

# Web Programming (CSci 130)

Department of Computer Science  
College of Science and Mathematics  
California State University Fresno  
H. Cecotti

# HTML5

---

- Learning outcomes
  - Presentation of HTML5
    - Focus on semantic
  - **Main** functionalities of HTML5

# Introduction (1)

---

- **HTML** to define the content's meaning
  - Semantic (logic)
- **CSS** to define how the content and webpage will look
  - Presentation (layout)
- **HTML5**: to build current and future websites.
  - Not completely different compared to the previous version of HTML
  - Browsers have to be able to read old HTML files as well
  - It is not HTML v5.
- Maintained by the WWW Consortium (W3C)
  - Directed by Tim Berners-Lee (inventor of the Web/HTML)
  - Everything is defined by documents: specifications, set of rules

# Introduction (2)

---

- Another organization
  - The Web Hypertext Application Technology Working Group (WHATWG)
    - HTML specifications
- With an agreed set of rules for Chrome, Firefox, MS Edge...
  - To display and present the webpages \*as expected\*
- New features compared to other versions...
  - Article, section, figure ...
  - Native audio and video playback functions
    - Goodbye the plugins !

# Introduction (3)

---

## ■ Definition of the specifications

➤ The need of a standard → competing standards with 2 ways for specifications (like politics?):

- WHATWG → HTML5

- Focus on Web technologies
  - (Apple, Mozilla, Opera...)

- W3C → pushing XHTML

- But Special constraints of web developers
- 2009: No XHTML2 , we keep HTML5

## ■ What to follow?

➤ WHATWG is more followed

# Introduction (4)

---

- When you create websites...
  - Keep in mind how printed documents are made
    - Be consistent
  - Empathy
    - Think about who is going to read, use what you are making.
    - The content should be accessible to all visitors
      - Old people, people with disabilities....
  - Think about how and where it will be displayed
    - Computer screen/TV/Tablet
      - Size of the screen, distance between the screen and the user

# Semantic in HTML

---

## ■ Why it matters?

### ➤ Improved accessibility and interoperability

- content is available to assistive technologies for visitors with disabilities, and to browsers on desktop, mobile, tablet, ...

### ➤ Improved search engine optimization (SEO)

- search engines put an emphasis on the parts of your content that are marked up in a particular way:
  - headings tell the search engine spider the primary topics of your page, helping the search engine determine how to index your page's content
- **Spider**: program that visits Web sites and reads their pages and other information for creating entries for a **search engine** index

### ➤ Typically lighter code and faster pages

### ➤ Easier code **maintenance** and **styling**

# What you need

---

## ■ Software requirement

- Just a text editor (like notepad)
- Better to use notepad++
  - Free download
  - To better see the syntax and the different tags
    - Easier to find errors
- To see the results
  - Any web browser (Chrome, Edge, Firefox,...)

## ■ Hardware requirement

- Any laptop
- You can use a text editor on a tablet/phone



# Before starting

## ■ Planning your site

➤ Like the Lab of the first week

➤ Graph of the web pages

- Try to separate what is data related, what can be obtained from a database to fill the content of the webpage

➤ Think “meta”

- Classes describing classes of objects
- Documents describing how documents should be presented



# Rich Internet Application (RIA)

---

- RIA = desktop application through the browser
  - More than just a document
- Before HTML5
  - Adobe Flash
    - Lots of students/people trained ONLY on Adobe Flash ☹
      - Job for 5 years then unemployed...
  - Microsoft Silverlight [2007-2017]
    - Application framework for writing and running rich Internet applications
  - Java applet
    - Small application written in the Java programming language
- Now HTML5
  - 😊

# HTML Syntax

## ■ Main structures

➤ Tag alone (go to next line) `<br>` , `<hr>`

- `<tag>`

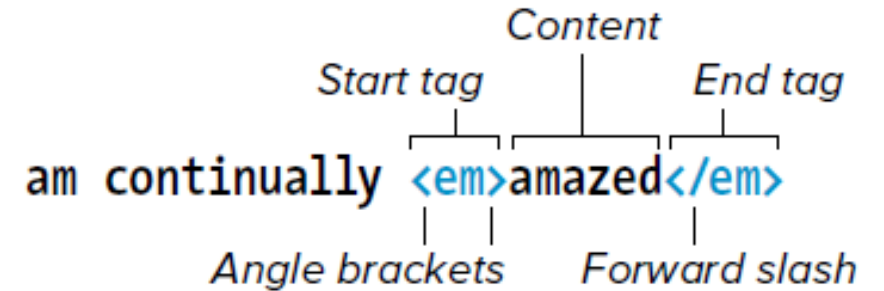
➤ Tag with content (that can contain tags) `<p>it is a paragraph</p>`

