

# CSE104 Web Programming

**HTML/CSS**

# Introduction to HTML

# What is HTML

## **HTML : HyperText Markup Language**

- A **descriptive** language based on tags (Markups)
- Handle **links** between html pages (Hypertext)

```
<!DOCTYPE html>
<html>

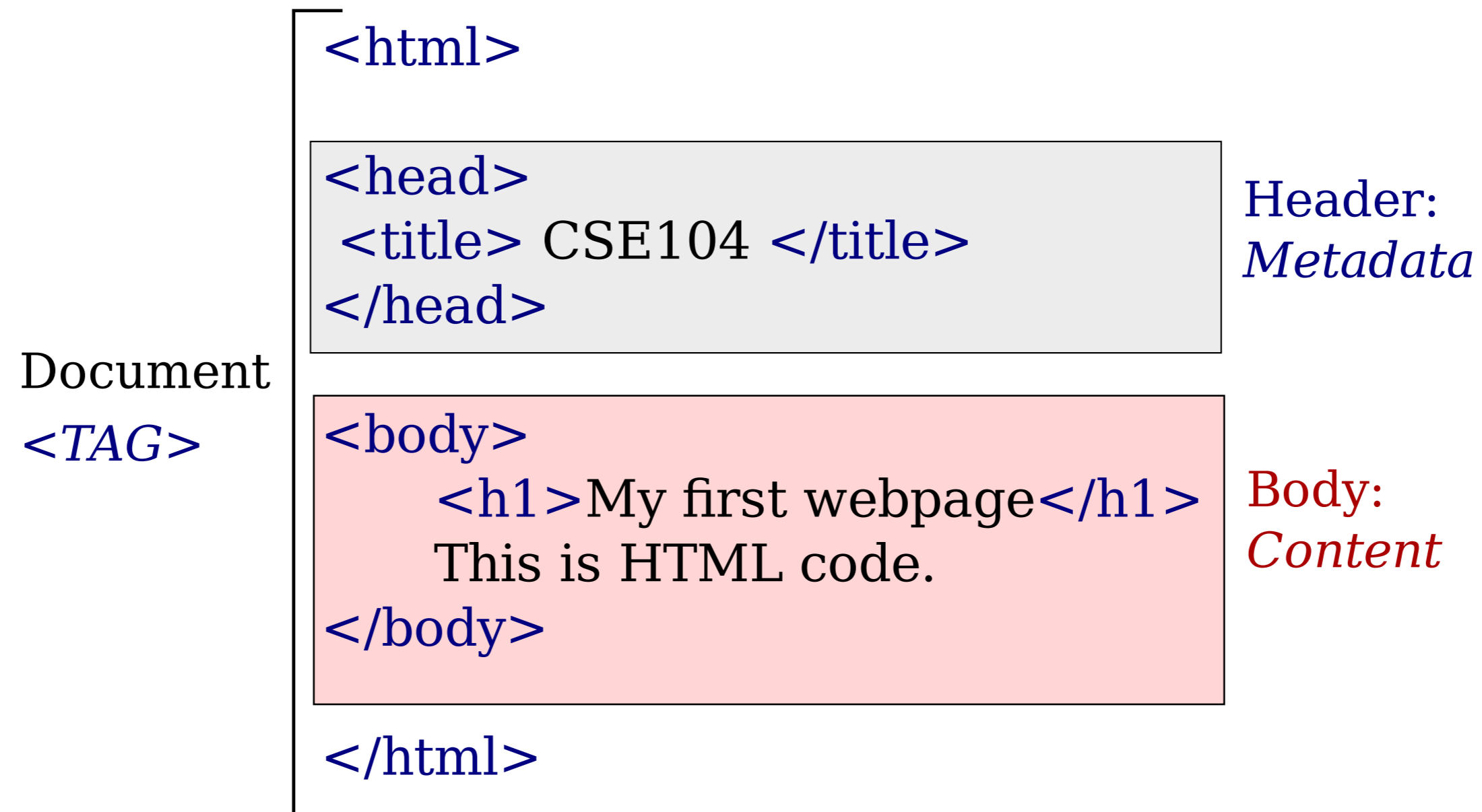
<head>
  <title> CSE104 </title>
</head>

<body>
  <h1>My first webpage</h1>
  This is HTML code.
</body>

</html>
```

# Example of HTML file

HTML5 tag `<!DOCTYPE html>`



# Display an HTML webpage in the Browser

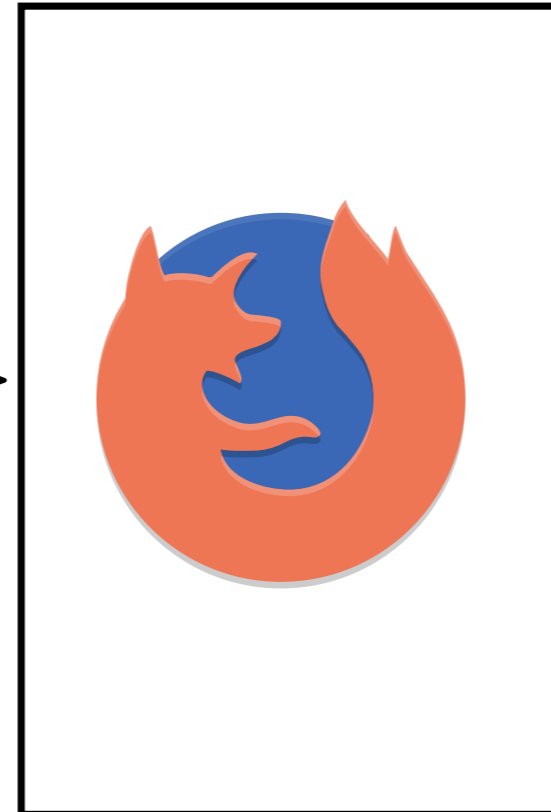
HTML file (text)

```
<!DOCTYPE html>
<html>
<head>
<title> CSE104 </title>
</head>
<body>
  <h1>My first webpage</h1>
  I am learning HTML
</body>
</html>
```

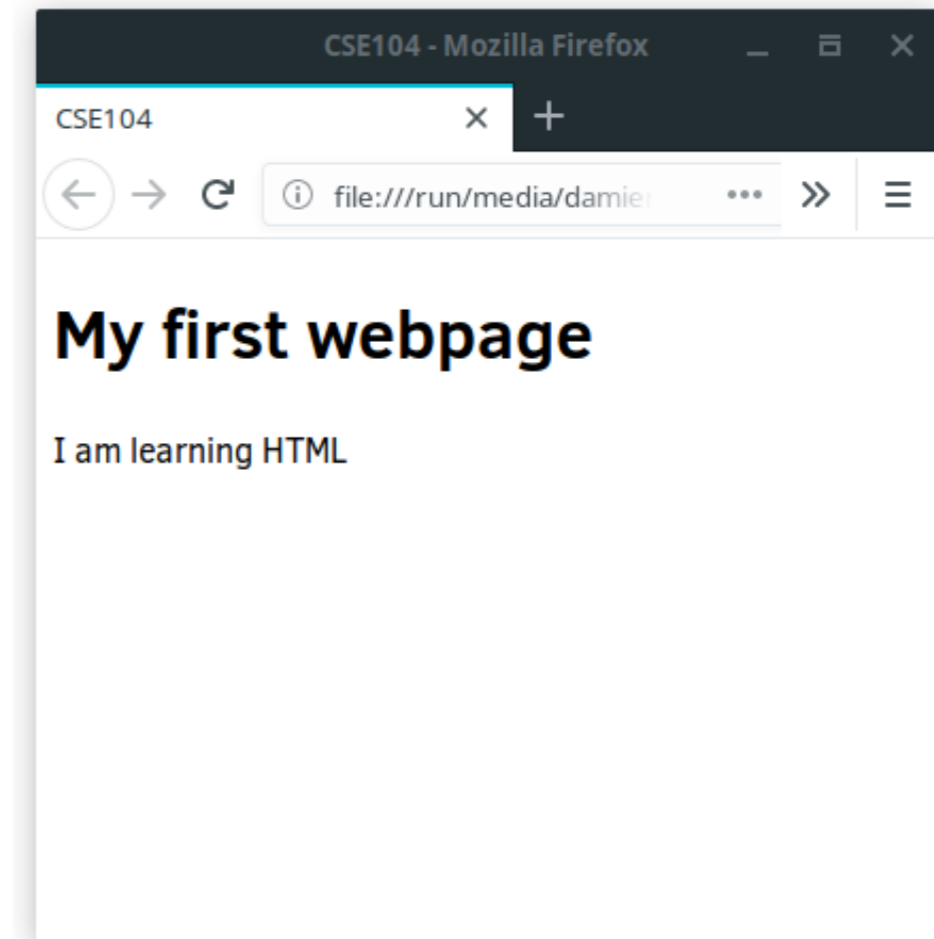
Read



Browser



Render



HTML code is **parsed**, **analysed**, and **rendered** by the browser

*Browser interpret HTML code (similar to Python interpreter VS Python code)*

*Browser also perform rendering of the result webpage*

# Web Browser

A web browser is a software that can

- Access **ressources** from the internet (distant servers using internet protocols)
- Parse and render these ressources

