

# HyperText Markup Language



# Html - rfc1866 rfc2854

## World Wide Web

The WorldWideWeb (W3) is a wide-area [hypermedia](#) information retrieval initiative aiming to give universal access to a large universe

Everything there is online about W3 is linked directly or indirectly to this document, including an [executive summary](#) of the project, [N](#)

### [What's out there?](#)

Pointers to the world's online information, [subjects](#) , [W3 servers](#) , etc.

### [Help](#)

on the browser you are using

### [Software Products](#)

A list of W3 project components and their current state. (e.g. [Line Mode](#) , [X11 Viola](#) , [NeXTStep](#) , [Servers](#) , [Tools](#) , [Mail robot](#) , [Lib](#)

### [Technical](#)

Details of protocols, formats, program internals etc

### [Bibliography](#)

Paper documentation on W3 and references.

### [People](#)

A list of some people involved in the project.

### [History](#)

A summary of the history of the project.

### [How can I help ?](#)

If you would like to support the web..

### [Getting code](#)

Getting the code by [anonymous FTP](#) , etc.

In March 1989, Berners-Lee gave managers at CERN a proposal for an information management system that used hypertext to link documents on different computers that were connected to the Internet. (Hypertext, a term coined in 1963, allows a person to get a document or piece of content by clicking on a coded word or phrase.)



# Xml – (Extensible Markup Language) - rfc5364

```
<note>
  <to>Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend!</body>
</note>
```

Esempi:

[https://www.w3schools.com/xml/xml\\_examples.asp](https://www.w3schools.com/xml/xml_examples.asp)

Publicato nel 1998 - <https://www.w3.org/XML/>

XML supporta lo scambio di informazioni tra sistemi informatici come siti Web, database e applicazioni di terze parti

<https://aws.amazon.com/it/what-is/xml/>

# Html - rfc1866 rfc2854

```
<!DOCTYPE html>
<html lang="en">

<meta charset="utf-8">
<title>Page Title</title>

<body>
  <h1>This is a Heading</h1>
  <p>This is a paragraph.</p>
  <p>This is another paragraph.</p>
</body>

</html>
```

HTML elements are the building blocks of HTML pages



# Html - rfc1866 rfc2854

```
<div style="background-color:lightblue">  
  <h3>This is a heading</h3>  
  <p>This is a paragraph.</p>  
</div>
```

```
<a href="https://www.w3schools.com">This is  
a link</a>
```

```
id</b> ="table01"                                      |
| class       | <p <b>class</b> ="normal">                                       |
| style       | <p <b>style</b> ="font-size:16px">                               |
| data-       | <div <b>data-id</b> ="500">                                      |
| onclick     | <input <b>onclick</b> ="myFunction()">                           |
| onmouseover | <a <b>onmouseover</b> ="this.setAttribute('style','color:red')"> |

# CSS - Cascading Style Sheets - rfc7993

CSS



CSS describes how **HTML** elements are to be **displayed**

```
<style>
```

```
body {background-color:lightblue; text-align:center;}  
h1 {color:blue; font-size:40px;}  
p {font-family:verdana; font-size:20px;}
```

```
</style>
```

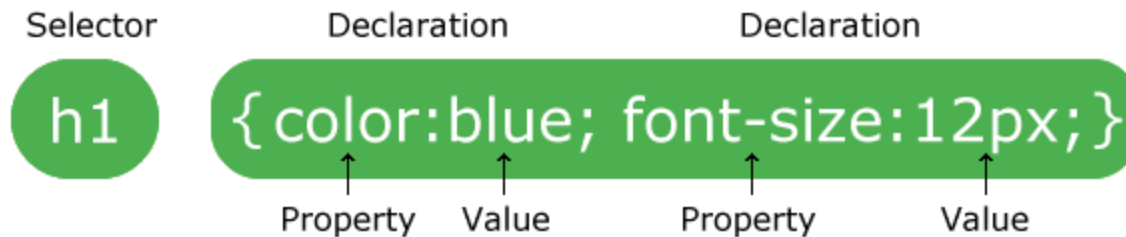
## My First CSS Heading

My first CSS paragraph.

[https://www.w3schools.com/whatis/whatis\\_css.asp](https://www.w3schools.com/whatis/whatis_css.asp)

# CSS - Cascading Style Sheets - rfc7993

A CSS rule consists of a **selector** and a **declaration** block:



[https://www.w3schools.com/cssref/css\\_selectors.asp](https://www.w3schools.com/cssref/css_selectors.asp)

<https://www.w3schools.com/cssref/trysel.asp>

# CSS - Cascading Style Sheets - rfc7993

## Bootstrap

Build responsive, mobile-first projects on the web with the world's most popular front-end component library.

Bootstrap is an open source toolkit for developing with HTML, CSS, and JS. Quickly prototype your ideas or build your entire app with our Sass variables and mixins, responsive grid system, extensive prebuilt components, and powerful plugins built on jQuery.

Get started

Download

Currently v4.3.1



```
<button type="button" class="btn btn-primary">Primary</button>
<button type="button" class="btn btn-secondary">Secondary</button>
<button type="button" class="btn btn-success">Success</button>
<button type="button" class="btn btn-danger">Danger</button>
<button type="button" class="btn btn-warning">Warning</button>
<button type="button" class="btn btn-info">Info</button>
<button type="button" class="btn btn-light">Light</button>
<button type="button" class="btn btn-dark">Dark</button>

