INTRODUCTION TO WEB PROGRAMMING

Chap. 1 / HTML5 & CSS3, part 1

Anne Jeannin-Girardon, PhD | anne.jeannin@unistra.fr
Associate Professor, University of Strasbourg

Today's topics

- How to structure a webpage with HTML5
- Specifies how a webpage looks

First of all, the take home message of this course: structure & look and feel are separated (HTML / CSS)

About this course

- It is NOT an exhaustive list of HTML tags and CSS properties
- Basic and essential stuff will be presented...
- It's up to you to be proactive during your labs to find stuff you need that was not necessarily covered during classes

What is HTML ?

- HyperText Markup Language
- Language based on tags used to structure documents
- A structured document contains titles, paragraphs, tables, lists… (hierarchical / nested containers)
- … the structure has nothing to do with look & feel
- Web browsers interpret the tags to display the document (“Hey, this is a title ; and this is a paragraph”)

What do you have to remember?

- HTML is about describing the nature of elements (is it a title? A paragraph?) but does not focus on their look & feel or where they are displayed in the page
- Proper structuration is important, especially for accessibility reasons

More technically, HTML is

- A set of tags defined in HTML specification (there is a predefined set of tags, you cannot invent your own)
- `<tag>text</tag>`
- The pair *opening tag;closing tag* is an HTML element
- Some tags don't have a closing mate: `<tag/>`
- HTML elements can have attributes:

```
<tag attr="property">text</tag>
<tag attr="property"/>
```

A first HTML page

- Basic template for every HTML page
- All tags are hierarchically organized
- No compilation or anything: just save your document as index.html and open it with a web browser

index.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Our first page</title>
  </head>

  <body>
    <h1>Page heading</h1>
    <p>Some text in a paragraph</p>
  </body>
</html>
```

Let’s dive into the code (1)

- `DOCTYPE`: tell the browser which HTML version was used to create the document (6 HTML versions since 1991)
  - Simplified with HTML5 : <!DOCTYPE html>
  - V.S. <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd"> for HTML4 strict

index.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Our first page</title>
  </head>

  <body>
    <h1>Page heading</h1>
    <p>Some text in a paragraph</p>
  </body>
</html>
```

Let's dive into the code (2)

- `html`: document root (the document can be seen as a tree)
- `head`: meta-data about the document (title, style, scripts, meta-information)

index.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Our first page</title>
  </head>

  <body>
    <h1>Page heading</h1>
    <p>Some text in a paragraph</p>
  </body>
</html>
```

Let's dive into the code (3)

- `body`: document content (what's displayed to the user by the browser)
- `h1`: level 1 heading -- only one of these on a page
  - + 5 other headings: `h2` to `h6`
  - Used to hierarchize the document
- `<!-- comment -->` won't be displayed to the user

index.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Our first page</title>
  </head>

  <body>
    <h1>Page heading</h1>
    <p>Some text in a paragraph</p>
    <!-- This is a HTML comment -->
  </body>
</html>
```

Let's visualize the nested boxes

- If your HTML document is properly structured, you should always be able to draw nice boxes like this
- Tools can help you checking your structure: https://validator.w3.org/#validate_by_upload

index.html

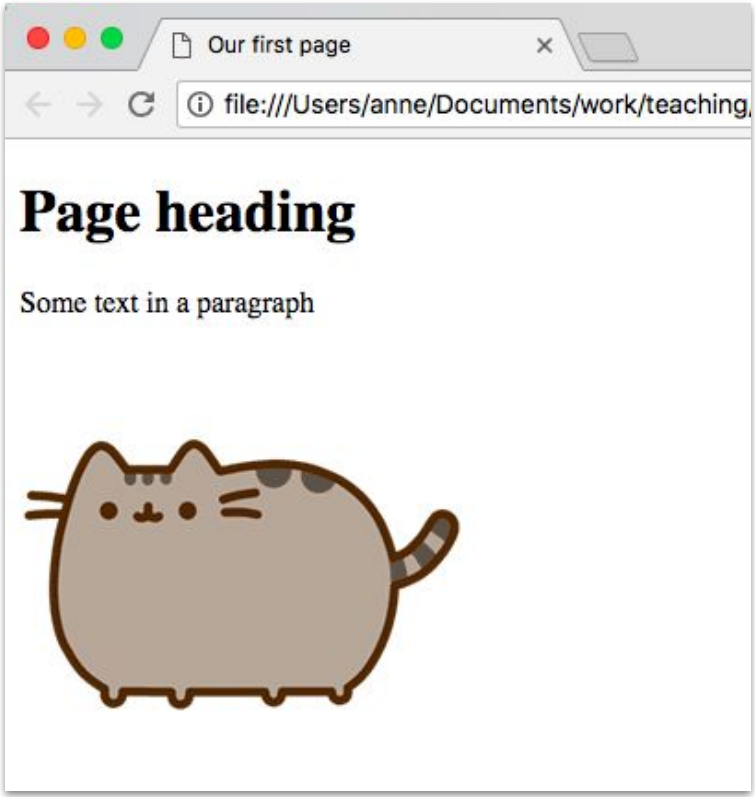
```
<!DOCTYPE html>
<html>
  <head>
    <title>Our first page</title>
  </head>

  <body>
    <h1>Page heading</h1>
    <p>Some text in a paragraph</p>
    <!-- This is a HTML comment -->
  </body>
</html>
```

