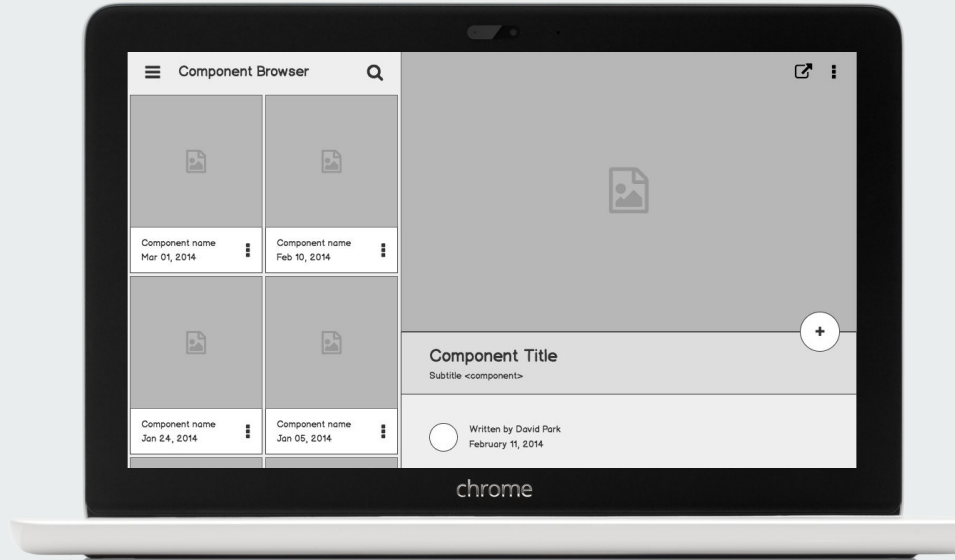




Introduction To Web Technology



Outline

What is Internet ?

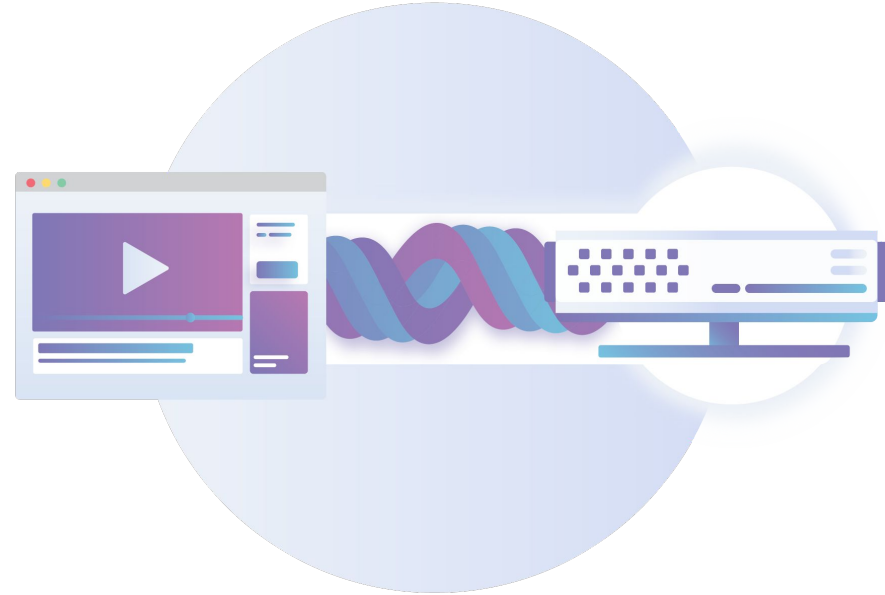
How does your computer connect to the Internet?

What is Website?

What is URL?

What is the internet?