**Ressources** are commonly **HTML files**

*But can also be images, videos, etc.*

Examples of Browser

Mozilla Firefox, Google Chrome, Chromium, Edge, Safari, etc.

# Quick history of HTML

1990 - Creation of the *World Wide Web*

Formatted text document using HTML tags, embedding links to navigate between web pages

Several evolutions

- 1995 HTML 2
- 1997 HTML 3,4
- 2000 XHTML
- 2007 HTML 5

Today - HTML 5 as a *Living Standard*

- Updated by the [WHATWG](#)  
**Web Hypertext Application Technology Working Group**
- HTML 5 → **HTML**

# HTML - a programming language ?

HTML is a textual way to describe the **structure** and **content** of a document

- Parts: header, navigation, body, footer
- Titles, subtitles, etc.
- Type of elements (image, video, code, etc.)
- Relative importance of text element (emphasize, strong, un-relevant, etc.)

HTML is **not** a programming language able to automatize tasks (*wait for JavaScript*)

- No loops, if, include, etc.

HTML(5) : describe the **semantic** of the document, not its appearance (it will be CSS)

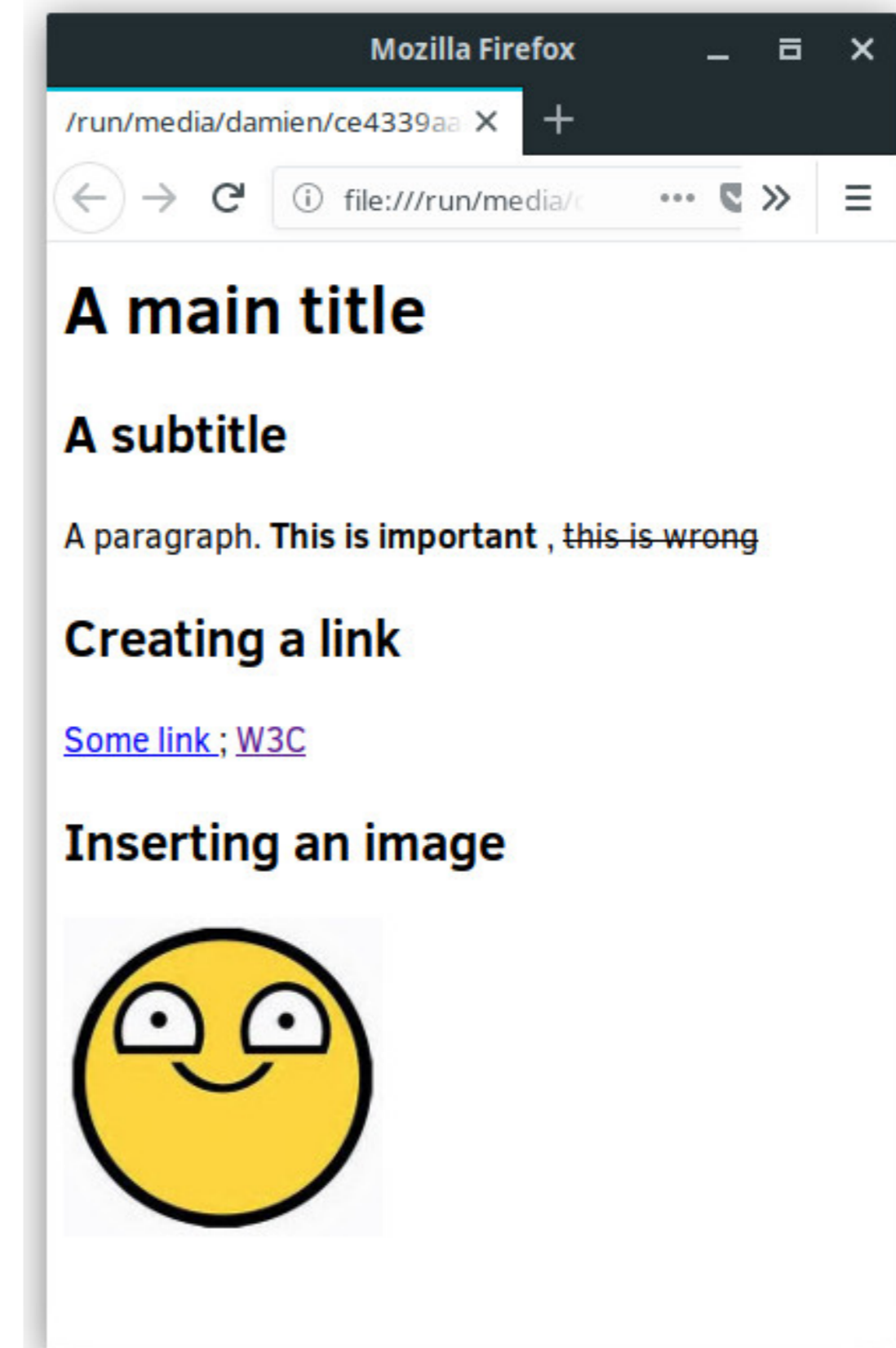


# HTML example

```
<h1> A main title </h1>
<h2> A subtitle </h2>
<p>
  A paragraph.
  <strong> Important to know </strong>,
  <s>this is wrong</s>
</p>
<h2> Creating a link </h2>
<a href="www.google.com"> Some link </a>
<a href="https://www.w3.org/"> W3C </a>
<h2> Inserting an image </h2>