HTML page example

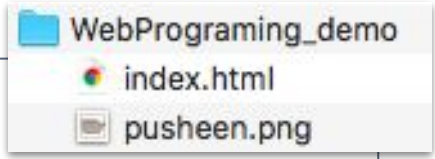
- Use your browser development tools to inspect this webpage: https://www.w3schools.com/css/demo_default.htm
  - For Chrome: menu View → Developer → Developer tools
  - For Firefox: menu Tools → Web developer → Toggle tools
  - For Safari: (1) activate the developer tools in the Preferences, Advanced Tab: check “Show Develop menu in menu bar” and (2) go to menu Develop → Show Web Inspector
  - For MS Internet Explorer / Edge: don’t use MS browser

Inserting images (1)



index.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Our first page</title>
  </head>
  <body>
    <!-- This is a HTML comment -->
    <h1>Page heading</h1>
    <p>Some text in a paragraph</p>
    <img src="pusheen.png" alt="Pusheen
     the cat"/>
  </body>
</html>
```



Source: http://www.pusheen.com/

Inserting images (2)

- `img`: HTML tag
- `src`: attribute to specify the source of the image (local or distant)
- `alt`: attribute to specify an alternative text (if the image doesn't display or if the page is read by a screen reader)
- The size of the image can be modified with the attributes **`width`** and **`height`**

index.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Our first page</title>
  </head>
  <body>
    <!-- This is a HTML comment -->
    <h1>Page heading</h1>
    <p>Some text in a paragraph</p>
    <img src="pusheen.png" alt="Pusheen
     the cat"/>
  </body>
</html>
```

Inserting links

- `<a href="another_page.html" title="Tooltip text">Click here</a>`
- `<a href="http://domain.ext">Link to an external page</a>`

Links to all kind of resources can be made (html page, image, audio, ...), as well as link to different protocols (ftp, mail, ...)

- `<a href="mailto:name@domain.com">Contact</a>`

Inserting lists

- `<ol>` ordered list
  - Number style can be modified (alpha, roman, …)
- `<ul>` unordered list (bullets)
  - Bullet style can be modified (circle, square, disc, …)
- `<li>` list item

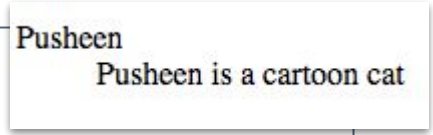
index.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Our first page</title>
  </head>
  <body>
    <h1>Page heading</h1>
    <ol>
      <li>Item 1</li>
      <li>Item 2</li>
    </ol>
  </body>
</html>
```

Inserting Definition Lists

index.html

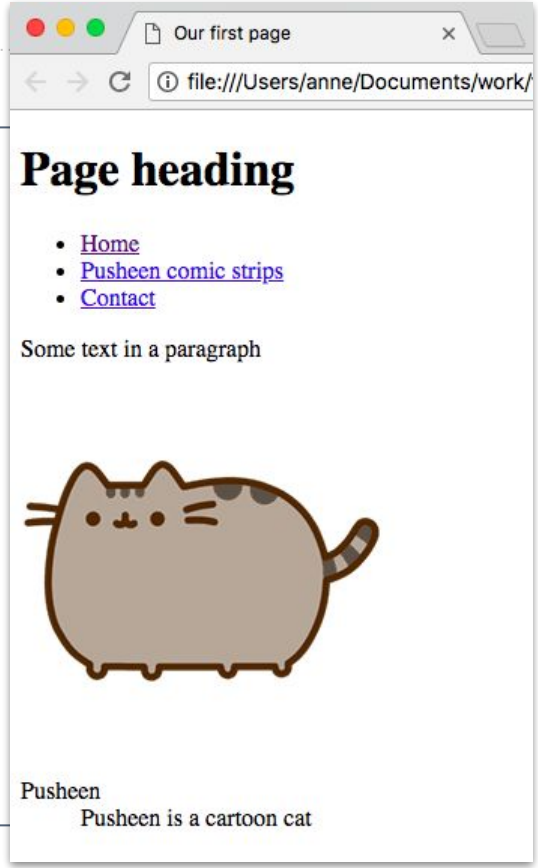
```
<!DOCTYPE html>
<html>
  <head>
    <title>Our first page</title>
  </head>
  <body>
    <h1>Page heading</h1>
    <dl> <!-- Definition List -->
      <dt>Pusheen</dt><!-- Definition Term -->
      <dd>Pusheen is a cartoon cat</dd> <!-- Definition Description -->
    </dl>
  </body>
</html>
```



Using a list for a navigation menu

index.html

