).

```
<style>
p.dotted {border-style: dotted;}
p.dashed {border-style: dashed;}
p.solid {border-style: solid;}
p.double {border-style: double;}
p.groove {border-style: groove;}
p.ridge {border-style: ridge;}
p.inset {border-style: inset;}
p.outset {border-style: outset;}
p.none {border-style: none;}
p.hidden {border-style: hidden;}
p.mix {border-style: dotted dashed solid double;}
</style>
```

## The border-style Property

This property specifies what kind of border to display:

A dotted border.

A dashed border.

A solid border.

A double border.

A groove border.

A ridge border.

An inset border.

An outset border.

No border.

A hidden border.

A mixed border.

## The border-style Property

This property specifies what kind of border to display:

A dotted border.

A dashed border.

A solid border.

A double border.

A groove border.

A ridge border.

An inset border.

An outset border.

No border.

A hidden border.

A mixed border.

## border-width property

The width can be set as a specific size (in px, pt, cm, em, etc) or by using one of the three pre-defined values: thin, medium, or thick:

```
<style>
p.one {
    border-style: solid;
    border-width: 5px;
}
p.two {
    border-style: solid;
    border-width: medium;
}
p.three {
    border-style: dotted;
    border-width: 2px;
}
p.four {
    border-style: dotted;
    border-width: thick;
}
```

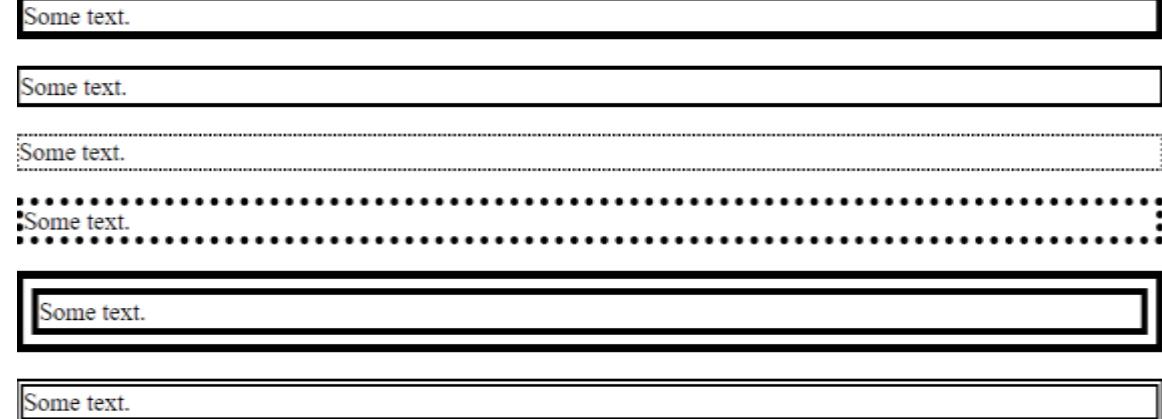
```
p.five {
    border-style: double;
    border-width: 15px;
}
```

```
p.six {
    border-style: double;
    border-width: thick;
}
```

```
</style>
```

```
<p>This property specifies the width of the four borders:</p>
<p class="one">Some text.</p>
<p class="two">Some text.</p>
<p class="three">Some text.</p>
<p class="four">Some text.</p>
<p class="five">Some text.</p>
<p class="six">Some text.</p>
```

This property specifies the width of the four borders:



## CSS Border Color

```
p.one {  
    border-style: solid;  
    border-color: red;  
}  
  
p.two {  
    border-style: solid;  
    border-color: green;  
}  
  
p.three {  
    border-style: dotted;  
    border-color: blue;  
}
```

Red border

Green border

Blue border

The border-color property can have from one to four values (for the top border, right border, bottom border, and the left border).

```
p.one {  
    border-style: solid;  
    border-color: red green blue  
    yellow; /* red top, green right,  
    blue bottom and yellow left */  
}
```



A solid multicolor border

# CSS Border Sides

In CSS, there are also properties for specifying each of the borders (top, right, bottom, and left)

```
p {  
    border-top-style: dotted;  
    border-right-style: solid;  
    border-bottom-style: dotted;  
    border-left-style: solid;  
}
```

Different Border Styles

If the border-style property has four values:

border-style: dotted solid double dashed;

top border is dotted

right border is solid

bottom border is double

left border is dashed

If the border-style property has three values:

border-style: dotted solid double;

top border is dotted

right and left borders are solid

bottom border is double

If the border-style property has two values:

border-style: dotted solid;

top and bottom borders are dotted

right and left borders are solid

If the border-style property has one value:

border-style: dotted;

all four borders are dotted

4 different border styles.