```

*HTML elements have special default visual appearance*



# Independence of line breaks and spaces

```
<h1> A main
title </h1>

<h2>
A                subtitle
</h2>
<p>
A paragraph.
<strong> Important

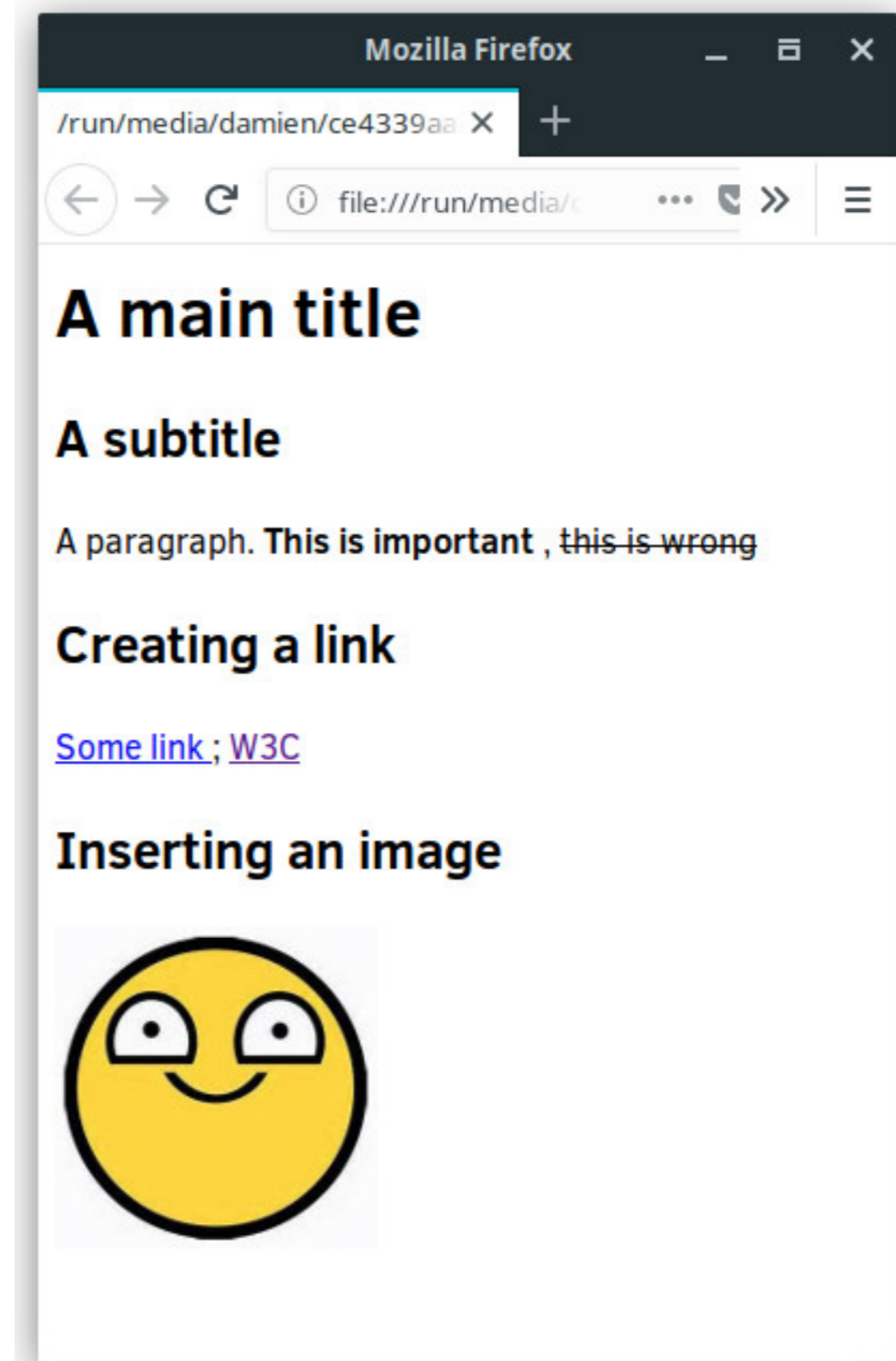
to know
  </strong>, <s>this is wrong</s>
</p>

<h3> Creating a link </h3>
<a href="www.google.com"> Some link </a>
<a href="https://www.w3.org/"> W3C </a>

<h3> Inserting an image </h3>

```

*Same visual appearance*

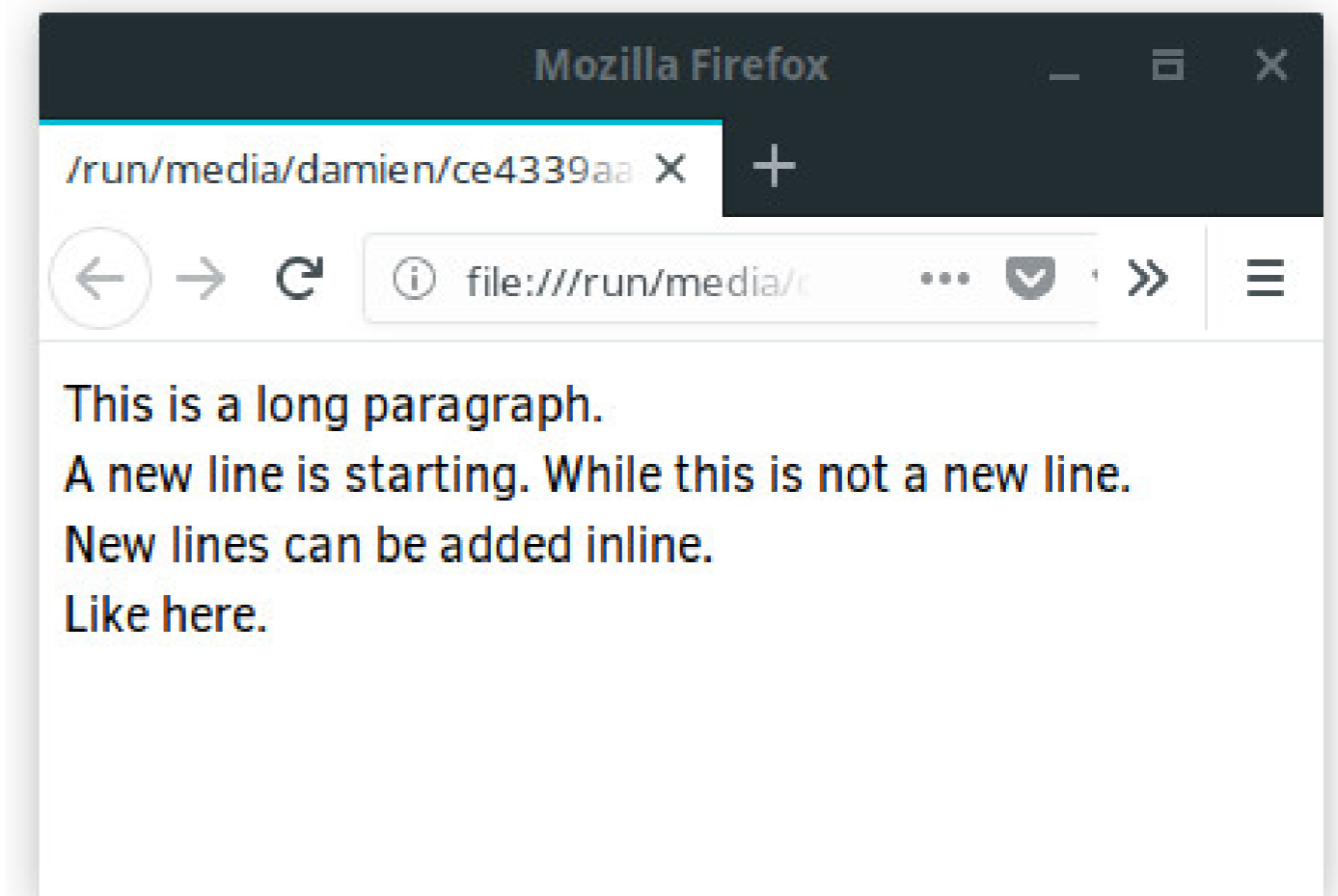


# Manual line break

Specific tag: **< br >**

```
<p>  
This is a long paragraph. <br>  
A new line is starting.  
While this is not a new line. <br>  
  
New lines can be added inline. <br> Like here.  
</p>
```