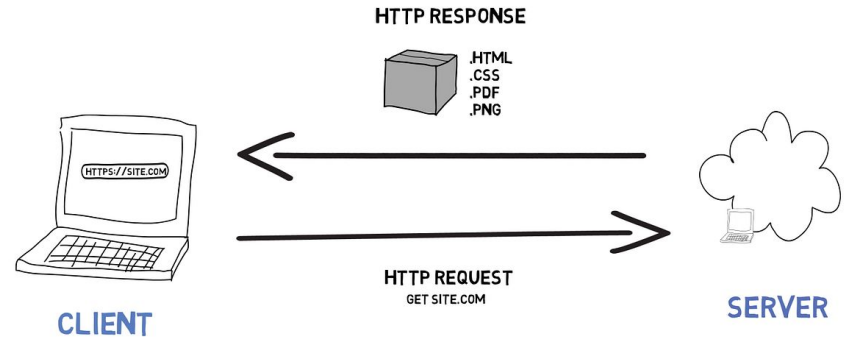
Computers connect to each other and to the Internet via wires, cables, radio waves, and other types of networking infrastructure. All data sent over the Internet is translated into pulses of light or electricity, also called "bits," and then interpreted by the receiving computer. The wires, cables, and radio waves conduct these bits at the speed of light. The more bits that can pass over these wires and cables at once, the faster the Internet works.



How internet works?

There are two main concepts that are fundamental to the way the Internet functions: packets and protocols.

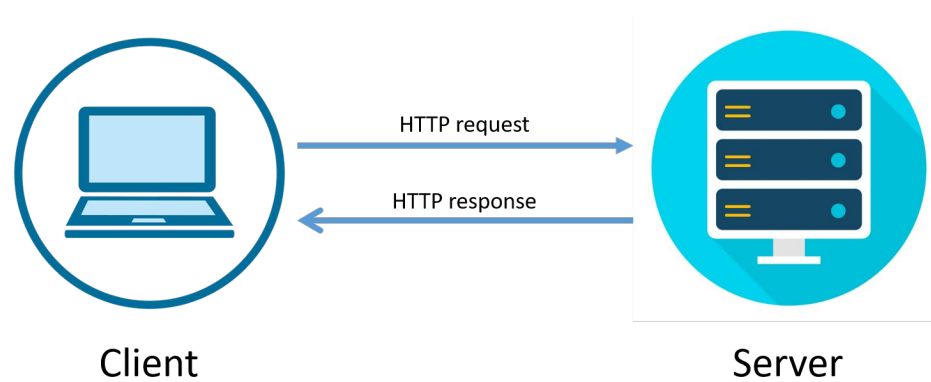
HOW THE INTERNET WORKS



How website run?

A website is a collection of related and linked web pages via hyperlinks, images, videos or other digital assets that are addressed with a common domain name or IP address.

o A web page is a document, typically written in Hypertext Markup Language (e.g. 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

HTML elements label pieces of content such as "this is a heading", "this is a paragraph", "this is a link", etc.

HTML Versions

Version	Year
HTML	1991
HTML 2.0	1995
HTML 3.2	1997
HTML 4.01	1999
XHTML	2000
HTML5	2014

Code Example

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title>Page Title</title>
```

```
</head>
```

```
<body>
```

```
<h1>My First Heading</h1>
```

```
<p>My first paragraph.</p>
```

```
</body>
```

```
</html>
```


HTML Page Structure

Below is a visualization of an 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>
```

Example Explained

- The `<html>` element is the root element of an HTML page
- The `<head>` element contains meta information about the HTML page
- The `<title>` element specifies a title for the HTML page (which is shown in the browser's title bar or in the page's tab)
- The `<body>` element defines the document's body, and is a container for all the visible contents, such as headings, paragraphs, images, hyperlinks, tables, lists, etc.
- The `<h1>` element defines a large heading
- The `<p>` element defines a paragraph

What is an HTML Element?

