# Layout with CSS

Pat Morin

COMP 2405

Carleton
UNIVERSITY
Canada's Capital University

# *Outline*

- Inline versus block elements
- Blocks
  - properties
  - the CSS box model
- Example layouts using blocks
- Changing the `display` property
  - Reformatting lists

Carleton UNIVERSITY

Canada's Capital University

# *Inline versus Block Elements*

- Some HTML tags define *block* elements
  - Normally rendered with a space before and a space after
  - They take up the entire width of the browser
  - Examples: `P`, `H1`, `,H2`, `UL`, `PRE`, `DIV`
- Other HTML tags define *inline* elements
  - The are rendered inline (to the right of the preceding element)
  - They only take up as much width as necessary
  - Examples: `A`, `EM`, `CODE`, `SPAN`
- Block elements can contain inline elements but not the other way around (try this with a validator)

# *Laying Out Blocks*

- Block elements can be used for layout (positioning) on pages
- We do this by specifying their positions, width, margins, borders, …

Canada's Capital University

# A Simple 2-Column Example

```
#content1 {
      position: absolute;
      left: 0;
      right: 50%;
}

#content2 {
      position: absolute;
      left: 50%;
      right: 0;
}
```

# *The width and height Properties*

- A block can be assigned a `width` and/or `height` property
- This can be `auto`, a percentage, or a length in inches, pixels, cm, mm, em, etc.
  - `auto` fills alls available width, and uses just enough height
  - Percentages are treated as a percentage of enclosing box
- This defines the space available *for the contents* of the box. It does not include margins, borders, or padding

```
div.menu { width: 10em; }
#firstcolumn { width: 33%; }
```

# The position Property

- The position property determines how the properties `left` and `top` are interpreted
  - `static`: The element is placed in a normal position
    - the `left` and `top` properties are not used
  - `relative`: The element is placed relative to the normal position
    - the `left` and `top` properties specify offsets from the normal position
  - `absolute`: The element can be placed anywhere within the containing block
    - The element's position can be specified with the `left`, `top`, `right`, and `bottom` properties

Carleton UNIVERSITY
Canada's Capital University

# *The left, right, top, and bottom Properties*

- For absolute placements, the `left`, `right`, `bottom`, and `top` properties specify the *distance* of a box side from its enclosing box
- **Tip:** Setting `bottom` to `0` can be used to force a box to fill the enclosing box