- `<tag>` content `</tag>`

➤ Attributes: type, id, name, value

- Examples

- `<input type="radio" id="gender_male" name="gender" value="male" />`
- `<h2 class="account">account</h2>`
- `<form method="post" action="page01.html">`
- ``



# HTML Syntax

---

- Attributes: value and name
- Example
  - Textbox will have a default **value**
  - If this form is submitted
    - The browser collects all the data of this form
    - The browser sends it to the server
    - PHP on the server side → access this data by referencing the key given in the **name**
- Client (HTML)  
`<input type="text" name="name" value="Jim">`
- Server (PHP)  

```
<?php
$name = $_POST['name'];
// $name is set to "Jim"
?>
```

# HTML Syntax



*for is an attribute of label*  
`<label for="email">Email Address</label>`  
*The value of the for attribute*

*href is an attribute of a*  
*Value for href*  
*rel is also an attribute of a*  
*Value for rel*  
`<a href="http://en.wikipedia.org/wiki/Linum_lewisii" rel="external"`  
*title is an attribute of a*  
*Value for title*  
`→ title="Learn more about Blue Flax">Blue Flax</a>`

`<link rel="stylesheet" media="screen" href="blueflax.css" />`  
*Predefined value*

# The main elements

---

- List of components in the HTML file
  - The DOCTYPE (remember the DTD in XML)
  - **Html** element (with the **lang** attribute)
  - **Head** element
  - The character encoding in a meta element
  - The **title** element
  - The **body** element
- Two main parts in the HTML document
  - Head
  - Body

# Start of HTML5 page

---

- First, to tell it is HTML5

- `<!DOCTYPE html>` (simple and short)

- XHTML:

- `<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">`

- To declare the page as an HTML document

- To tell the language: `<html lang="fr">` for French

- In the head element

- `<meta charset="UTF-8"/>` : character encoding

- The title (of the page)

# The main document (1)

---

- Example:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<meta charset="utf-8" />
```

```
<title>title here</title>
```

```
</head>
```

```
<body> content here </body>
```

```
</html>
```



# The main document (2)

---

- The previous example

- The ideal structure

- If the program parsing the file knows it is HTML5 then several assumptions can be done

- Many elements can be removed because

- Expectation about the file (structure)
    - To generate automatically the document
    - Example:
      - <https://code.google.com/archive/p/htmlcompressor/>
    - Brotli
      - HTML compression
      - <https://opensource.googleblog.com/2015/09/introducing-brotli-new-compression.html>

# The main document (3)

- Example
  - Red: HTML tags
  - Black: content

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8" />
  <title>Blue Flax (Linum lewisii)</title>
</head>
<body>
  <article>
    <h1>The Ephemeral Blue Flax</h1>

    <p>I am continually <em>amazed</em> at the beautiful, delicate <a href="http://
    → en.wikipedia.org/wiki/Linum_lewisii" rel="external" title="Learn more about Blue
    → Flax">Blue Flax</a> that somehow took hold in my garden. They are awash in color every
    → morning, yet not a single flower remains by the afternoon. They are the very definition
    → of ephemeral.</p>
  </article>
</body>
</html>
```