`<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>`

HTML Editor

Step 1: Open [visual studio code](#) (VSCode) (PC)

Step 2: Write Some HTML

Step 3: Save the HTML Page

Step 4: View the HTML Page in Your Browser

HTML Heading Tags

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

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

`<h3>This is heading 3</h3>`

`<h4>This is heading 1</h4>`

`<h5>This is heading 2</h5>`

`<h6>This is heading 3</h6>`

HTML Paragraphs

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

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

HTML Links

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

HTML Images

```

```


The 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;">
```

HTML Comment Tag

`<!-- Write your comments here -->`

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

`<!-- <p>This is another paragraph </p> -->`

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

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:

HTML Line Breaks

The HTML `
` element defines a line break.

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

Example

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

HTML Tag

An unordered HTML list:

```
<ul>
```

```
  <li>Coffee</li>
```

```
  <li>Tea</li>
```

```
  <li>Milk</li>
```

```
</ul>
```

HTML Tag with style

```
<ul style="list-style-type:circle">
```

```
<ul style="list-style-type:disc">
```

```
<ul style="list-style-type:square">
```

HTML Tag

```
<ol>
```

```
  <li>Coffee</li>
```

```
  <li>Tea</li>
```

```
  <li>Milk</li>
```

```
</ol>
```

HTML Tag

different list types for ol with CSS

```
<ol start="50">
```

```
  <li>Coffee</li>
```

```
  <li>Tea</li>
```

```
  <li>Milk</li>
```

```
</ol>
```


HTML <iframe> Tag

```
<iframe  
  src="https://www.w3schools.com"  
  title="W3Schools Free Online Web Tutorials"  
>  
</iframe>
```

HTML <table> Tag

```
<table>
```

```
<tr>
```

```
<th>Month</th>
```

```
<th>Savings</th>
```

```
</tr>
```

```
<tr>
```

```
<td>January</td>
```

```
<td>$100</td>
```

```
</tr>
```

```
</table>
```

HTML <tr> Tag

```
<tr>
```

```
  <th>Month</th>
```

```
  <th>Savings</th>
```

```
</tr>
```

HTML <th> Tag

```
<tr>
```

```
  <th>Month</th>
```

```
  <th>Savings</th>
```

```
</tr>
```

HTML <td> Tag

```
<tr>
```

```
  <td>Month</td>
```

```
</tr>
```

Colspanned Cell

`<td colspan="2"> </td>`

rowspanned Cell

`<td rowspan="2"> </td>`

HTML Form

HTML Forms

`<form>`

.

form elements

.

`</form>`

HTML Text Fields

```
<form>
```

```
  <input type="text" id="fname" name="fname">
```

```
</form>
```

The <label> Element

```
<label for="html">HTML</label>
```

The Submit Button

```
<form>
```

```
  <input type="submit" value="Submit">
```

```
</form>
```

The Reset Button

```
<form>
```

```
  <input type="reset">
```

```
</form>
```

The Name Attribute for <input>

Notice that each input field must have a name attribute to be submitted.

If the name attribute is omitted, the value of the input field will not be sent at all.

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

HTML Form Attributes

The Action Attribute

```
<form action="/index.html">  
  <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 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.

```
<form action="/action_page.php" method="get">
```

```
<form action="/action_page.php" method="post">
```

HTML Input Types



Input Type Text

`<input type="text">` defines a single-line text input field:



Input Type Password

`<input type="password">` defines a password field:



Input Type Color

```
<form>
```

```
  <label for="favcolor">Select your favorite color:</label>
```

```
  <input type="color" id="favcolor" name="favcolor">
```

```
</form>
```



Input Type Date

```
<form>
```

```
  <label for="birthday">Birthday:</label>
```

```
  <input type="date" id="birthday" name="birthday">
```

```
</form>
```

Radio Buttons

```
<form>
```

```
<input type="radio" id="html" name="fav_language" value="HTML">
```

```
<label for="html">HTML</label><br>
```

```
<input type="radio" id="css" name="fav_language" value="CSS">
```

```
<label for="css">CSS</label><br>
```

```
<input type="radio" id="javascript" name="fav_language"  
value="JavaScript">
```

```
<label for="javascript">JavaScript</label>
```

```
</form>
```