3 different border styles.

2 different border styles.

1 border style.

## CSS Border - Shorthand Property

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

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

border-width

border-style (required)

border-color

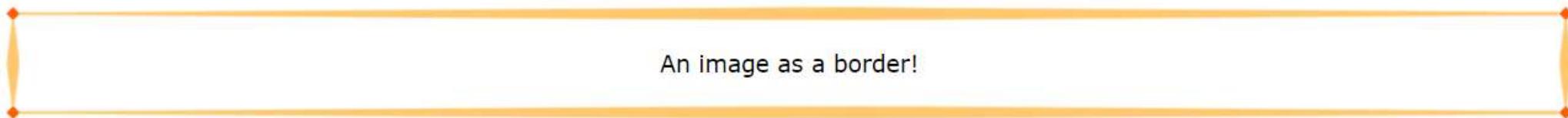
Some text

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

# border-image property

- [border-image-source](#)
- [border-image-slice](#)
- [border-image-width](#)
- [border-image-outset](#)
- [border-image-repeat](#)

```
#borderimg {  
    border: 10px solid transparent;  
    padding: 15px;  
    border-image: url(border.png) 30  
    round;  
}
```



# CSS Margins

Margins are used to create space around elements, outside of any defined borders.

This element has a margin of 70px.

CSS has properties for specifying the margin for each side of an element:

- margin-top
- margin-right
- margin-bottom
- margin-left

All the margin properties can have the following values:

- auto - the browser calculates the margin
- length - specifies a margin in px, pt, cm, etc.
- % - specifies a margin in % of the width of the containing element
- inherit - specifies that the margin should be inherited from the parent element

# CSS Padding

Padding is used to create space around an element's content, inside of any defined borders.

CSS has properties for specifying the padding for each side of an element:

padding-top  
padding-right  
padding-bottom  
padding-left

All the padding properties can have the following values:

- length* - specifies a padding in px, pt, cm, etc.
- % - specifies a padding in % of the width of the containing element
- inherit - specifies that the padding should be inherited from the parent element

This element has a padding of 70px.

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
    border: 1px solid black;
    background-color: lightblue;
    padding-top: 50px;
    padding-right: 30px;
    padding-bottom: 50px;
    padding-left: 80px;
}
</style>
</head>
<body>
<h2>Using individual padding properties</h2>