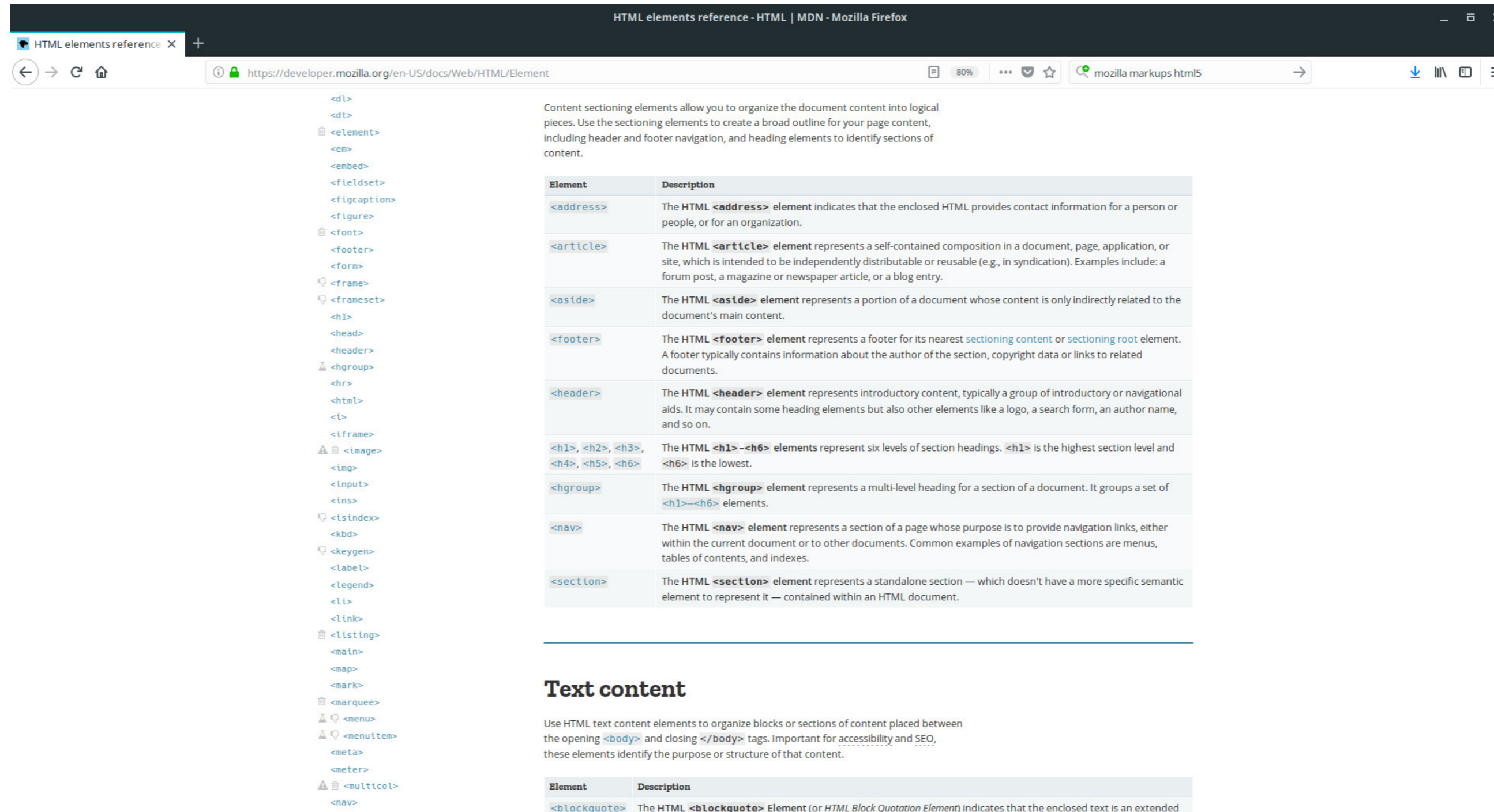
**<p>** indicates semantically a paragraph of text.



# Main HTML Tags

# Tags

Many different **HTML tags** exists



The screenshot shows the MDN HTML elements reference page. On the left is a scrollable list of HTML tags. The main content area features a section on 'Content sectioning elements' with a table of elements and their descriptions. Below this is a section on 'Text content' with a table of elements and their descriptions.

Content sectioning elements allow you to organize the document content into logical pieces. Use the sectioning elements to create a broad outline for your page content, including header and footer navigation, and heading elements to identify sections of content.

Element	Description
<code>&lt;address&gt;</code>	The HTML <b>&lt;address&gt;</b> element indicates that the enclosed HTML provides contact information for a person or people, or for an organization.
<code>&lt;article&gt;</code>	The HTML <b>&lt;article&gt;</b> element represents a self-contained composition in a document, page, application, or site, which is intended to be independently distributable or reusable (e.g., in syndication). Examples include: a forum post, a magazine or newspaper article, or a blog entry.
<code>&lt;aside&gt;</code>	The HTML <b>&lt;aside&gt;</b> element represents a portion of a document whose content is only indirectly related to the document's main content.
<code>&lt;footer&gt;</code>	The HTML <b>&lt;footer&gt;</b> element represents a footer for its nearest <b>sectioning content</b> or <b>sectioning root</b> element. A footer typically contains information about the author of the section, copyright data or links to related documents.
<code>&lt;header&gt;</code>	The HTML <b>&lt;header&gt;</b> element represents introductory content, typically a group of introductory or navigational aids. It may contain some heading elements but also other elements like a logo, a search form, an author name, and so on.
<code>&lt;h1&gt;</code> , <code>&lt;h2&gt;</code> , <code>&lt;h3&gt;</code> , <code>&lt;h4&gt;</code> , <code>&lt;h5&gt;</code> , <code>&lt;h6&gt;</code>	The HTML <b>&lt;h1&gt;</b> - <b>&lt;h6&gt;</b> elements represent six levels of section headings. <code>&lt;h1&gt;</code> is the highest section level and <code>&lt;h6&gt;</code> is the lowest.
<code>&lt;hgroup&gt;</code>	The HTML <b>&lt;hgroup&gt;</b> element represents a multi-level heading for a section of a document. It groups a set of <code>&lt;h1&gt;</code> - <code>&lt;h6&gt;</code> elements.
<code>&lt;nav&gt;</code>	The HTML <b>&lt;nav&gt;</b> element represents a section of a page whose purpose is to provide navigation links, either within the current document or to other documents. Common examples of navigation sections are menus, tables of contents, and indexes.
<code>&lt;section&gt;</code>	The HTML <b>&lt;section&gt;</b> element represents a standalone section — which doesn't have a more specific semantic element to represent it — contained within an HTML document.

**Text content**

Use HTML text content elements to organize blocks or sections of content placed between the opening `<body>` and closing `</body>` tags. Important for accessibility and SEO, these elements identify the purpose or structure of that content.

Element	Description
<code>&lt;blockquote&gt;</code>	The HTML <b>&lt;blockquote&gt;</b> Element (or <i>HTML Block Quotation Element</i> ) indicates that the enclosed text is an extended

# Main tags

- **<h1>**, **<h2>**, ..., **<h6>**: titles, subtitles, etc.
- **<a >**: Link (href="url") - *can be distant url (https://...), or local html file*
- **<img >**: Image (src="url")
- **<p >**: Paragraph

# Main tags - lists

## Unordered list of elements

```
<ul>
<li> First </li>
<li> Second </li>
<li> Third </li>
</ul>
```

- First
- Second
- Third

## Ordered list of elements

```
<ol>
<li> First </li>
<li> Second </li>
<li> Third </li>
</ol>
```

1. First
2. Second
3. Third

## Nested lists

```
<ul>
  <li> First </li>
  <li>
    <ul>
      <li> Subpart one </li>
      <li>
        <ol>
          <li> a </li>
          <li> b </li>
          <li> c </li>
        </ol>
      </li>
      <li> Subpart two </li>
    </ul>
  </li>
  <li> Second </li>
  <li> Third </li>
</ul>
```

- First
  - Subpart one
    1. a
    2. b
    3. c
  - Subpart two
- Second
- Third

# Good Practices (general rules)

- Use tags for their **semantic meaning**, not their default appearance
- Check tags documentation
  - <https://developer.mozilla.org/en-US/docs/Web/HTML/Element>
  - <https://www.w3schools.com/tags/>



# Metadata

- Tags which are not visualized may serve as metadata

*titles, favicon, Keywords, Description, Author, Encoding, etc.*

- Metadata are mostly found in the <head> section
- Metadata are used by search-engine

*Clean and accurate meta-data improve automatic analysis of websites (and indexation)*

```
<!DOCTYPE html>
<html>
<head>
  <title> TITLE </title>
  <meta name="author" content="">
  <meta name="keywords" XXX >
  <meta name="description" YYY >
  <meta name="icon" ZZZ >
</head>
<body>...</body>
</html>
```

# Special tag

- **<!DOCTYPE html>** indicates the start of HTML(5) document
- You may also see
  - XHTML : `<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">`
  - HTML4 : `<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">`