Checkboxes

```
<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>
```




Input Type Email

```
<form>
```

```
  <label for="email">Enter your email:</label>
```

```
  <input type="email" id="email" name="email">
```

```
</form>
```



Input Type File

```
<form>
```

```
  <label for="myfile">Select a file:</label>
```

```
  <input type="file" id="myfile" name="myfile">
```

```
</form>
```



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>
```



Input Type Search

```
<form>
```

```
  <label for="gsearch">Search Google:</label>
```

```
  <input type="search" id="gsearch" name="gsearch">
```

```
</form>
```

HTML Input Attributes



The value Attribute

```
<input type="text" id="fname" name="fname" value="John"><br>
```



The readonly Attribute

```
<form>
```

```
  <label for="fname">First name:</label><br>
```

```
  <input type="text" id="fname" name="fname" value="John" readonly><br>
```

```
  <label for="lname">Last name:</label><br>
```

```
  <input type="text" id="lname" name="lname" value="Doe">
```

```
</form>
```



The disabled Attribute

```
<input type="text" id="fname"  
      name="fname" value="John" disabled>
```




The size Attribute

```
<label for="fname">First name:</label><br>
```

```
<input type="text" id="fname" name="fname" size="50"><br>
```

```
<label for="pin">PIN:</label><br>
```

```
<input type="text" id="pin" name="pin" size="4">
```



The maxlength Attribute

```
<label for="pin">PIN:</label><br>
```

```
<input type="text" id="pin" name="pin" maxlength="4"  
      size="4">
```



The min and max Attributes

```
<label for="quantity">Quantity (between 1 and  
5):</label>
```

```
<input type="number" id="quantity" name="quantity"  
min="1" max="5">
```



The placeholder Attribute

```
<label for="phone">Enter a phone number:</label>
```

```
<input type="tel" id="phone" name="phone"
```

```
placeholder="123-45-678">
```



The required Attribute

```
<form>
```

```
  <label for="username">Username:</label>
```

```
  <input type="text" id="username" name="username" required>
```

```
</form>
```

Questions?

CSS

CSS Saves a Lot of Work!

Selector

h1

Declaration

{ color:blue; font-size:12px; }

Declaration

Property

Value

Property

Value

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

Why Use CSS ?

- The Separation of Structure and Presentation Managing Style at Large Sites
- Improved performance
- Decreased production work
- Rich design and layout

CSS Versions

- Cascading Style Sheets 1 (CSS1)
- Cascading Style Sheets 2 (CSS2, CSS2.1)
- Cascading Style Sheets 3 (CSS3)

How to Link CSS?

- Inline
- Internal
- External

Cascading Order

- Priority 1: Inline styles
- Priority 2: External and internal style sheets
- Priority 3: Browser default

Cascading Order

- **External Style sheet**

```
H3  
{  
  color: red;  
  text-align: left;  
  font-size: 8pt  
}
```

- **Internal Style sheet**

```
h3  
{  
  text-align: right;  
  font-size: 20pt  
}
```

- **Resultant attributes**

```
color: red;  
text-align: right;  
font-size: 20pt
```

```
graph TD; A["H3 { color: red; text-align: left; font-size: 8pt; }"] --- B; C["h3 { text-align: right; font-size: 20pt; }"] --- B; B --> D["color: red; text-align: right; font-size: 20pt;"]
```

CSS Selectors

The CSS element Selector

```
p {
```

```
    text-align: center;
```

```
    color: red;
```

```
}
```

The CSS id Selector

```
#para1 {
```

```
    text-align: center;
```

```
    color: red;
```

```
}
```


The CSS class Selector

```
.center {
```

```
    text-align: center;
```

```
    color: red;
```

```
}
```

The CSS Universal Selector *

```
* {
```

```
    text-align: center;
```

```
    color: blue;
```

```
}
```

The CSS Grouping Selector

```
h1, h2, p {
```

```
    text-align: center;
```

```
    color: red;
```

```
}
```

CSS [attribute] Selector

```
a[target] {
```

```
    background-color: yellow;
```

```
}
```

```
Input [type="button"] {background-color: blue;}
```

Descendant Selector (Space)