<div>This div element has a top padding of 50px, a right padding of 30px, a
bottom padding of 50px, and a left padding of 80px.</div>
</body>
</html>
```

## Using individual padding properties

This div element has a top padding of 50px, a right padding of 30px, a bottom padding of 50px, and a left padding of 80px.

# CSS Height, Width and Max-width

The height and width properties are used to set the height and width of an element.

The height and width properties do not include padding, borders, or margins. It sets the height/width of the area inside the padding, border, and margin of the element.

The height and width properties may have the following values:

auto - This is default. The browser calculates the height and width

length - Defines the height/width in px, cm etc.

% - Defines the height/width in percent of the containing block

initial - Sets the height/width to its default value

inherit - The height/width will be inherited from its parent value

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
    height: 100px;
    width: 500px;
    background-color: powderblue;
}
</style>
</head>
<body>

<h2>Set the height and width of an
element</h2>

<div>This div element has a height of 100px and
a width of 500px.</div>

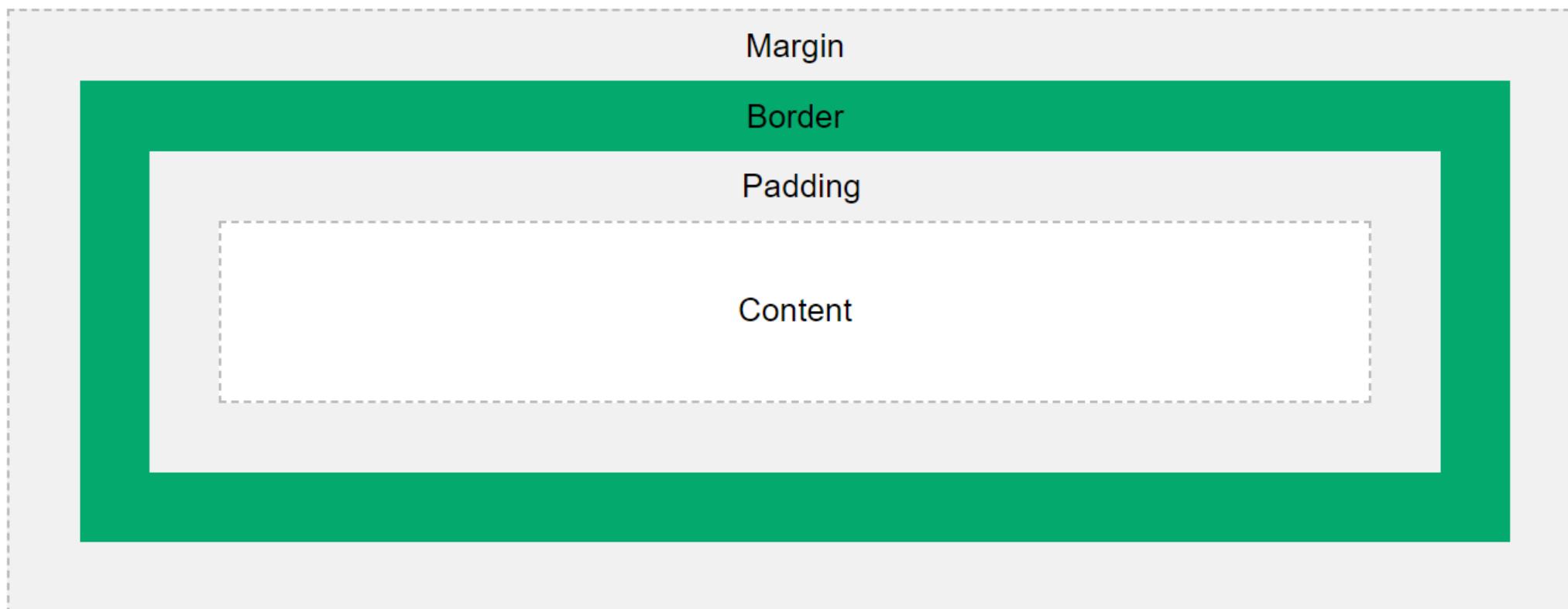
</body>
</html>
```

## Set the height and width of an element

This div element has a height of 100px and a width of 500px.

# CSS Box Model

- In CSS, the term "box model" is used when talking about design and layout.
- The CSS box model is essentially a box that wraps around every HTML element. It consists of: margins, borders, padding, and the actual content.



```
<!DOCTYPE html>
<html>
<head>
<style>
div {
    width: 320px;
    padding: 10px;
    border: 5px solid gray;
    margin: 0;
}
</style>
</head>
<body>
```

## Calculate the total width:

```

<div>The picture above is 350px wide. The total width of this element
is also 350px.</div>
```

```
</body>
</html>
```

**Calculate the total width:**



The picture above is 350px wide. The total width of this element is also 350px.

The total width of an element should be calculated like this:

Total element width = width + left padding + right padding + left border + right border + left margin + right margin

The total height of an element should be calculated like this:

Total element height = height + top padding + bottom padding + top border + bottom border + top margin + bottom margin

# CSS Layout - The position Property

## The position Property

The position property specifies the type of positioning method used for an element.