<button type="button" class="btn btn-link">Link</button>
```

Copy

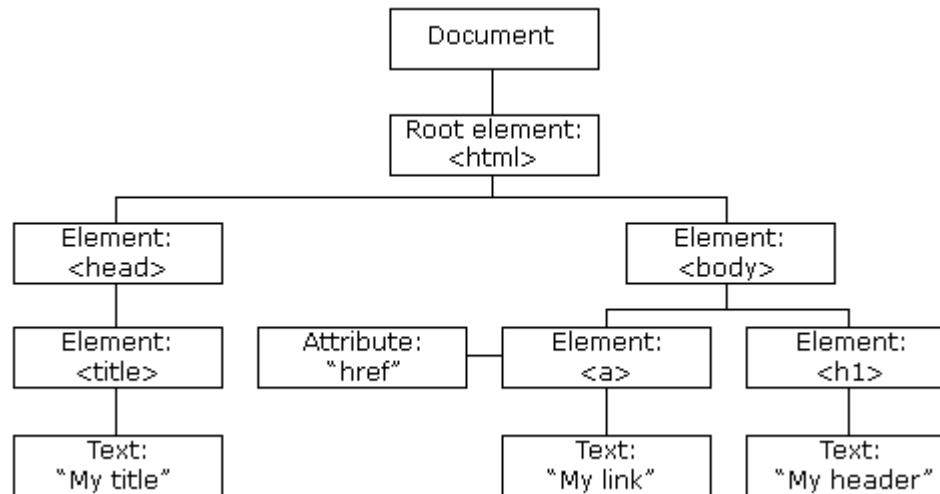




# HTML e DOM

The HTML DOM (Document Object Model)

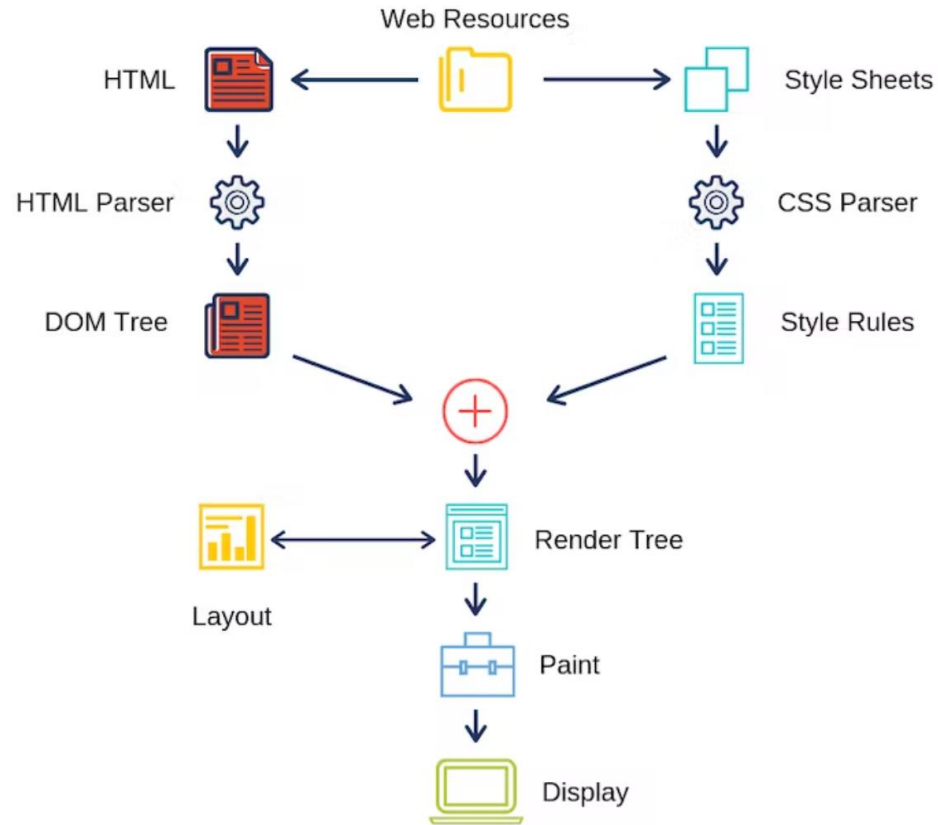
When a web page is loaded, the browser creates a **D**ocument **O**bject **M**odel of the page.



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

[https://www.w3schools.com/whatis/whatis\\_htmlDOM.asp](https://www.w3schools.com/whatis/whatis_htmlDOM.asp)

# How Do Web Browsers Work?



<https://hackernoon.com/how-do-web-browsers-work-40cefd2cb1e1>

# Security

## SQL Injection



## Cosa è:

**SQL injection** è una tecnica di *code injection* dove si inietta del codice SQL

```
# Define POST variables
uname = request.POST['username']
passwd = request.POST['password']

# SQL query vulnerable to SQLi
sql = "SELECT id FROM users WHERE username='" + uname + "' AND password='" + passwd + "'"

# Execute the SQL statement
database.execute(sql)
```

<https://www.acunetix.com/websitesecurity/sql-injection/>

## Come si combatte?

Semplicemente usando: **prepared statements and parameterized queries**

```
$stmt = $dbConnection->prepare('SELECT * FROM employees WHERE name = ?');  
$stmt->bind_param('s', $name);
```

Oppure pulendo tutti gli input:

```
mysqli_real_escape_string ( mysqli $link , string $escapestr ) : string
```

This function is used to create a legal SQL string that you can use in an SQL statement. The given string is encoded to an escaped SQL string, taking into account the current character set of the connection.

```
$unsafe_variable = $_POST["user-input"];  
$safe_variable = mysqli_real_escape_string($unsafe_variable);  
mysql_query("INSERT INTO table (column) VALUES (" . $safe_variable . ")");
```