```
div p {
```

```
    background-color: yellow;
```

```
}
```

Child Selector (>)

```
div > p {
```

```
background-color: yellow;
```

```
}
```



CSS Pseudo-classes

Syntax

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


Hover on <div>

```
div:hover {  
  
    background-color: blue;  
  
}
```

Anchor Pseudo-classes

```
/* unvisited link */
```

```
a:link {
```

```
    color: #FF0000;
```

```
}
```

```
/* visited link */
```

```
a:visited {
```

```
    color: #00FF00;
```

```
}
```

```
/* selected link */
```

```
a:active {
```

```
    color: #0000FF;
```

```
}
```



CSS Pseudo-elements

Syntax

```
selector::pseudo-element {
```

```
    property: value;
```

```
}
```

The ::first-line Pseudo-element

```
p::first-line {  
  
    color: #ff0000;  
  
    font-variant: small-caps;  
  
}
```

CSS - The ::before Pseudo-element

```
h1::before {
```

```
    content: url(smiley.gif) ;
```

```
}
```

CSS - The ::after Pseudo-element

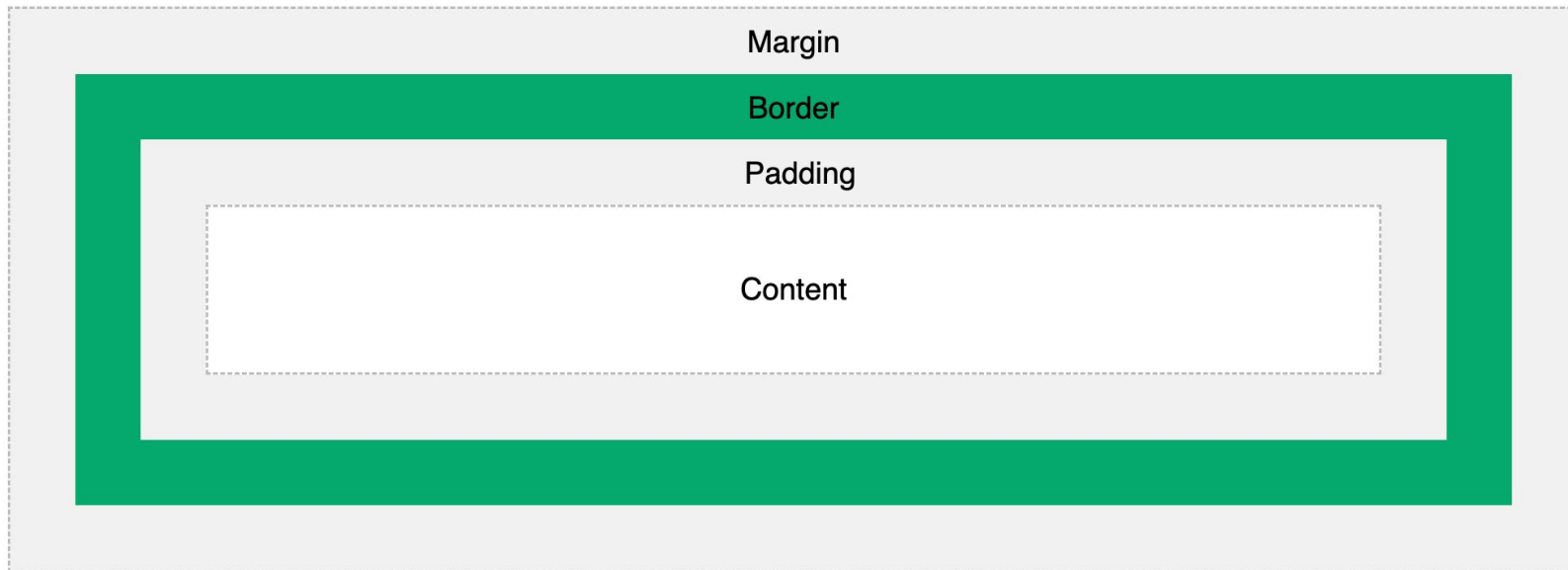
```
h1::after {  
  
