

# CMPT 165

## Advanced XHTML + CSS – Part 1

June 3<sup>rd</sup>, 2015

# Today's Agenda

- CSS reviewed
- Unit 4:
  - id vs. class
  - Classifying elements: blocks vs. inline
  - More markup tags
    - <div>
    - <span>

# “Style rules” vs. “Attribute specification”

## Style declaration in CSS:

```
p {  
  text-align: center;  
  font-size: 1.5em;  
}
```

**declaration**

**rule**

## Attribute specification in markup:

<p style="text-align:center; font-size:1.5em">...

<table summary="Table of measured pH.">...

# CSS Level 1

Style support for:

- **Color** of element text, backgrounds, etc.
- **Alignment** of elements (text, images, etc.)
- **Text** formatting: e.g. spacing of words, letters, lines
- **Font** properties: typeface and emphasis
- **Boxing**: margin, border, padding, and positioning
- **Unique identification**: id
- **Generic classification**: explained later

# Margin, padding, border (review)

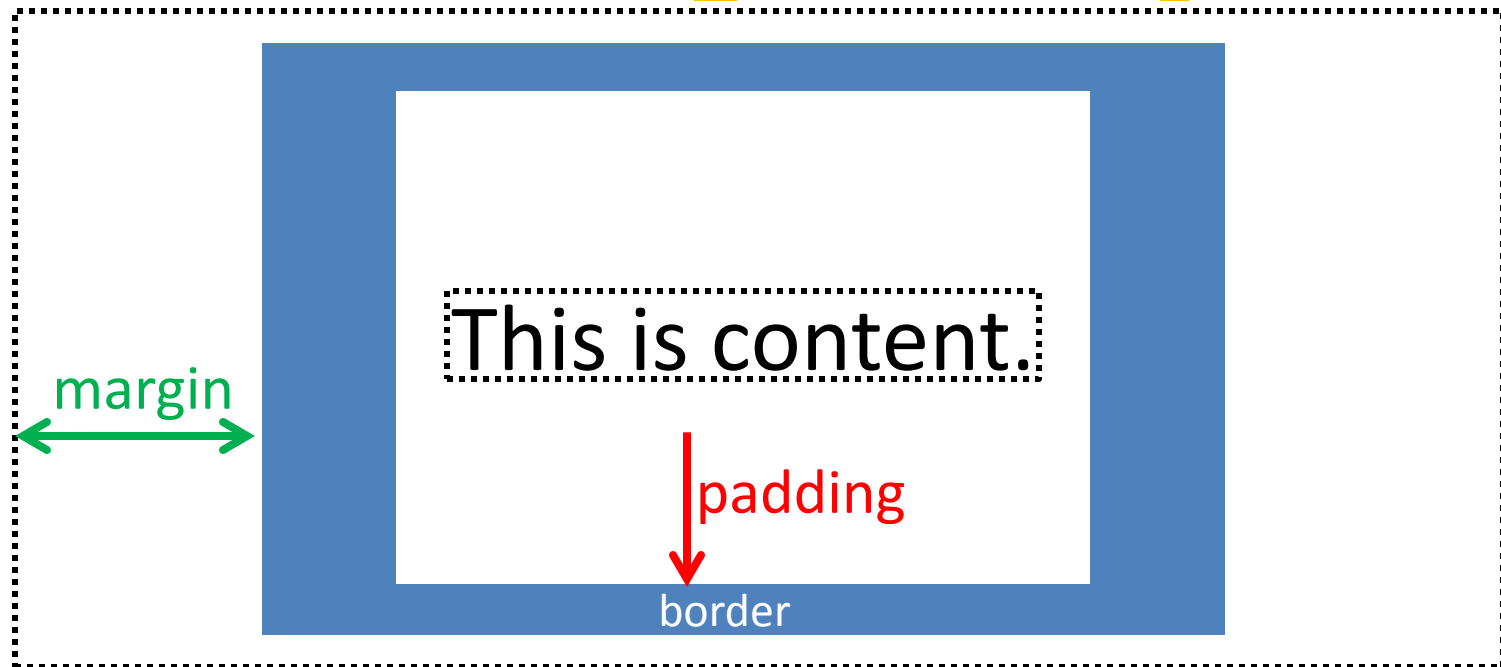
**Content** - Content of the box, where text and images appear