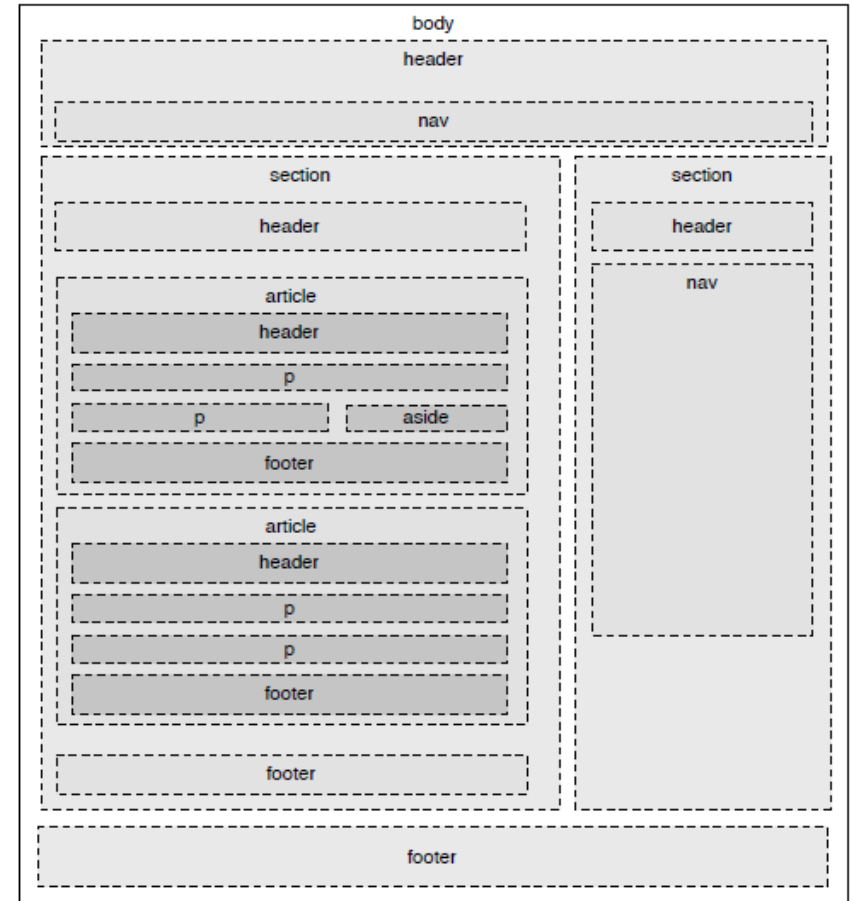
# The body

---

- Decomposition of the page into blocks
  - With a semantic
    - Goal of HTML: applying **meaning** to content
- Blocks with no semantic – for the presentation
  - Special case: div and span
    - **span** element (in-line)
      - Used for a small chunk of **HTML** inside a line
    - **div** (division) element (block-line)
      - Used to group larger chunks of code

# The body

- Typical HTML5 structure of the body
  - basic tags to know for midterms



# The body

---

- Bad practice: too many useless div
  - If you use an HTML editor (wysiwyg)
    - Be careful: the editor is not smart to generate the base page, you have to check what has been generated

```
<div id="page">
  <div id="navbar_wrapper">
    <div id="navbar">
      <ul>
        <li><a href="/">Home</a></li>
        <li><a href="/products">Products</a></li>
        ...
      </ul>
    </div>
  </div>
</div>
```

# The body

---

- `<header>`
  - Defines a header region of a page or section
- `<footer>`
  - Defines a footer region of a page or section
- `<nav>`
  - Defines a navigation region of a page or section
- `<section>`
  - Defines a logical region of a page or a grouping of content
- `<article>`
  - Defines an article or complete piece of content
- `<aside>`
  - Defines secondary or related content.

# The body

---

- Headers `<header>`
  - `/!\` not to be confused with headings: `<h1>`, `<h2>`, and `<h3>`
  - It may contain all sorts of content
    - company logo, the search box,...
- Footers `<footer>`
  - Defines footer information for a document or an adjacent section
  - It may contain information: copyright date, who owns the site, complex navigation structures...
  - You can have multiple footers in a document
    - → footers within blog articles
- Navigation `<nav>`
  - Navigation: critical UX feature
    - Must be easy for people to find what they need → navigation to get its own HTML tag
- Sections and Articles
  - **logical** regions of a page: `<section>` tag to replace the abused `<div>`
  - `<section>`: logical part of a document with related content
  - `<article>`: actual content
    - magazine article, blog post, news item,...

# The body

## ■ Example

```
<article class="post">
  <header>
    <h2>How Many Peanuts Should We Eat Daily?</h2>
    <p>Posted by Jacob on
    <time datetime="2015-09-01T14:25">September 1st, 2015 at 2:39PM</time>
    </p>
  </header>
  <p>
    Paid was hill sir high. For him precaution any advantages dissimilar comparison few terminated projecting.
    Repair summer one winter living feebly pretty his.
    In so sense am known these since. Shortly respect ask cousins brought add tedious nay.
  </p>
  <p>
    Greatest properly off ham exercise all. Unsatiabable invitation its possession nor off.
    All difficulty estimating unreserved increasing the solicitude.
    Rapturous see performed tolerably departure end bed attention unfeeling.
    On unpleasing principles alteration of. Be at performed preferred determine collected.
  </p>
  <footer>
    <p><a href="comments"><i>4 comments</i></a> ...</p>
  </footer>
</article>
```



