Markup

GERARD ROMA 2021

Markup languages

- Markup languages are **formal** languages but different from programming languages:
  - They are **declarative**, they describe how the data should look, without specifying how to do that.
  - They need to be understood by another program (using some programming language).
  - They focus on the structure or presentation of data.
- Markup languages may be used by either humans or machines.

Examples

- HTML
- CSS
- XML
- SVG
- JSON
- YAML
- Markdown

Uses

- Formatting text documents
- Describing Graphical User Interfaces
- Configuration
- Transmitting information (e.g. Web APIs)

HTML

- Hypertext Markup Language: formatted text + links
- HTML is used to describe the **structure** of content.
- Importantly, it should not be used to describe **appearance**.

HTML Elements

```
<tag attr1="val1" attr2=“val2">contents</tag>
```

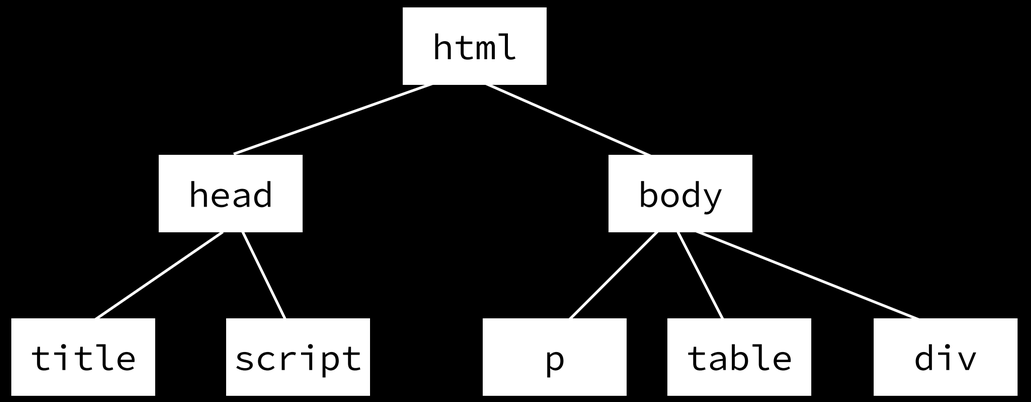
Self-closing tag (only for **void** elements):

```
<tag attr="val" attr2="val">
```

- Notes:
  - Attributes may be optional or mandatory
  - XHTML was an attempt to “fix” HTML4, especially for self-closing tags, but it failed

Document Structure

- Elements form a hierarchical structure.
- The HEAD includes information about the document (metadata).
- The BODY includes the content of the document.



Common elements

- Headings: <h1>, <h2> … <h6>
- Paragraphs: <p>,
- Lists: <ul>, <ol>,
- Anchor: <a>
- Style: ,
- Forms: <form>, <input>
- Text style: ,

Good HTML practices

- HTML has evolved along with CSS and JavaScript. Best practice is to use each language for its purpose.
  - Avoid layout / appearance attributes (width, height, border).
  - Avoid inline **style** attribute.
  - Add additional / alternative information for multiple clients and accessibility.

HTML example: tables

```
<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>
```

Source: w3schools.com

HTML example: forms

```
<form action="" method="get" class="form-example">
  <div class="form-example">
    <label for="name">Enter your name: </label>
    <input type="text" name="name" id="name" required>
  </div>
  <div class="form-example">
    <label for="email">Enter your email: </label>
    <input type="email" name="email" id="email" required>
  </div>
  <div class="form-example">
    <input type="submit" value="Subscribe!">
  </div>
</form>
```

Source: https://developer.mozilla.org

HTML example: boilerplate

```
<!doctype html>
<html class="no-js" lang="">

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

  <meta property="og:title" content="">
  <meta property="og:type" content="">
  <meta property="og:url" content="">
  <meta property="og:image" content="">

  <link rel="manifest" href="site.webmanifest">
  <link rel="apple-touch-icon" href="icon.png">
  <!-- Place favicon.ico in the root directory -->

  <link rel="stylesheet" href="css/normalize.css">
  <link rel="stylesheet" href="css/main.css">

  <meta name="theme-color" content="#fafafa">
</head>
```

Source: https://html5boilerplate.com

XML

- eXtensible Markup Language.
- Roughly same syntax as HTML (but strict).
- User-defined tags.
- Used to store and transmit information (typically in web services).

XML Example

```
<employees>
  <employee>
    <firstName>John</firstName> <lastName>Doe</lastName>
  </employee>
  <employee>
    <firstName>Anna</firstName> <lastName>Smith</lastName>
  </employee>
  <employee>
    <firstName>Peter</firstName> <lastName>Jones</lastName>
  </employee>
</employees>
```

Source: w3schools.com

XHTML

- HTML is **lenient** as a formal markup language, it accommodates user errors.
- XHTML was an attempt to transition to strict HTML following XML rules.
- It failed and delayed the inception of HTML5.