**Margin** - Clears an area outside the border; transparent

**Border** - A border that goes around the padding and content

**Padding** - Clears an inner area around the content; transparent

“m” before “p” → outer margin before “inner padding”



# CSS Level 1

Style support for:

- **Color** of element text, backgrounds, etc.
- **Alignment** of elements (text, images, etc.)
- **Text** formatting: e.g. spacing of words, letters, lines
- **Font** properties: typeface and emphasis
- **Boxing**: margin, border, padding, and positioning
- **Unique identification of elements**: id
- **Generic classification of elements**: explained later

# id attribute

```
<p> Q1: Gfe dsf sdf? </p>
```

```
<p> A1: dsf jjf jsf.</p>
```

```
<p> Q2: Fsjf jsadf? </p>
```

```
<p> A2: Sbjffa al eef. </p>
```

```
<p> Q3: Msjf jsadf? </p>
```

```
<p> A3: Pbjff. La al. </p>
```

```
p {  
  font-family: sans-serif  
}
```

Browser:

Q1: Gfe dsf sdf?

A1: dsf jjf jsf.

Q2: Fsjf jsadf?

A2: Sbjffa al eef.

Q3: Msjf jsadf?

A3: Pbjff. La al.

# id attribute

```
<p id="q1"> Q1: Gfe dsf sdf? </p>
```

```
<p> A1: dsf jjf jsf.</p>
```

```
<p> Q2: Fsjf jsadf? </p>
```

```
<p> A2: Sbjffa al eef. </p>
```

```
<p> Q3: Msjf jsadf? </p>
```

```
<p> A3: Pbjff. La al. </p>
```

```
p {  
  font-family: sans-serif  
}
```

Browser:

Q1: Gfe dsf sdf?

A1: dsf jjf jsf.

Q2: Fsjf jsadf?

A2: Sbjffa al eef.

Q3: Msjf jsadf?

A3: Pbjff. La al.



# id attribute

```
<p id="q1"> Q1: Gfe dsf sdf? </p>
```

```
<p> A1: dsf jjf jsf.</p>
```

```
<p> Q2: Fsjf jsadf? </p>
```

```
<p> A2: Sbjffa al eef. </p>
```

```
<p> Q3: Msjf jsadf? </p>
```

```
<p> A3: Pbjff. La al. </p>
```

```
p {  
  font-family: sans-serif  
}  
#q1{ font-style: bold; }
```

Browser:

**Q1: Gfe dsf sdf?**

A1: dsf jjf jsf.

Q2: Fsjf jsadf?

A2: Sbjffa al eef.

Q3: Msjf jsadf?

A3: Pbjff. La al.

# id attribute

```
<p id="q1"> Q1: Gfe dsf sdf? </p>  
<p> A1: dsf jjf jsf.</p>
```

```
<p id="q2"> Q2: Fsjf jsadf? </p>  
<p> A2: Sbjffa al eef. </p>
```

```
<p id="q3"> Q3: Msjf jsadf? </p>  
<p> A3: Pbjff. La al. </p>
```

```
p {  
  font-family: sans-serif  
}  
  
#q1{ font-style: bold; }  
#q2{ font-style: bold; }  
#q3{ font-style: bold; }  
#q4{ font-style: bold; }  
#q5{ font-style: bold; }
```

Browser:

**Q1: Gfe dsf sdf?**

A1: dsf jjf jsf.

**Q2: Fsjf jsadf?**

A2: Sbjffa al eef.

**Q3: Msjf jsadf?**

A3: Pbjff. La al.

If you have 5  
questions?

An alternative:  
group into single  
declaration

# id attribute

```
<p id="q1"> Q1: Gfe dsf sdf? </p>  
<p> A1: dsf jjf jsf.</p>
```

```
<p id="q2"> Q2: Fsjf jsadf? </p>  
<p> A2: Sbjffa al eef. </p>
```

```
<p id="q3"> Q3: Msjf jsadf? </p>  
<p> A3: Pbjff. La al. </p>
```

```
p {  
  font-family: sans-serif  
}  
#q1, #q2, #q3, #q4, #q5 {  
  font-style: bold;  
}
```

Browser:

**Q1: Gfe dsf sdf?**

A1: dsf jjf jsf.

**Q2: Fsjf jsadf?**

A2: Sbjffa al eef.