# Headers

## ■ Example

```
...
<body>
<header role="banner">
  ... [site logo, navigation, etc.] ...
</header>

<article>
  <header>
    <h1>Frequently Asked Questions</h1>
    <nav>
      <ul>
        <li><a href="#answer1">What is your return policy?</a></li>
        <li><a href="#answer2">How do I find a location?</a></li>
        ...
      </ul>
    </nav>
  </header>

  <!-- the header links point to these -->
  <article id="answer1">
    <h2>What is your return policy?</h2>
    <p> ... [answer] ... </p>
  </article>

  <article id="answer2">
    <h2>How do I find a location?</h2>
    <p> ... [answer] ... </p>
  </article>
  ...
</article> <!-- end parent article -->

</body>
</html>
```

# Nav

---

## ■ Example

➤ Role → to improve accessibility

```
...  
<body>  
<header>  
  <nav role="navigation">  
    <ul>  
      <li><a href="#gaudi">Barcelona's  
        → Architect</a></li>  
      <li lang="es"><a href="#sagrada-  
        → familia">La Sagrada Família</a>  
        → </li>  
      <li><a href="#park-guell">Park  
        → Guell</a></li>  
    </ul>  
  </nav>  
</header>  
</body>  
</html>
```

# Aside

## ■ Example

```
...
<body>

<header>
  <nav role="navigation">
    ... [ul with links] ...
  </nav>
</header>

<article>
  <h1 id="gaudi">Barcelona's Architect</h1>
  ... [introductory paragraphs] ...

  <section>
    <h2 id="sagrada-familia" lang="es">La Sagrada Família</h2>
    ... [image and paragraph] ...
  </section>

  <section>
    <h2 id="park-guell">Park Guell</h2>
    ... [another image and paragraphs] ...
  </section>
</article>

<aside role="complementary">
  <h1>Architectural Wonders of Barcelona</h1>

  <p>Barcelona is home to many architectural wonders in addition to Gaudí's work. Some of them
  → include:</p>
  <ul>
    <li lang="es">Arc de Triomf</li>
    <li>The cathedral <span lang="es">(La Seu)</span></li>
    <li lang="es">Gran Teatre del Liceu</li>
    <li lang="es">Pavillion Mies van der Rohe</li>
    <li lang="es">Santa Maria del Mar</li>
  </ul>

  <p>Credit: <a href="http://www.barcelona.de/en/barcelona-architecture-buildings.html"
  → rel="external"><cite>Barcelona.de</cite></a></p>
</aside>

</body>
</html>
```

# Footer

## ■ Example

```
...
<body>
<header>
  <nav role="navigation">
    ... [ul with links] ...
  </nav>
</header>

<article>
  <h1 id="gaudi">Barcelona's Architect</h1>
  ... [introductory paragraphs] ...

  <section>
    <h2 id="sagrada-familia" lang="es">La
    → Sagrada Família</h2>
    ... [image and paragraph] ...
  </section>

  <section>
    <h2 id="park-guell">Park Guell</h2>
    ... [another image and paragraphs] ...
  </section>
</article>

<aside role="complementary">
  <h1>Architectural Wonders of Barcelona
  → </h1>
  ... [rest of aside] ...
</aside>

<footer>
  <p><small>&copy; Copyright 2011</small>
  → </p>
</footer>

</body>
</html>
```

# Creation of content (1)

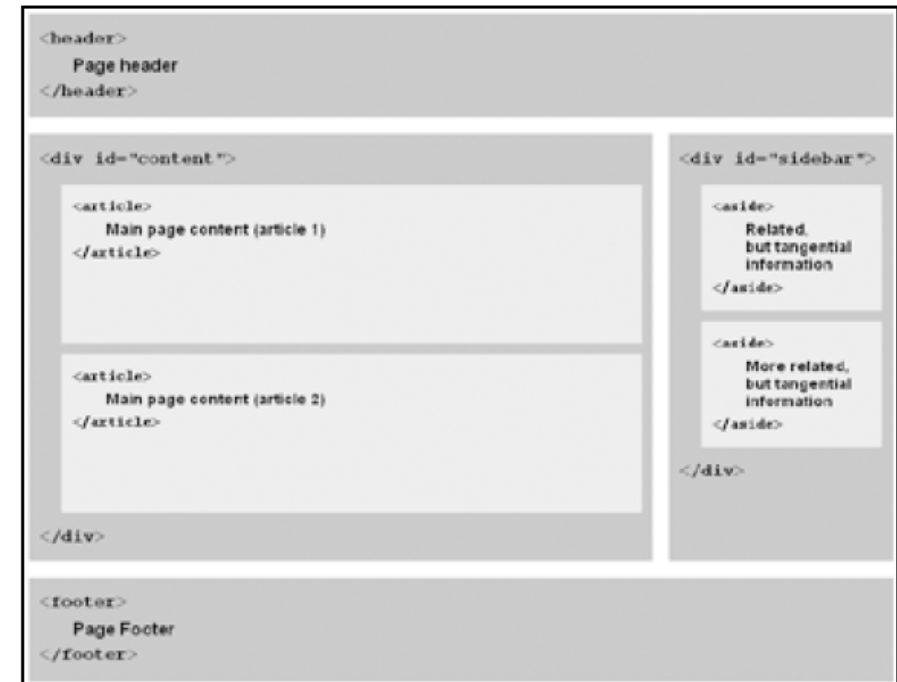
---

- Even without CSS you will have some effects for the presentation, it won't be just plain text BUT you have to keep in mind that all the presentation should be done with CSS.
- See HTML examples
  - Files on Canvas