There are five different position values:

- static
- relative
- fixed
- absolute
- sticky

# CSS Text

## Text Color

The color property is used to set the color of the text. The color is specified by:

a color name - like "red"

a HEX value - like "#ff0000"

an RGB value - like "rgb(255,0,0)"

```
<style>
```

```
body {
```

```
    color: blue;
```

```
}
```

```
h1 {
```

```
    color: green;
```

```
}
```

## This is heading 1

This is an ordinary paragraph. Notice that this text is blue. The default text color for a page is defined in the body selector.

Another paragraph.

## CSS Text Alignment and Text Direction

The `text-align` property is used to set the horizontal alignment of a text.

A text can be left or right aligned, centered, or justified.

**Heading 1 (center)**

```
h1 {  
    text-align: center;  
}
```

The `text-align-last` property specifies how to align the last line of a text.

```
p.c {  
    text-align-last: justify;  
}
```

**text-align-last: justify;**

  Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam semper diam at erat pulvinar, at pulvinar felis blandit. Vestibulum volutpat tellus diam, consequat gravida libero rhoncus ut.

The **direction** and **unicode-bidi** properties can be used to change the text direction of an element:

```
p.ex1 {  
    direction: rtl;  
    unicode-bidi: bidi-override;  
}
```

This is the default text direction.

.noitcerid txet tfel-ot-thgir si sihT

<p>This is the default text direction.</p>

<p class="ex1">This is right-to-left text direction.</p>

The **vertical-align** property sets the vertical alignment of an element.

```
img.b {  
    vertical-align: text-top;}
```

**vertical-align: text-top:**

An  image with a text-top alignment.

## CSS Text Decoration

```
text-decoration-line  
text-decoration-color  
text-decoration-style  
text-decoration-thickness  
text-decoration
```

---

### Overline text decoration

### ~~Line-through text decoration~~

### Underline text decoration

---

Overline and underline text decoration.

```
h1 {  
    text-decoration-line: overline;  
}  
  
h2 {  
    text-decoration-line: line-through;  
}  
  
h3 {  
    text-decoration-line: underline;  
}  
  
p {  
    text-decoration-line: overline  
    underline;  
}
```

```
h1 {  
    text-decoration-line: overline;  
    text-decoration-color: red;  
}
```

## Overline text decoration

```
p.ex3 {  
    text-decoration-line: underline;  
    text-decoration-color: red;  
    text-decoration-style: wavy;  
}
```

Another paragraph.

```
h2 {  
    text-decoration-line: underline;  
    text-decoration-thickness: 5px;  
}  
  
Heading 2
```

```
h3 {  
    text-decoration-line: underline;  
    text-decoration-thickness: 25%;  
}
```

```
p {  
    text-decoration: underline red double  
5px;  
}  
  
A paragraph.
```

## Text Transformation

```
p.uppercase {  
    text-transform: uppercase;  
}
```

THIS TEXT IS TRANSFORMED TO UPPERCASE.

```
p.lowercase {  
    text-transform: lowercase;  
}
```

this text is transformed to lowercase.

```
p.capitalize {  
    text-transform: capitalize;  
}
```

This Text Is Capitalized.

## Text Shadow

The text-shadow property adds shadow to text.

In its simplest use, you only specify the horizontal shadow (2px) , the vertical shadow (2px), blur effect (5px), shadow color (red).

```
h1 {  
    text-shadow: 2px 2px 5px red;  
}
```

**Text-shadow effect!**

# CSS 2D Transforms

CSS transforms allow you to move, rotate, scale, and skew elements.

With the CSS transform property you can use the following 2D transformation methods:

translate()

rotate()

scaleX()

scaleY()

scale()

skewX()

skewY()

skew()

matrix()

```
div {  
    transform: translate(50px, 100px);  
}
```

```
<h1>The translate() Method</h1>  
<p>The translate() method moves an element from its current  
position:</p>
```

```
<div>
```

This div element is moved 50 pixels to the right, and 100 pixels down  
from its current position.

```
</div>
```

## The translate() Method

The translate() method moves an element from its current position:

This div element is moved 50 pixels to the  
right, and 100 pixels down from its current  
position.