**Q3: Msjf jsadf?**

A3: Pbjff. La al.

If you have 5  
questions?

An alternative:  
group into single  
declaration

# id attribute

```
<p id="q1"> Q1: Gfe dsf sdf? </p>
<p> A1: dsf jjf jsf.</p>
```

```
<p id="q2"> Q2: Fsjf jsadf? </p>
<p> A2: Sbjffa al eef. </p>
```

```
<p id="q3"> Q3: Msif jsadf? </p>
<p> A3:
```

```
p {
  font-family: sans-serif
}
#q1, #q2, #q3, #q4, #q5 {
  font-style: bold;
}
```

What if you have  
1000 Q's?

Browse

**Q1:** G

A1: dsf jjf jst.

**Q2:** Fsjf jsadf?

A2: Sbjffa al eef.

**Q3:** Msif jsadf?

A3: Pbfff. La al.

# Solution: class

```
<p class="q"> Q1: Gfe dsf sdf? </p>  
<p> A1: dsf jjf jsf.</p>
```

```
<p class="q"> Q2: Fsjf jsadf? </p>  
<p> A2: Sbjffa al eef. </p>
```

```
<p class="q"> Q3: Msjf jsadf? </p>  
<p> A3: Pbjff. La al. </p>
```

```
p {  
  font-family: sans-serif  
}  
  
.q {  
  font-style: bold;  
}
```

Browser:

**Q1: Gfe dsf sdf?**

A1: dsf jjf jsf.

**Q2: Fsjf jsadf?**

A2: Sbjffa al eef.

**Q3: Msjf jsadf?**

A3: Pbjff. La al.

# id, class attributes

## id attribute

Instantiate:

```
<p id="name_of_id">H</p>
```

Reference:

```
#name_of_id
```

Anchoring in html:

```
<a href="#name_of_id">top</a>
```

Styling in CSS:

```
#name_of_id {  
    color: red;  
}
```

## class attribute

Instantiate:

```
<h1 class="class_name">H</h1>
```

Reference:

```
.class_name
```

Styling in CSS:

```
.class_name {  
    color: red;  
}
```

# id vs. class?

“Class vs. id” explained:

- There are many students in a class
- There is a unique ID for every student

Usage:

- When some visual properties need to be reapplied, use `class`
- If an element has some unique property, use `id`

# Side note

Use underscore in place of white spaces

e.g.

“my\_first\_exercise.html”

e.g.

“my first exercise.html”

URL:

my%20first%20assignment.html

e.g.

my-first-exercise.html

... but not as easy to read(?)



# Today's Agenda

- CSS reviewed
- Unit 4:
  - id vs. class
  - Classifying elements: blocks vs. inline
  - More markup tags
    - <div>
    - <span>