```
<!DOCTYPE html>
<html>
  <head> <title>Our first page</title> </head>
  <body>
    <h1>Page heading</h1>
    <ul>
      <li><a href="index.html">Home</a></li>
      <li><a href="pusheen_comics.html">Pusheen comic strips</a></li>
      <li><a href="contact.html">Contact</a></li>
    </ul>
    <!-- etc -->
  </body>
</html>
```



Inserting tables

```
<h2>Web Programming course achievements</h2>
  <table>
    <tr> <!-- Table Row -->
      <th>Chapter</th> <!-- Table Header -->
      <th>Achievements</th>
    </tr>
    <tr>
      <td>HTML/CSS part 1</td> <!-- Table Data -->
      <td>Basic HTML document structure</td>
    </tr>
  </table>
```

HTML semantic for basic text formatting

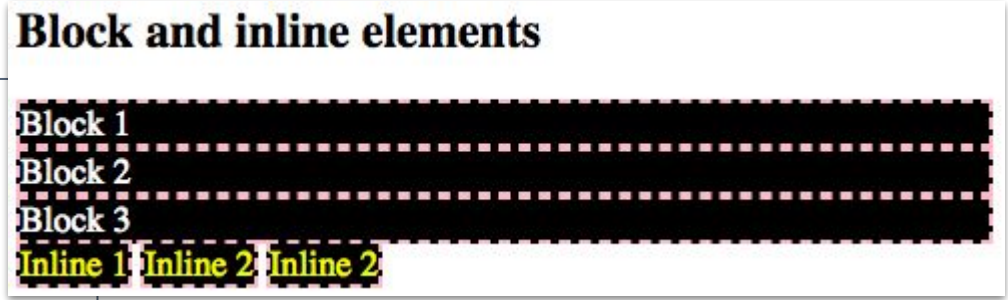
```
<h2>Formatting text with HTML semantic</h2>
<p>
  Text can have a <strong>strong importance</strong>. <br>
  Text can be <em>emphasized</em>. <br>
  Text can be <mark>marked</mark>. <br>
  Text can be <del>deleted</del>. <br>
</p>
```

HTML containers

- Block containers --- default = vertical display ; occupy the full width (stack of boxes)
  - List of block level element: https://developer.mozilla.org/en-US/docs/Web/HTML/Block-level_elements
  - Generic block:
- Inline elements --- default = horizontal display ; occupy only the necessary space
  - List of inline elements: https://developer.mozilla.org/en-US/docs/Web/HTML/Inline_elements
  - Generic inline:

HTML containers demo

```
<h2>Block and inline elements</h2>
  <div class="test_block">Block 1</div>
  <div class="test_block">Block 2</div>
  <div class="test_block">Block 3</div>
  <span class="test_inline">Inline 1</span>
  <span class="test_inline">Inline 2</span>
  <span class="test_inline">Inline 2</span>
```



Classes & identifiers

- `class`: can be used on multiple elements ; multiple classes can be used on the same element
- `id`: each element can have only 1 identifier ; each page can have only 1 element with that identifier

```
<div id="main_container">
  <p class="important">Some important text.</p>
  <p>Some regular text with an <span class="important">inline important
element</span>.</p>
</div>
```

Modifying the look & feel

- Styles can be applied in a html tag using the attribute `style`
  - ```

      Some text

    ```
- Styles can be defined in the meta-data of the document (directly within `<style></style>` or by linking a style sheet)

index.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Our first page</title>
    <link rel="stylesheet" type="text/css"
      href="style.css">
  </head>
  <body>
  <!-- Page content -->
  </body>
</html>
```
```