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
    width: 300px;
    height: 100px;
    background-color: yellow;
    border: 1px solid black;
}

div#myDiv {
    transform: rotate(20deg);
}
</style>
</head>
```

```
<body>

<h1>The rotate() Method</h1>

<p>The rotate() method rotates an element clockwise or counter-clockwise.</p>

<div>
This a normal div element.
</div>

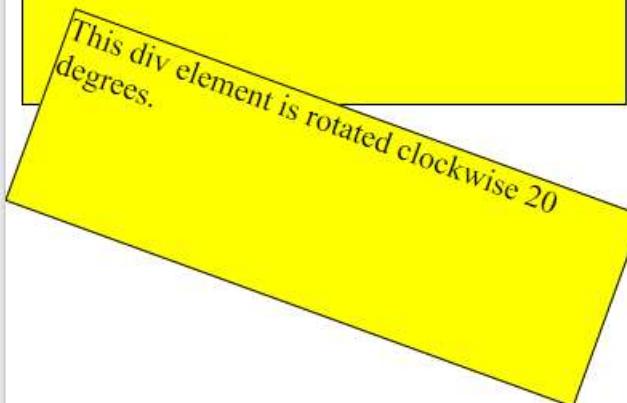
<div id="myDiv">
This div element is rotated clockwise 20 degrees.
</div>

</body>
</html>
```

## The rotate() Method

The rotate() method rotates an element clockwise or counter-clockwise.

This a normal div element.



```
<!DOCTYPE html>
<html>
<head>
<style>
div {
    margin: 150px;
    width: 200px;
    height: 100px;
    background-color: yellow;
    border: 1px solid black;
    transform: scale(2,3);
}
</style>
</head>
<body>
    <h1>The scale()
Method</h1>
    <p>The scale() method
increases or decreases the
size of an element.</p>
    <div>
        This div element is two
        times of its original width,
        and three times of its
        original height.
    </div>
</body>
</html>
```

## The scale() Method

The `scale()` method increases or decreases the size of an element.

This div element is two times of its original width, and three times of its original height.

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
    width: 300px;
    height: 100px;
    background-color: yellow;
    border: 1px solid black;
}

div#myDiv {
    transform: skewX(20deg);
}
</style>
</head>
```

```
<body>
<h1>The skewX() Method</h1>

<p>The skewX() method skews an element along the X-axis by the given angle.</p>

<div>
This a normal div element.
</div>

<div id="myDiv">
This div element is skewed 20 degrees along the X-axis.
</div>

</body>
</html>
```

## The skewX() Method

The skewX() method skews an element along the X-axis by the given angle.

This a normal div element.

This div element is skewed 20 degrees along the X-axis.

# CSS Transitions

CSS transitions allows you to change property values smoothly, over a given duration.

```
<style>
div {
  width: 100px;
  height: 100px;
  background: red;
  transition: width 2s;
}

div:hover {
  width: 300px;
}
</style>
```

```
<h1>The transition
Property</h1>

<p>Hover over the div
element below, to see the
transition effect:</p>
<div></div>
```

## The transition Property

Hover over the div element below, to see the transition effect:



## The transition Property

Hover over the div element below, to see the transition effect:



# 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 as 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.

```
<style>
div {
  width: 100px;
  height: 100px;
  background-color: red;
  animation-name: example;
  animation-duration: 4s;
}

@keyframes example {
  from {background-color: red;}
  to {background-color: yellow;}
}
</style>
</head>

<h1>CSS Animation</h1>
<div></div>
<p><b>Note:</b> When an animation is
finished, it goes back to its original style.</p>
```

## CSS Animation



**Note:** When an animation is finished, it goes back to its original style.

```
@keyframes boxanim
{
0% {background-color:red; left:0px; top:0px;}
25% {background-color:yellow; left:200px; top:0px;}
50% {background-color:blue; left:200px; top:200px;}
75% {background-color:green; left:0px; top:200px;}
100% {background-color:red; left:0px; top:0px;}
}
```

```
animation-name:boxanim;
animation-duration:4s;
animation-delay:2s;
animation-iteration-count:infinite;
animation-direction:alternate;
```