# Key elements learned so far

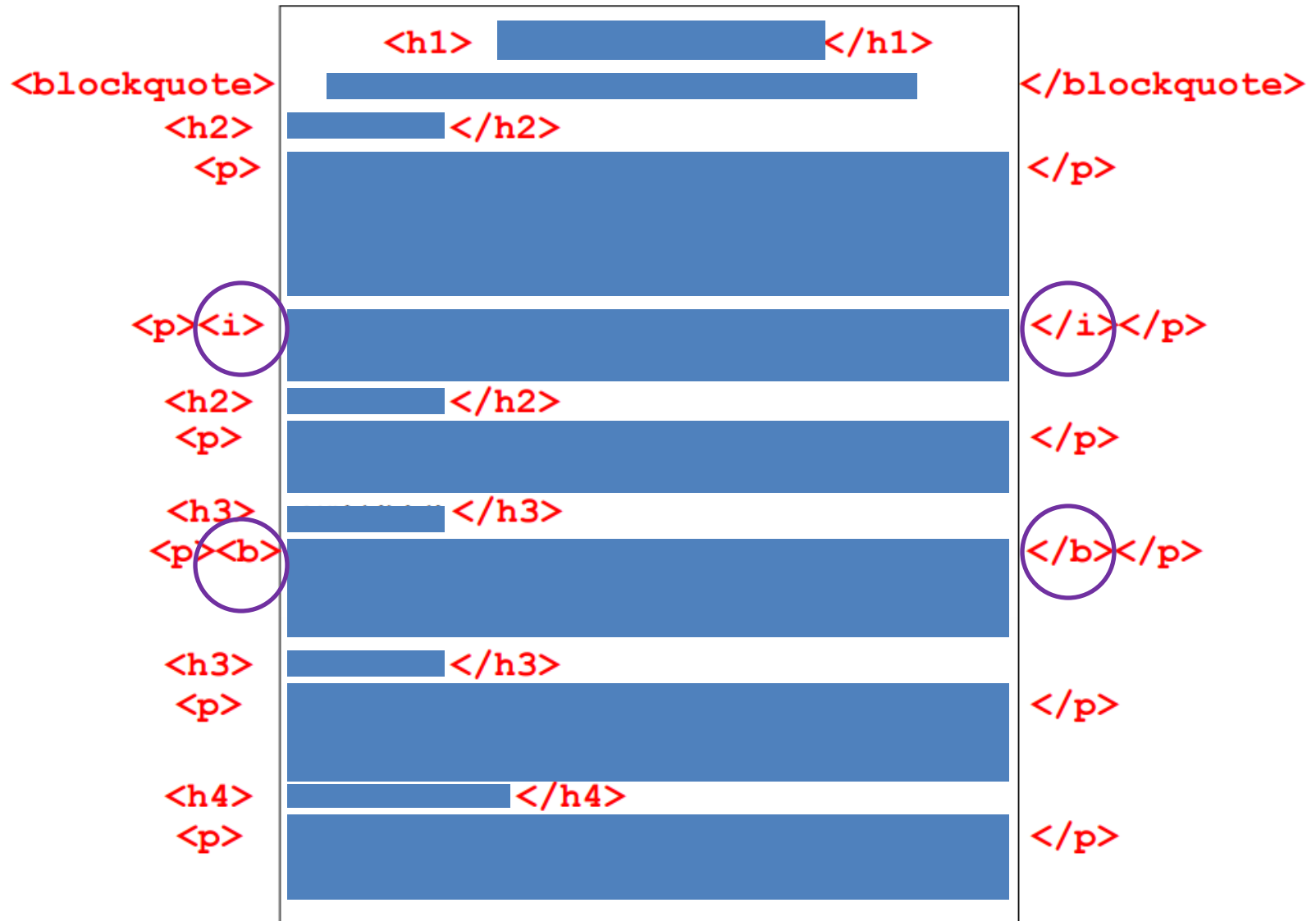
## Block

<p>  
<h1>  
<hr>  
<blockquote>  
<ol>  
<ul>  
<li>  
<table>

## Inline

<b>  
<i>  
<em>  
<strong>  
<cite>  
<tt>  
<sub>  
<sup>

# Block vs. inline elements



# Block vs. Inline elements

## Block

Nature: structural

Guidelines:

- Can contain inline elements

`<h1>I <em>can</em>!</h1>`

- Can be inside another block element
- Not inside an *inline* element!

Behaviour (without using style rules):

- Expands across the page
- Drop below other elements

## Inline

Nature: used along with text

Guidelines:

- Usually can contain another inline element

`<q>I <em>can</em>!</q>`

- Usually must be inside block elements!  
i.e. Cannot be directly inside `<body>`

Behaviour (without using style rules):

- Flows along with text

Expands across the page  
Next element is forced to next line



# Elements

## Block

<p>  
<h1>  
<hr>  
<blockquote>  
<ol>  
<ul>  
<li>  
<table>

<a>?

## Inline

<b>  
<i>  
<em>  
<strong>  
<cite>  
<tt>  
<sub>  
<sup>

# Exceptions

- Anchor tag
  - An inline element (flows along with text)
  - **But:**
    - *Can* be placed directly in the body!
    - `<a>` *cannot* contain another `<a>`
- Paragraph, header tags
  - Cannot be placed in within each other; wrong:  
~~`<p><p>Abcdefg</p></p>`~~
  - Its content must be an inline element; wrong:  
~~`<h1><p>Abcdefg</p></h1>`~~
- `<ol>` `<ul>` must contain other elements `<li>`

# Key elements learned so far...

## Block

<p>  
<h1>  
<hr>  
<blockquote>  
<ol>  
<ul>  
<li>  
<table>

## Inline

<b>  
<i>  
<em>  
<strong>  
<cite>  
<img>  
<a>  
<tt>  
<sub>  
<sup>  
<abbr>

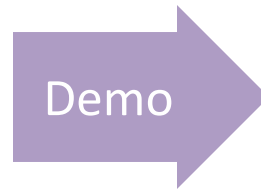


# <abbr> tag

## Inline elements:

```
<abbr title="HyperText Markup Language">HTML</abbr>
```

- Provides a form of interaction
- Nice way of separating high and low level information
- Omitting the title attribute will not give error!
  - But you should not omit it!