    content: url(smiley.gif) ;  
  
}
```



CSS Units

Unit	Description
cm	centimeters
mm	millimeters
in	inches (1in = 96px = 2.54cm)
px *	pixels (1px = 1/96th of 1in)

The CSS Box Model



The CSS Box Model

- Content - The content of the box, where text and images appear
- Padding - Clears an area around the content. The padding is transparent
- Border - A border that goes around the padding and content
- Margin - Clears an area outside the border. The margin is transparent



CSS Layout - The position Property

The CSS Box Model

- static
- relative
- fixed
- absolute
- sticky

position: static;

```
div.static {  
    position: static;  
    border: 3px solid #73AD21;  
}
```

position: relative;

```
div.relative {  
    position: relative;  
    left: 30px;  
    border: 3px solid #73AD21;  
}
```

position: fixed;

```
div.fixed {  
    position: fixed;  
    bottom: 0;  
    right: 0;  
    width: 300px;  
    border: 3px solid #73AD21;  
}
```


position: absolute;

```
div.absolute {  
    position: absolute;  
    top: 80px;  
    right: 0;  
    width: 200px;  
    height: 100px;  
    border: 3px solid #73AD21;  
}
```



CSS Layout - The display Property

Block-level Element

The `<div>` element is a block-level element.

Examples of block-level elements:

- `<div>`
 - `<h1>` - `<h6>`
 - `<p>`
 - `<form>`
 - `<header>`
 - `<footer>`
 - `<section>`
-

Inline Elements

This is an inline `` element inside a paragraph.

Examples of inline elements:

- ``
- `<a>`
- ``

Display: none;

`display: none;` is commonly used with JavaScript to hide and show elements without deleting and recreating them. Take a look at our last example on this page if you want to know how this can be achieved.

The `<script>` element uses `display: none;` as default.

visibility:hidden;

Hiding an element can be done by setting the display property to none. The element will be hidden, and the page will be displayed as if the element is not there:



CSS Layout - Overflow

CSS Overflow

The overflow property has the following values:

- visible - Default. The overflow is not clipped. The content renders outside the element's box
- hidden - The overflow is clipped, and the rest of the content will be invisible
- scroll - The overflow is clipped, and a scrollbar is added to see the rest of the content
- auto - Similar to scroll, but it adds scrollbars only when necessary



CSS Layout - The z-index Property

The z-index Property

The z-index property specifies the stack order of an element (which element should be placed in front of, or behind, the others).

An element can have a positive or negative stack order

This is a heading

Because the image has a z-index of -1, it will be placed behind the text.





CSS Opacity / Transparency

Transparent Image

```
img {
```

```
  opacity: 0.5;
```

```
}
```

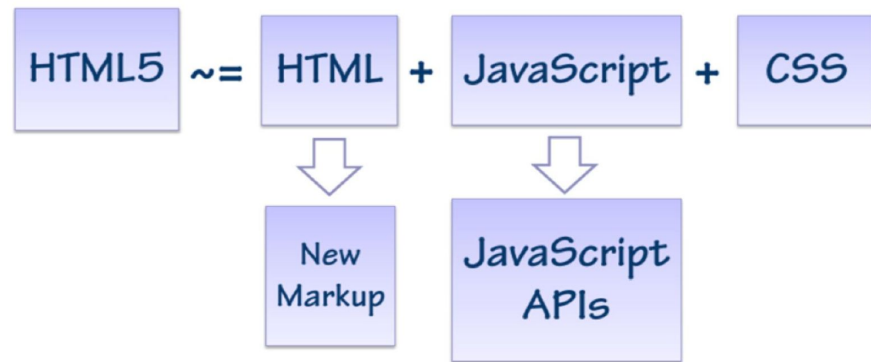
Transparent Hover Effect

```
img {  
  
    opacity: 0.5;  
  