# Good practice: Head metadata

You should use the minimal following template for your webpage

```
<!DOCTYPE html>

<html lang="en">

<head>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title> Your title </title>
</head>

<body>...</body>

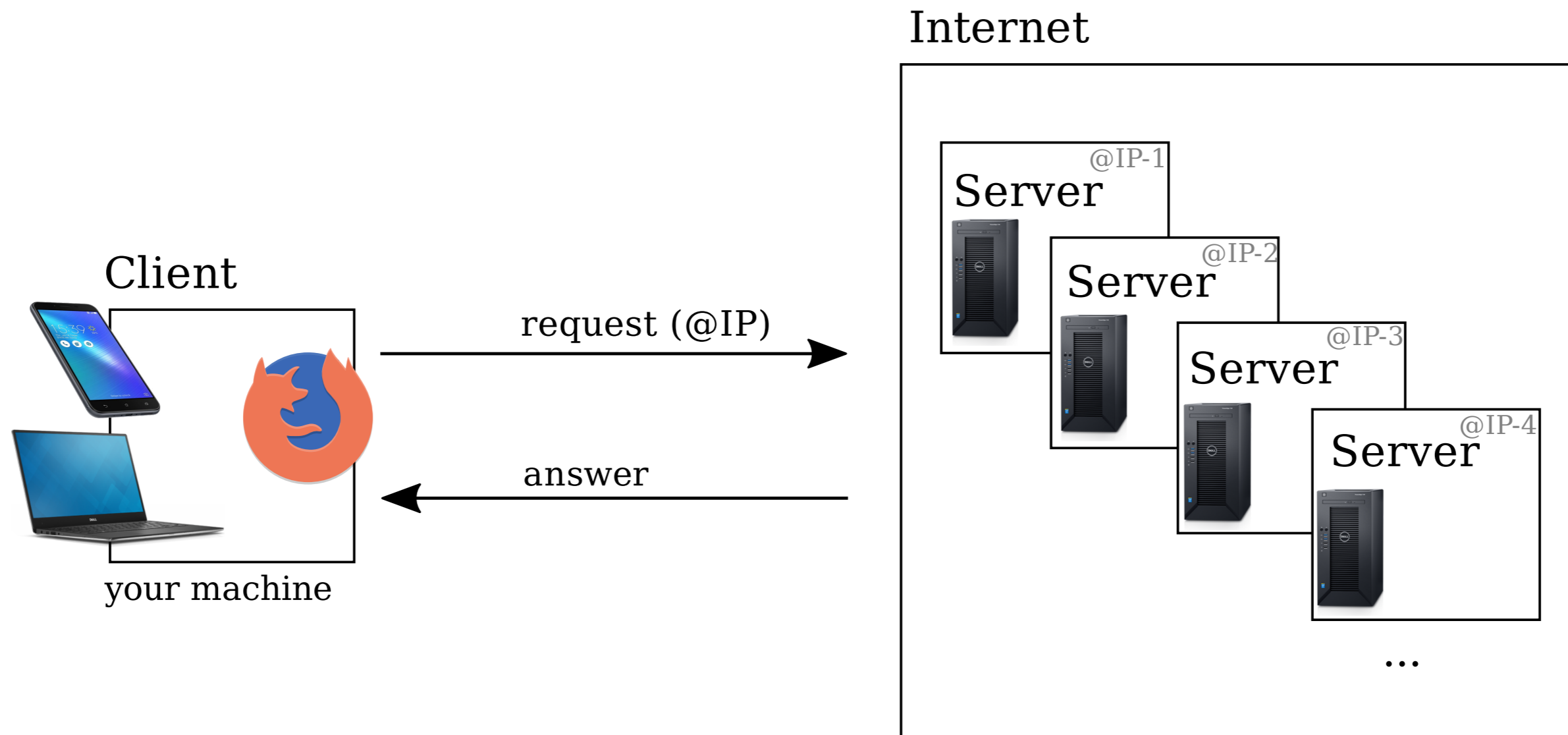
</html>
```

- **<meta charset="utf-8">** : Text encoding (prefer utf-8 to other)  
Text encoding = how letters are encoded in binary  
If you see characters like ãfâ©â¿ ♦ ⇒ problem with encoding  
*Non ASCII characters (ex. accent letters) may be encoded in different ways*
- **<meta name="viewport" content="width=device-width, initial-scale=1">** Set scale to fit screen width  
Used to get correct rendering in smartphone

# Browser, HTML and internet

# What happen when you visit a website

- Client/server communication
  - Client: Your computer/smartphone
  - Server: A distant computer hosting the data
- Communication using protocols (HTTP, FTP, etc)  
*(See networks)*



# Loading a webpage

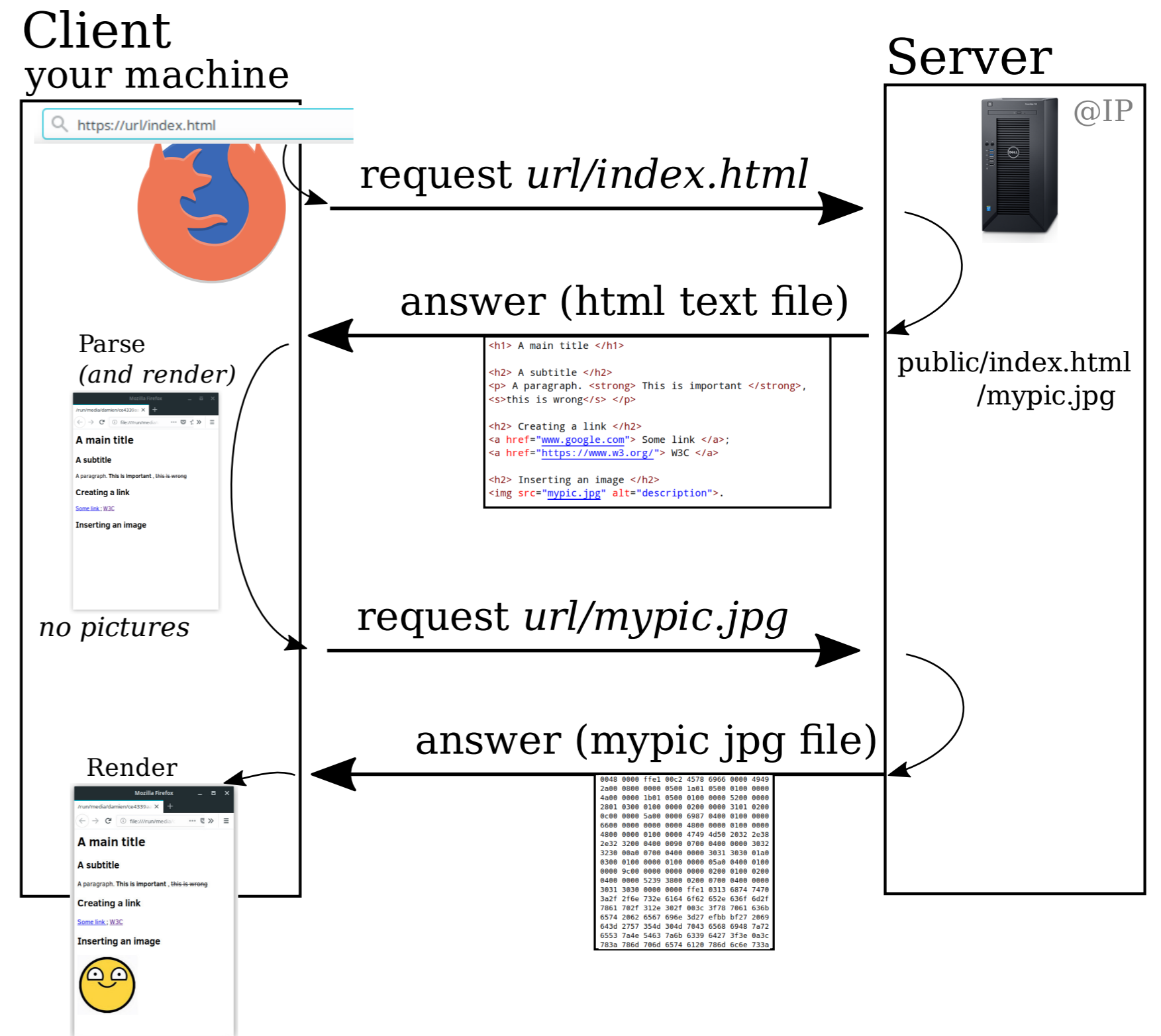
Loading Webpage = Several requests/answers steps

- First request HTML text
- Then request all resources separately (img)  
*(text may appear before images)*

Group of webpages on the same site is called Website

## Terminology

- Front-end = client size  
*appearance, local user interaction*
- Back-end = server size  
*storage, data-base, service*
- "Full-stack": client and server programming



# Browser and incorrect HTML

Many possibilities to write incorrect HTML

- *Forget to close tags* (<ul> <li> <li>)
- *Embedding incompatible tags* (<p><h1></h1></p>)
- *Typos* (<ui></ul>)

Browsers are made to be extremely robust to wrong HTML

- *Try to correct your HTML and give visual result*
- *Try to adapt to non standard practices*
- *Try to be convenient to non expert developers*

⇒ Silent correction : a good looking page may have wrong HTML

⇒ **Warning:** Incorrect/non standard HTML may

- work on one browser, but not on another one
- work today, but not in a future version

As a Computer Scientist: Should **check the validity** of your HTML by external tools.