Syntax of CSS (Cascading Style Sheets)

style.css

```css
/* General syntax :
 * selector {
 *   property: value;
 * }
 */

p { /*applies to _all_ document paragraphs*/
  color: red;
  text-align: justify;
  background-color: yellow;
}
```

```css
/* Class specification */
.highlight {
  color: red;
  background-color: yellow;
}

/* Id specification */
#main_content {
  background-color: gray;
  font-family: sans-serif;
}
```

Combining HTML & CSS

index.html

```
<div id="main_content">
  <p class="highlight">Some text</p>
  <span class="highlight">Some other text</span>
</div>
```

CSS and pseudo-classes

style.css

```
/* Pseudo-classes */
a:link { /*blue*/
  color: #0000ff;
}

a:hover { /*yellow*/
  color: #ffff00;
}
```

```
a:visited { /*pink*/
  color: #ff00ff;
}

a:active { /* being
clicked - red*/
  color: #ff0000;
}
```

- Specify styles given the state of an element
-
```
selector:pseudoclass {
    property: value;
}
```

Properties to manipulate fonts

- `font-family: Verdana, Times, Arial;`
  - The selected font is the first font found in the user font repository ; otherwise the browser default font is used
- `font-size: 150%;`
  - It's always a good idea to use relative sizes (for everything, not just fonts)
  - Other units can be used (pixel, point, em, …)
- `font-style: italic; /*normal, oblic*/`
- `font-weight: bold; /*normal*/`
- `color: #ffffff;`
  - Color picker: https://www.w3schools.com/colors/colors_picker.asp

Webpage layout

- Arranging a page requires the use of containers
- The containers must be defined in the HTML document
- The layout is defined in the style sheet
- Main elements found in a web page: header, navigation menu, body, footer

Look at https://www.w3schools.com/css/demo_default.htm and see if you can spot those elements

Defining containers (1)

- Use the generic elements (block level) and (inline)
- With these, you can define specific parts of a webpage thanks to class and identifiers

index.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Our first page</title>
  </head>
  <body>
    <div id="nav_menu"></div>
    <div id="main_content">
       <div class="article"></div>
       <div class="article"></div>
    </div>
    <div id="footer"></div>
  </body>
</html>
```

Defining containers (2)

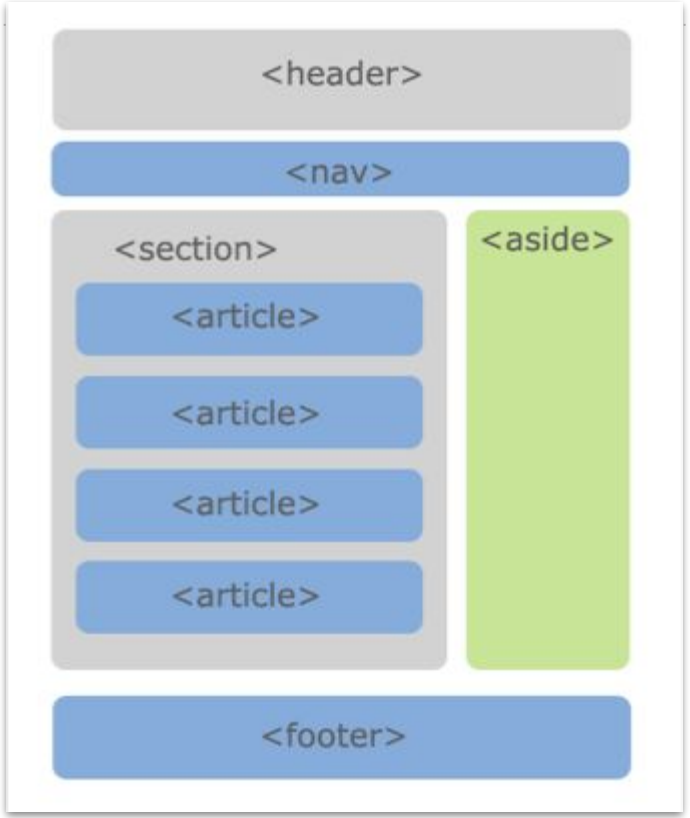
- Alternatively, you can use HTML5 semantic elements specific to page layout
- Header, nav, footer, section, article, aside

index.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Our first page</title>
  </head>
  <body>
    <nav></nav>
    <section>
       <article></article>
       <article></article>
    </section>
    <footer></footer>
  </body>
</html>
```

Layout example

- If the HTML document is properly structured, it's quite easy with CSS do specify the layout of the page



Chapter recap

- Introduction to HTML, a language based on tags
- Basic HTML elements: headings, paragraphs, images, links, lists, tables
- Identifying elements and assigning them classes
- Using a CSS file to define styles
- Using generic elements to structure the page further