# Elements

## Block

<p>  
<h1>  
<hr>  
<blockquote>  
<ol>  
<ul>  
<li>  
<table>  
<div>

## Inline

<b>  
<i>  
<em>  
<strong>  
<img>  
<cite>  
<a>  
<tt>  
<abbr>  
<sub>  
<sup>  
<span>

# Generic Tags

`<div>`     division

- Generic block element
- Used to divide a webpage into related parts

`<span>`     span

- Generic inline element
- Mostly used to style a specific parts of content

# <span> illustrated

Lorem ipsum dolor sit amet, cum ea periculis complectitur, ex quo option alienum. *Ex tale temporibus mei, graeco fuisset omittam an vel, sed ex brute decore.* Ea putant fuisset patrioque eum, atqui integre vivendum cum ex, cu sea sadipscing neglegentur. Mea ne wisi commodo, his eu. No decore voluptua nam.



```
<p>Lorem ipsum dolor sit amet, cum ea periculis  
complectitur, ex quo option alienum. <span  
class="cool_quote"> Ex tale temporibus mei, graeco  
fuisset omittam an vel, sed ex brute decore. </span>  
Ea putant fuisset patrioque eum, atqui integre  
vivendum cum ex, cu sea sadipscing neglegentur. Mea  
ne wisi commodo, his eu. No decore voluptua nam.</p>
```

# <span> illustrated

Lorem ipsum dolor sit amet, cum ea periculis complectitur, ex quo option alienum. *Ex tale temporibus mei, graeco fuisset omittam an vel, sed ex brute decore.* Ea putant fuisset patrioque eum, atqui integre vivendum cum ex, cu sea sadipscing neglegentur. Mea ne wisi commodo, his eu. No decore voluptua nam.



```
<p>Lorem ipsum dolor sit amet, cum ea periculis  
complectitur, ex quo option alienum. <span  
class="cool_quote"> Ex tale temporibus mei, graeco  
fuisset omittam an vel, sed ex brute decore.  
</span> Ea putant fuisset patrioque eum, atqui  
integre vivendum cum ex, cu sea sadipscing  
neglegentur. Mea ne wisi commodo, his eu. No decore  
voluptua nam.</p>
```

```
.cool_quote{  
    font-family: serif;  
    color: #33b;  
    font-weight: bold;  
    font-style: italic;  
}
```