}
```

```
img:hover {  
  
    opacity: 1.0;  
  
}
```

HTML5

HTML5 | Introduction



What's HTML 5?

HTML5 will be the new standard for HTML.

The previous version of HTML, HTML 4.01, came in 1999.

The web has changed a lot since then.

HTML5 is still a work in progress. However, the major browsers support many of the new HTML5 elements and APIs.

What's HTML 5?

HTML5 is a cooperation between the World Wide Web Consortium (W3C) and the Web Hypertext Application Technology Working Group (WHATWG).

WHATWG was working with web forms and applications, and W3C was working with XHTML 2.0. In 2006, they decided to cooperate and create a new version of HTML.

Some rules for HTML5 were established

New features should be based on HTML, CSS, DOM, and JavaScript

Reduce the need for external plugins (like Flash)

Better error handling

More markup to replace scripting

HTML5 should be device independent

The development process should be visible to the public

HTML 5 Simple page

❑ The HTML5 <!DOCTYPE>

- In HTML5 there is only one <!doctype> declaration, and it is very simple:

```
<!DOCTYPE html>
```

❑ Example for HTML 5 simple page:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Title of the document</title>
  </head>

  <body>
    The content of the document.....
  </body>
</html>
```

Some rules for HTML5 were established

New features should be based on HTML, CSS, DOM, and JavaScript

Reduce the need for external plugins (like Flash)

Better error handling

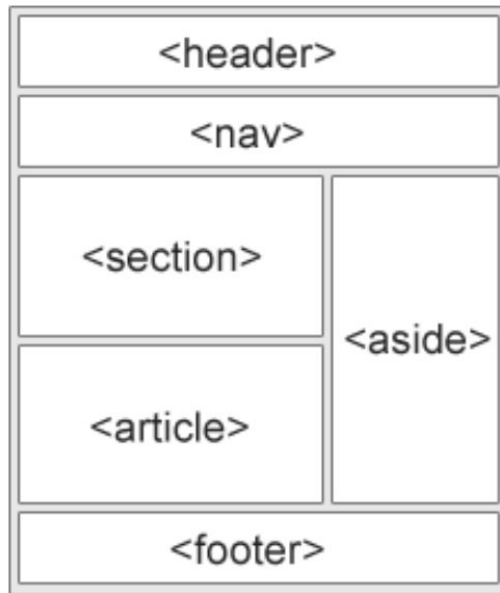
More markup to replace scripting

HTML5 should be device independent

The development process should be visible to the public

HTML Semantic Elements

- `<article>`
- `<aside>`
- `<details>`
- `<figcaption>`
- `<figure>`
- `<footer>`
- `<header>`
- `<main>`
- `<mark>`
- `<nav>`
- `<section>`
- `<summary>`
- `<time>`



Structural Semantic Usage

Typical HTML4	Typical HTML5
<code><div id="header"></code>	<code><header></code>
<code><div id="menu"></code>	<code><nav></code>
<code><div id="content"></code>	<code><section></code>
<code><div id="post"></code>	<code><article></code>
<code><div id="footer"></code>	<code><footer></code>

HTML <audio> Tag

```
<audio controls>
```

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

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

Your browser does not support the audio tag.

```
</audio>
```

HTML <video> Tag

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

```
<source src="movie.mp4" type="video/mp4">
```

```
<source src="movie.ogg" type="video/ogg">
```

Your browser does not support the video tag.

```
</video>
```

Questions?

CSS3