# HTML Validator

## W3C Markup Validator

<https://validator.w3.org/>

Good practice:

- Check often your code in the validator
- **Always check before final render of your website/project**

### Nu Html Checker

This tool is an ongoing experiment in better HTML checking, and its behavior

Showing results for contents of text-input area

Checker Input

Show  source  outline  image report Options...

Check by   css

```
<!doctype html>
<head>
  <title> CSE104 </title>
</head>

<body>
  <strong> Some text </slrong>
</body>
</html>
```

Check

Use the Message Filtering button below to hide/show particular messages,

Message Filtering

1. **Error** Stray end tag `<slrong>`.  
From line 7, column 28; to line 7, column 36  
Some text `</slrong>`

### Nu Html Checker

This tool is an ongoing experiment in better HTML checking, and its behavior

Showing results for contents of text-input area

Checker Input

Show  source  outline  image report Options...

Check by   css

```
<!doctype html>
<head>
  <title> CSE104 </title>
</head>

<body>
  <strong> Some text </strong>
</body>
</html>
```

Check

Use the Message Filtering button below to hide/show particular messages,

Message Filtering

Document checking completed. No errors or warnings to show.



# CSS to set webpage appearance

# CSS

## Cascading **S**tyle **S**heet

Language describing the appearance

Focus on the appearance: not the structure/semantic

*Orthogonal to HTML*

# CSS Example

## HTML

```
<body>
<h1>My title</h1>
<p> A paragraph with
<strong>special</strong> word. </p>
</body>
```

## CSS 1

```
h1 {
  color: salmon;
  text-align: center;
  text-decoration: underline;
}
p {
  color: blue;
}
strong {
  border: 3px solid red;
}
```

My title

A paragraph with special word.

## CSS 2

```
body {
  background-color: black;
  text-align: right;
}
h1 {
  color: yellow;
  font-variant: small-caps;
}
p {
  color: cyan;
}
strong {
  font-size: 200%;
  color: orange;
}
```

MY TITLE

A paragraph with **special** word.

# Associate CSS file to HTML

Add to the HTML header

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

*"style.css": pathname of the CSS file*

index.html

```
<!DOCTYPE html>
<html>

<head>
| | | | <title> CSE104 </title>
| | | | <link rel="stylesheet" href="style.css">
</head>

<body>
| | | | <h1>Content of Webpage</h1>
</body>

</html>
```

style.css

```
h1 {
| background-color: ■blue;
}
```

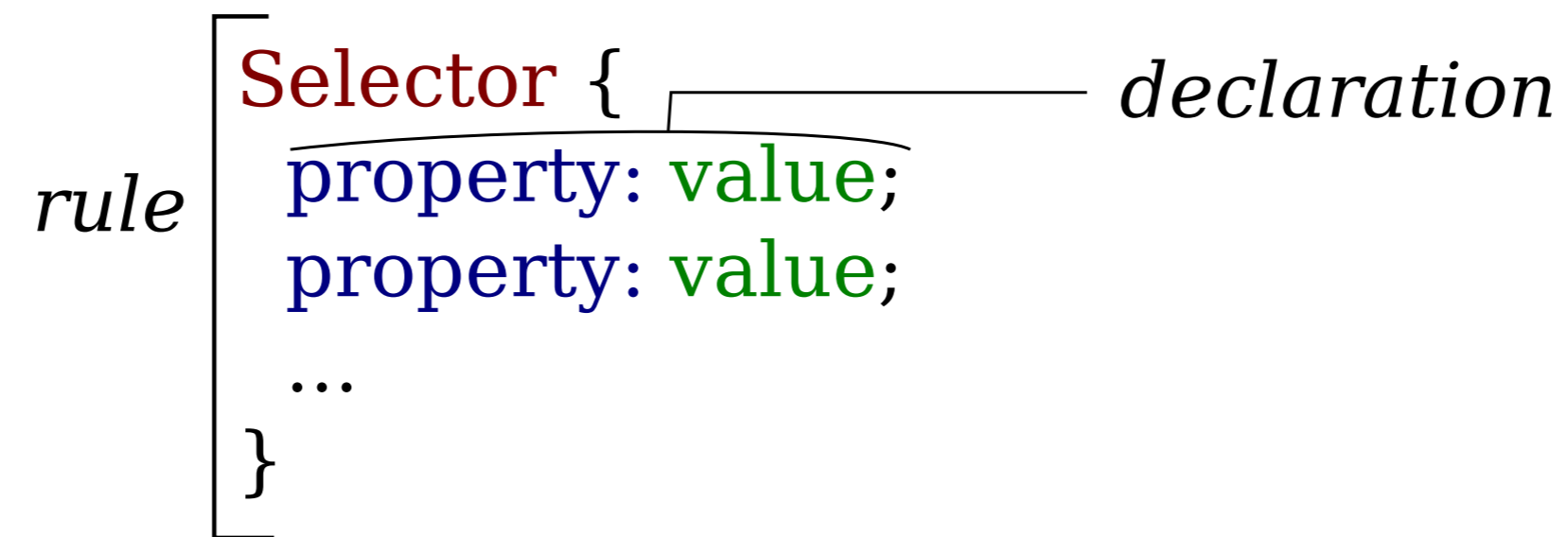
Note: Can associate multiple CSS files to a single HTML

```
<link rel="stylesheet" type="text/css" href="file_1.css">
```

```
<link rel="stylesheet" type="text/css" href="file_2.css">
```

etc.