# Generic Tags

`<div>`     division

- Generic block element
- Used to divide a webpage into related parts

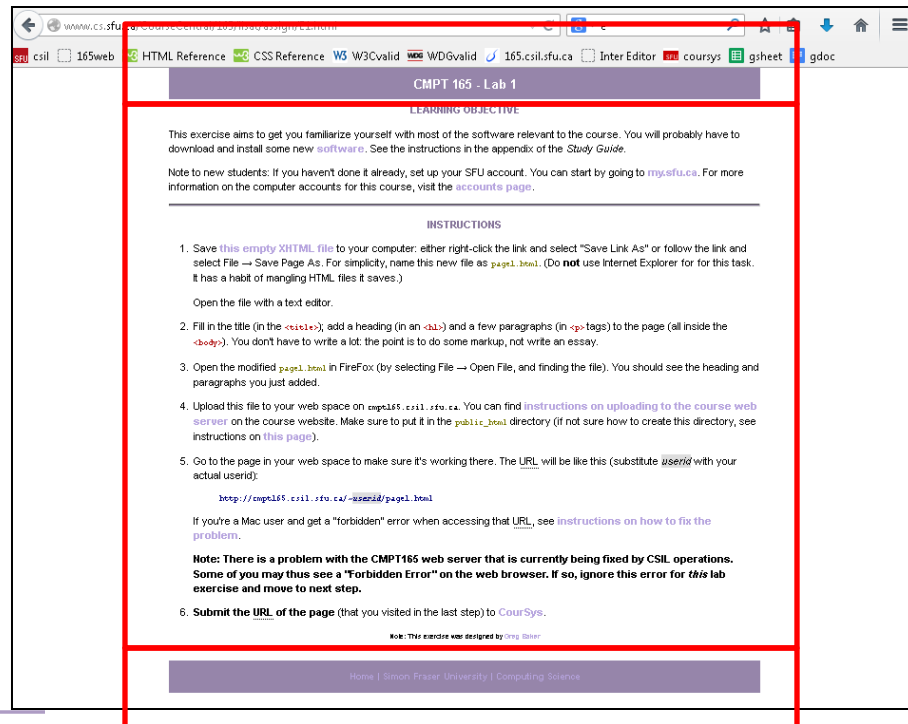
`<span>`     span

- Generic inline element
- Mostly used to style a specific parts of content

# <div> uses

(Very powerful tool, but don't abuse usage; complicated markup → difficult to maintain)

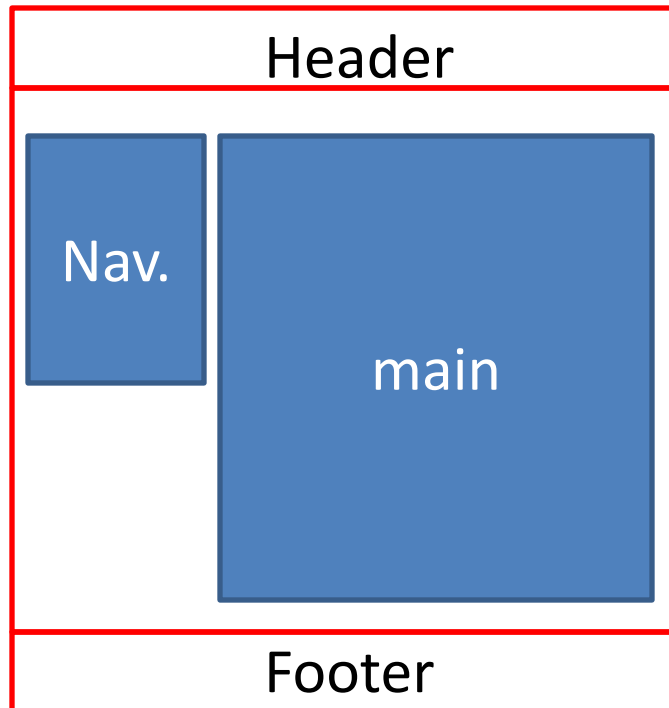
- Separate a page into *logical* sections
- Typical use:



# <div> uses

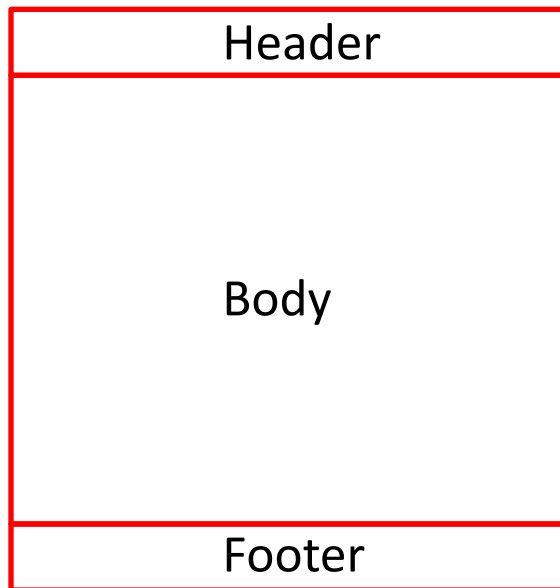
(Very powerful tool, but don't abuse usage; complicated markup → difficult to maintain)

- Separate a page into *logical* sections
- Typical use:





# <div> illustrated

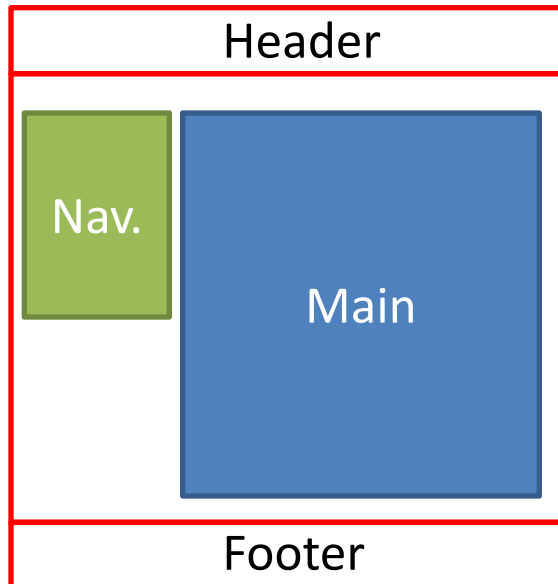


```
<div id="header">
    <h1> My Heading </h1>
</div>

<div id="body">
    <p>This is body paragraph1.</p>
    <p>This is body paragraph2.</p>
    <p>This is body paragraph3.</p>
    <p>This is body paragraph4.</p>
    <p>This is body paragraph5.</p>
</div>

<div id="footer">
    <p>Copyright goes here.</p>
</div>
```

# <div> illustrated



```
<div id="header"> <!-- divider 1 -->
  <h1> My Heading </h1>
</div>
<div id="body"> <!-- divider 2 -->
  <div id="nav"> <!-- subdivider#1 -->
    <ul>
      <li><a href="index.html">Home</a></li>
      <li><a href="me.html">About me</a></li>
      <li><a href="notes.html">My course notes</a></li>
      <li><a href="contact.html">Contact</a></li>
    </ul>
  </div>
  <div id="main_body"> <!-- subdivider#2 -->
    <p>Lorem ipsum dolor sit amet, cum ea periculis
      complectitur, ex quo option alienum. Ex tale
      temporibus mei, graeco fuisset omittam an vel, sed ex
      brute decore. Ea putant fuisset patrioque eum, atqui
      integre vivendum cum ex. Ea putant fuissetr. Mea ne
      wisi commodo, his eu. No decore voluptua nam.</p>

    <p>Bom ipsucomplectitur, ex quo option alienum. <i
      style="font-color;blue;"> Ex tale temporibus mei,
      graeco fuisset omittam an vel, sed ex brute decore.
      </i> Ea putant fuisset patr his eu. No decore
      voluptua nam.</p>
  </div>
</div>
<div id="footer"> <!-- divider 3 -->
  <p>Copyright goes here.</p>
</div>
```

# <div> illustrated

```

/* general styling */
div {
    padding: .01%;
    background-color: #afa;
    border: #5b5 solid .2em;
    margin: .01%;
}

#body {
    height: 50%;
}

/* specific items */

#header, #footer{
    text-align: center;
    clear: both;
}

#nav {
    width: 15%;
    float: left;
    min-height: 20em;
}

#main_body {
    width: 82%;
    float: left;
    min-height: 20em;
}

```

```

<div id="header"> <!-- divider 1 -->
    <h1> My Heading </h1>
</div>
<div id="body"> <!-- divider 2 -->
    <div id="nav"> <!-- subdivider#1 -->
        <ul>
            <li><a href="index.html">Home</a></li>
            <li><a href="me.html">About me</a></li>
            <li><a href="notes.html">My course notes</a></li>
            <li><a href="contact.html">Contact</a></li>
        </ul>
    </div>
    <div id="main_body"> <!-- subdivider#2 -->
        <p>Lorem ipsum dolor sit amet, cum ea periculis
        complectitur, ex quo option alienum. Ex tale
        temporibus mei, graeco fuisset omittam an vel, sed ex
        brute decore. Ea putant fuisset patrioque eum, atqui
        integre vivendum cum ex. Ea putant fuissetr. Mea ne
        wisi commodo, his eu. No decore voluptua nam.</p>

        <p>Bom ipsucomplectitur, ex quo option alienum. <i
        style="font-color;blue;"> Ex tale temporibus mei,
        graeco fuisset omittam an vel, sed ex brute decore.
        </i> Ea putant fuisset patr his eu. No decore
        voluptua nam.</p>
    </div>
</div>
<div id="footer"> <!-- divider 3 -->
    <p>Copyright goes here.</p>
</div>

```

# Important elements covered so far...

## Block

<p>  
<h1>  
<hr>  
<blockquote>  
<ol>  
<ul>  
<li>  
<table>  
<div>

## Inline

<em>  
<strong>  
<cite>  
<sub>  
<sup>  
<img>  
<a>  
<abbr>  
<span>