Modern and custom HTML

- After many years of stagnation, HTML5 introduced new tags and faster evolution.
- New APIs are introduced often. Examples: Web Audio, Sensor APIs.
- JavaScript frameworks tend to “augment” HTML by introducing custom names and attributes.
- Solid understanding of the syntax and main structure and a good reference documentation are essential.

JSON

- Javascript Object Notation.
- “Discovered in nature” by D.Crockford
- Same uses as XML, but more convenient.
- Uses JavaScript syntax, nowadays used beyond JavaScript.
- Example: VSCode configuration.

Dictionary

dictionary | ˈdɪkʃən(ə)ri |

noun (plural **dictionaries**)

a book or electronic resource that lists the words of a language (typically in alphabetical order) and gives their meaning, or gives the equivalent words in a different language, often also providing information about pronunciation, origin, and usage: *I'll look up 'love' in the dictionary* | *the website gives access to an online dictionary* | *[as modifier] : the dictionary definition of 'smile'.*

- a reference book on a particular subject, the items of which are typically arranged in alphabetical order: *a dictionary of quotations.*
- *Computing* a set of words or other text strings made for use in applications such as spellcheckers: *the worm attempts to crack account passwords using a built-in dictionary.*

- Data structure: find any information via a keyword (string) or other unique objects.

JSON example: XML comparison

```
{"employees":[
  { "firstName":"John", "lastName":"Doe" },
  { "firstName":"Anna", "lastName":"Smith" },
  { "firstName":"Peter", "lastName":"Jones" }
]}
```

Source: w3schools.com

JSON example: VSCode user settings


```
{
    "workbench.colorTheme": "Default Dark+",
    "security.workspace.trust.untrustedFiles": "open",
    "editor.fontSize": 18,
    "python.defaultInterpreterPath": "/Users/gerard/opt/anaconda3/bin/python",
    "workbench.startupEditor": "none",
    "terminal.integrated.inheritEnv": false,
    "workbench.editor.untitled.hint": "hidden",
    "json.maxItemsComputed": 500000,
    "editor.minimap.enabled": false
}
```

Markdown

- Text-based syntax for easier writing of text with style.
- Can be converted to HTML.
- Used in Content Management Systems: Blogs, Wikis…
- Example: Gitlab Readme.

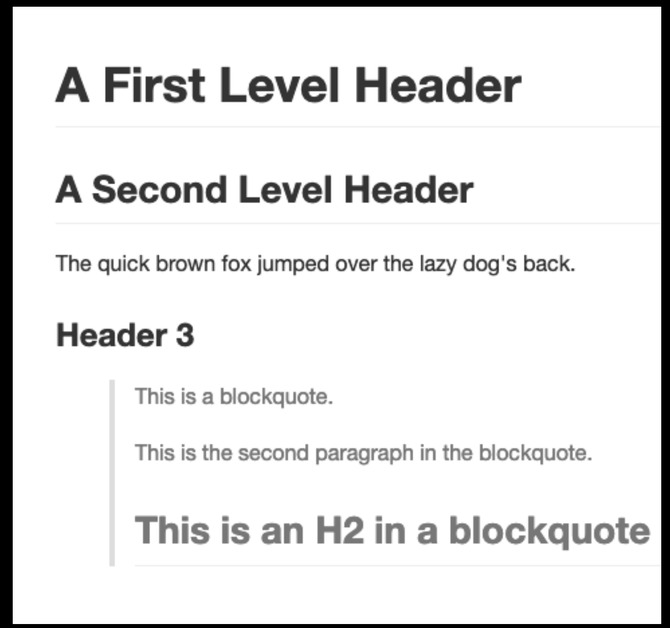
Markdown example

```
A First Level Header
====================

A Second Level Header
---------------------

The quick brown fox jumped over the lazy
dog's back.

### Header 3

> This is a blockquote.
>
> This is the second paragraph in the blockquote.
>
> ## This is an H2 in a blockquote
```

Markdown code

```
<h1 id="a-first-level-header">A First Level Header</h1>
<h2 id="a-second-level-header">A Second Level Header</h2>
<p>The quick brown fox jumped over the lazy
dog's back.</p>
<h3 id="header-3">Header 3</h3>
<blockquote>
<p>This is a blockquote.</p>
<p>This is the second paragraph in the blockquote.</p>
<h2 id="this-is-an-h2-in-a-blockquote">This is an H2 in a
blockquote</h2>
</blockquote>
```

Generated HTML



Rendered

Content Management Systems (CMS)

- One of the main applications of the Web is to share documents made of text and images.
- Creating new documents using plain HTML requires understanding of HTML, access to a server and other complications.
- Content Management Systems are applications with different levels of customisability which facilitate the creation of new content by users.
- Example: Wordpress
- Also: shops, forums, wikis…

Static Site Generators (SSG)

- Web applications, including CMSs, require a database and a scripting language running in a server exposed to the internet.This is a serious security risk which requires constant updates.
- Static Site Generators provide a solution by running the CMS locally, and generating a static HTML site.
- Mostly used by technical users.