# Creation of content

- Header
- Div = Generic container
  - Articles/Sections/Aside
- Footer

```
...  
<body>  
  
  <!-- Start page container -->  
  <div id="container">  
    <header>  
      ...  
    </header>  
  
    <!-- Column One when CSS applied -->  
    <div id="content">  
      <article>  
        ...  
      </article>  
  
      <article>  
        ...  
      </article>  
  
      ... [more sections as desired] ...  
    </div>  
    <!-- end column one -->  
  
    <!-- Column Two when CSS applied -->  
    <div id="sidebar">  
      <aside>  
        ...  
      </aside>  
  
      <aside>  
        ...  
      </aside>  
  
      ... [more sections as desired] ...  
    </div>  
    <!-- end column two -->  
  
    <footer>  
      ...  
    </footer>  
  </div>  
  <!-- end page container -->  
  
</body>  
</html>
```



# Links

## ■ Hyperlink

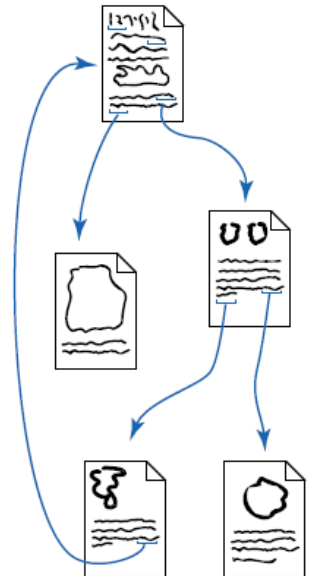
### ➤ <a> tag + href attribute

- <a href="page.html">
- <a href="http://www.site.com/directory/page.html" rel="external">Label text</a>

### ➤ rel="external"

- to indicate that a link points to a different site

```
<p>Generally considered the sweetest  
→ and yet most independent cats in the  
→ <a href="pioneer-valley.html">  
→ Pioneer Valley</a>, Cookie and Woody  
→ are consistently underestimated by  
→ their humble humans.</p>
```



# Anchors

- Add ID to your elements

- `<a href="#anchor-name">`

```
<header>
  <h2>Table of Contents</h2>
  <nav>
    <ul>
      <li><a href="#intro">Introduction</a></li>
      <li><a href="#main-characters">Description of the Main Characters</a></li>
      <li><a href="#rising-action">Rising Action</a></li>
    </ul>
  </nav>
</header>

<article>
  <h2 id="intro">Introduction</h2>
  <p>This is the intro. If I could think of enough things to write about, it could span a few
  → pages, giving all the introductory information that an introduction should introduce.</p>

  <h2 id="main-characters">Description of the Main Characters</h2>
  <p>Frankie and Johnny are the main characters. She's jealous, and seems to have a reason to be.
  → He's a sleaze, and will pay the price.</p>

  <h2 id="rising-action">Rising Action</h2>
  <p>This is where everything starts happening. Johnny goes out, without Frankie, without even
  → tellin' her where he's going. She's not crazy about it, but she lets him go. A while later,
  → she gets thirsty and decides to go down to the corner bar for some beer. Chatting with the
  → bartender, she learns that Johnny has been there with no other than Nellie Bly. Furious, she
  → catches the crosstown bus to find him.</p>
</article>
```



# Some words about images

---

- Quick edition of images
  - Think about the size (what is going to be represented on the screen)
- Properties
  - Format
  - Color
  - Size/resolution
  - Speed
  - Transparency
  - Animation
- See examples...
  - gimp
  - xnview

# Conclusion

---

- HTML5
  - It has a large number of tags and ways to set a semantic to elements in your document
  - For some element, there is no strict rule about what to use
    - Implied rules by developers
  - The best way to learn is to practice and to test the commands
- Reading
  - See links on Canvas
- Acknowledgment
  - Book: HTML5 and CSS3, 7<sup>th</sup> edition, Castro & Hyslop
- Questions ?