# CSS rule structure



## Example

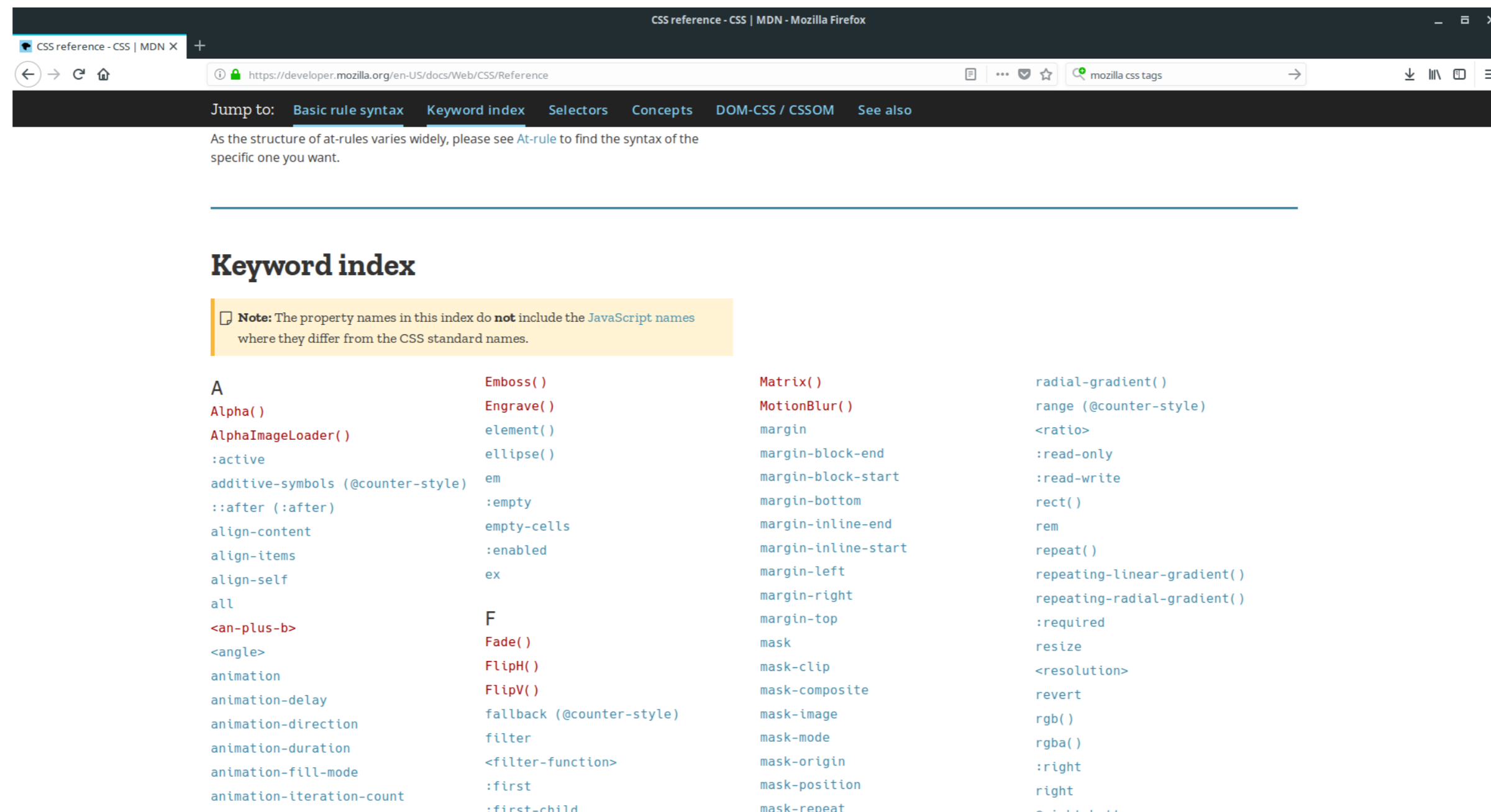
```
h1 {  
    background-color: yellow;  
    text-align: right;  
}  
  
h2 {  
    margin: 25px;  
    font-size: 150%;  
}
```

# CSS tags

Many tags

<https://developer.mozilla.org/en-US/docs/Web/CSS/Reference>

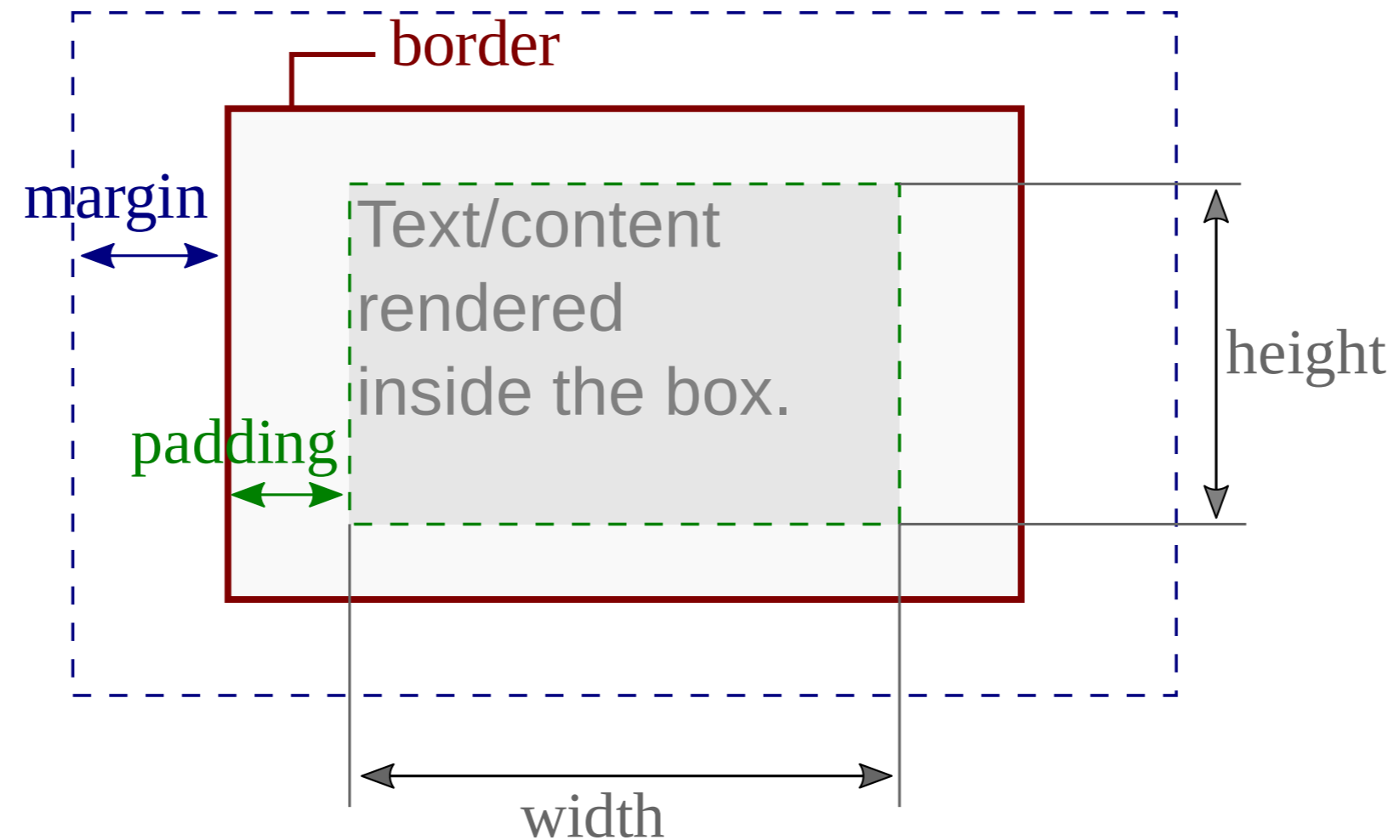
<https://www.w3schools.com/cssref/>



# Box model

# Box model

Each HTML element (between opening/closing tag) is a rectangular block



- Set in CSS: Margin, padding, border, width, height
- Can separate: left, right, top, down



# Boxes in webpage

Webpages contain several boxes

Boxes may include other boxes (cascading behavior)

*elements have relation of parenting (parent/child), and siblings.*

**<h1> this is <em> important </em> </h1>**

*<em> element **is a child** of <h1> element.*

*<h1> element is the parent of <em> element.*

**<body> <h1> this is <em> important </em> </h1>**

**</body>**

*<body> is the parent of <h1> and grand-parent (so it is "a parent") of <em>.*

**<h1> this is </h1> <em> important </em>**

*<em> element is **not** a child of <h1> element, only siblings.*

**<a href="https://google.com" > click here <a>**

*href is **not** a child of <a> element. It is parameter.*

**A main title**

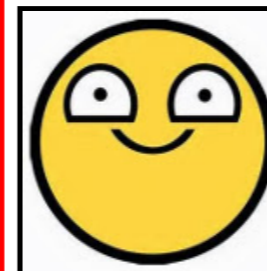
**A subtitle**

A paragraph. **This is important**, ~~this is wrong~~

**Creating a link**

[Some link](#); [W3C](#)

**Inserting an image**



# Display

- Elements may have different layout behavior
  - Flow one after the other
    - No fixed size: *ex. words and emphasized words (italic, bold, etc)*

Words are flowing left to right

- Fixed size: *ex. pictures*



- Span the entire line *ex. titles*

**Titles span a full line**

Other title go to the next line

- Display values: **inline, inline-block, block**

# Block

## CSS

```
div {  
  display: block;  
  background-color: yellow;  
  margin: 10px;  
  padding: 30px;  
  border: 2px solid black;  
}
```

## HTML

```
<div>Block element</div>  
<div>Block element</div>
```

Block element

Block element

- Block elements span the entire line
- Default behavior for titles

# Inline-Block

## CSS

```
div {  
  display: inline-block;  
  background-color: yellow;  
  margin: 10px;  
  padding: 30px;  
  border: 2px solid black;  
}
```

## HTML

```
<div>Inline Block element</div>  
<div>Inline Block element</div>
```

Inline Block element

Inline Block element

- Inline-block size can be fully set by the user
- Typically used for image-like elements

# Inline

## CSS

```
div {  
  display: inline;  
  background-color: yellow;  
  margin: 10px;  
  padding: 5px;  
  border: 2px solid black;  
}
```

Inline element

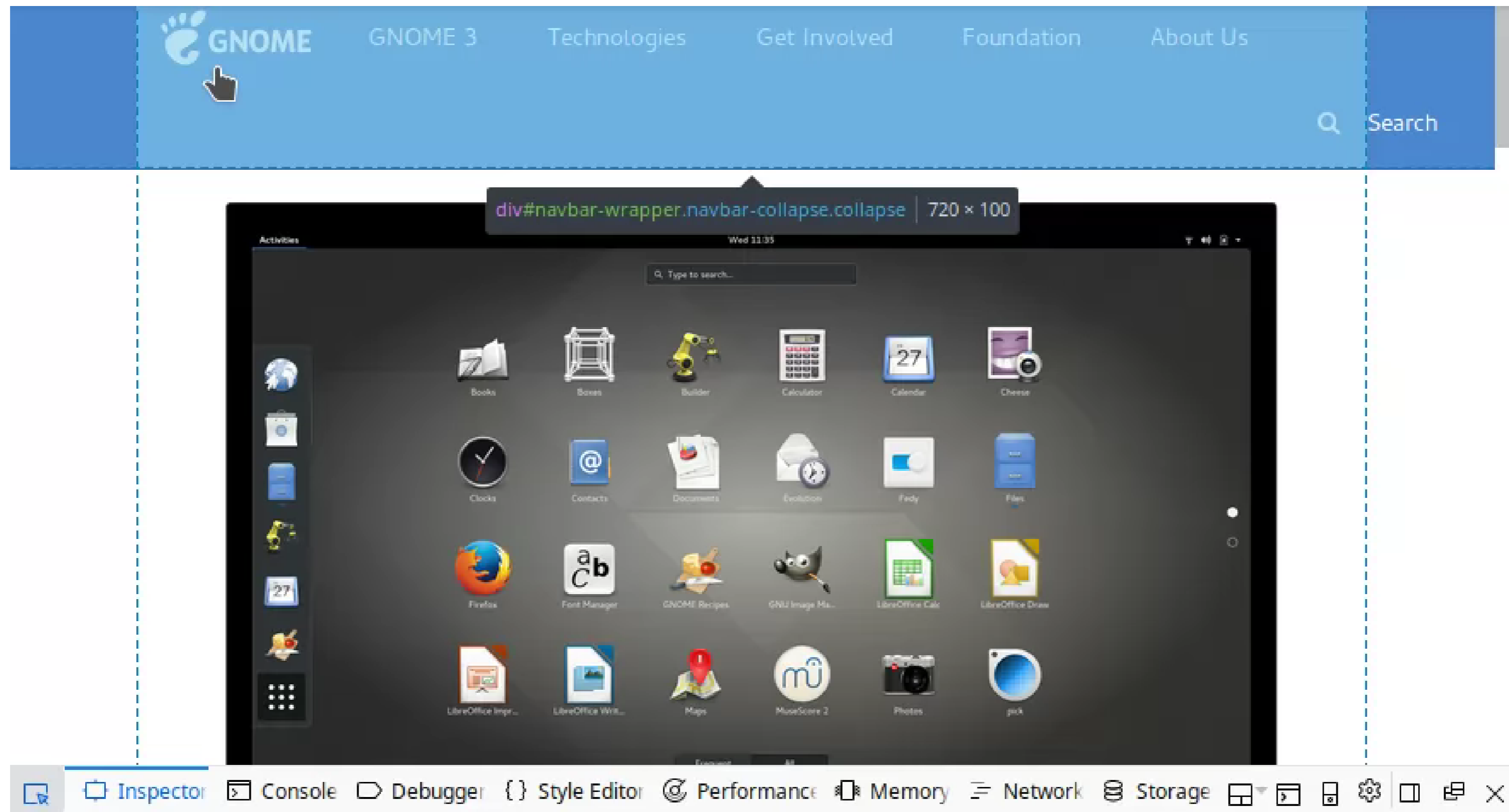
Inline element

## HTML

```
<div>Inline element</div>  
<div>Inline element</div>
```

- Inline elements flow one after the other
- Left/right margin/padding can be set by the user
- Top/Down margin/padding are ignored
- Default behavior for text elements (strong, emphasized, etc.)

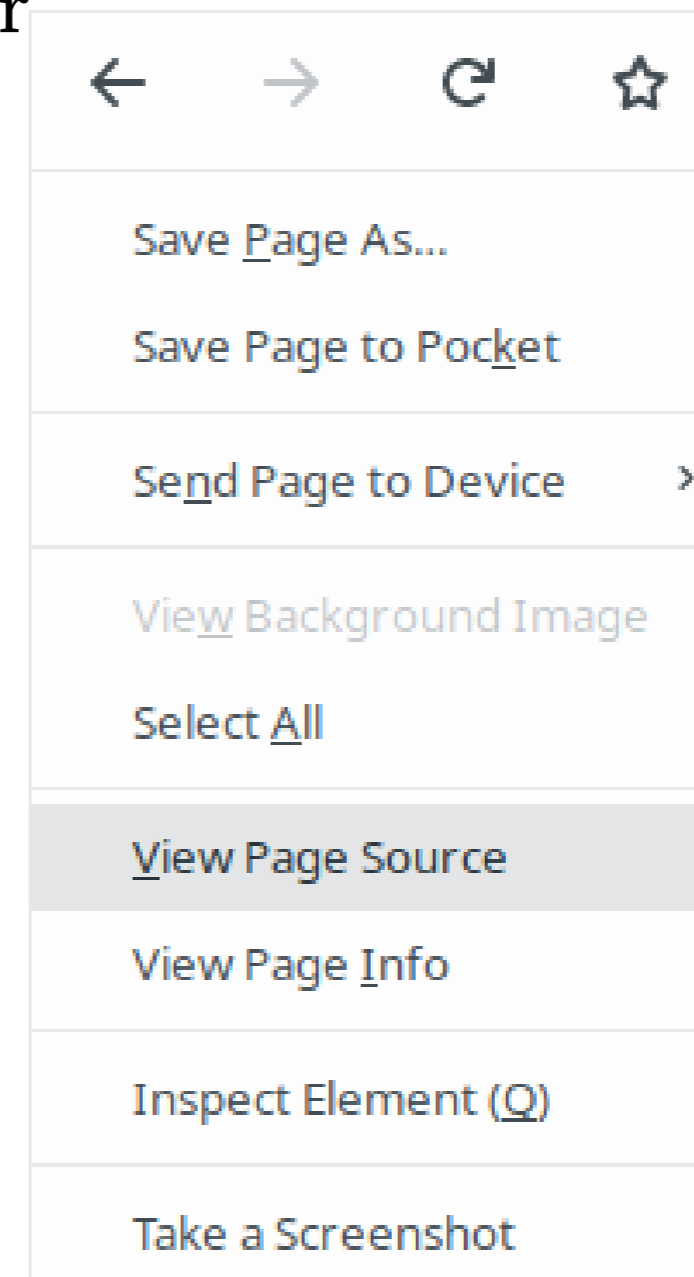
# Boxes on a real website



# Source and debug tools

# Source code

- HTML and CSS *open source* by design
- Don't hesitate to check source code of websites
  - In your browser **view-source:URL** in the address bar
  - Or graphically





# Browser debug and analysis tool

- Browser have embedded debugging tools (developer mode)
- Firefox, Chrome : access using F12 key