Canada's Capital University

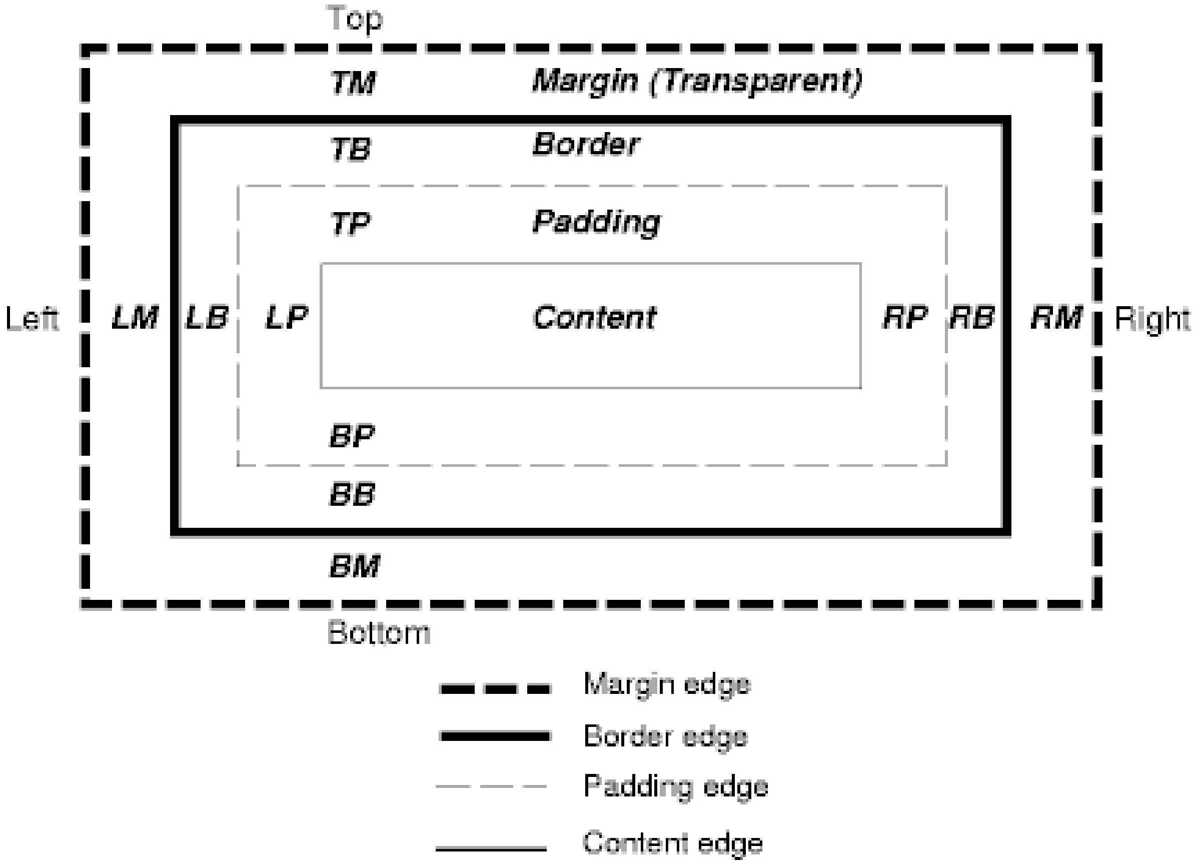
```
  #content1 {
        position: absolute;
        left: 0;
        right: 50%; bottom: 0;
  }
  #content2 {
        position: absolute;
        left: 50%;
        right: 0; bottom: 0;
  }
...

<div id=“c ontent1”>
  The left column text goes here
</div>

<div id=“c ontent2”>
  The right column text goes here
</div>
```

# The CSS Box Model

- Every block is rendered as a *box*:



Carleton UNIVERSITY
Canada's Capital University

# *Margins, Border, and Padding*

- Margins are transparent space around the outside of the box
  - margin color is determined by `background` property of *the box that contains this one*
- Border is a border around the box
- Padding is distance between the border and the contents of the box
  - padding color is determined by `background` property

Carleton UNIVERSITY

Canada's Capital University

## *The margin Properties*

- Margins are transparent space around the outside of the box
  - `margin-top`: The top margin size
  - `margin-right`: The right margin size
  - `margin-bottom`: The bottom margin size
  - `margin-left`: The left margin size
  - `margin`: Sets all four properties at once

Carleton UNIVERSITY
Canada's Capital University

# *The padding Properties*

- Padding is space between the border and the content
  - `padding-top`: The top margin size
  - `padding-right`: The right margin size
  - `padding-bottom`: The bottom margin size
  - `padding-left`: The left margin size
  - `padding`: Sets all four properties at once
- Don't be afraid to use lots of padding

Carleton UNIVERSITY
Canada's Capital University

## *The `border` Properties*

- The border is a (usually visible) border
  - `border-style`: The type of border
    - `none`, `hidden`, `dotted`, `dashed`, `solid`, `double`, `groove`, `ridge`, `inset`, `outset`
  - `border-color`: The color of the border
  - `border-width`: The width (thickness) of the border
  - `border`: Sets all three at once

- All four borders can also be set separately
  - `border-xxx-style`
  - `border-xxx-color`
  - `border-xxx-width`
  - `border-xxx`
  - Where `xxx` is one of `left`, `right`, `top`, `bottom`

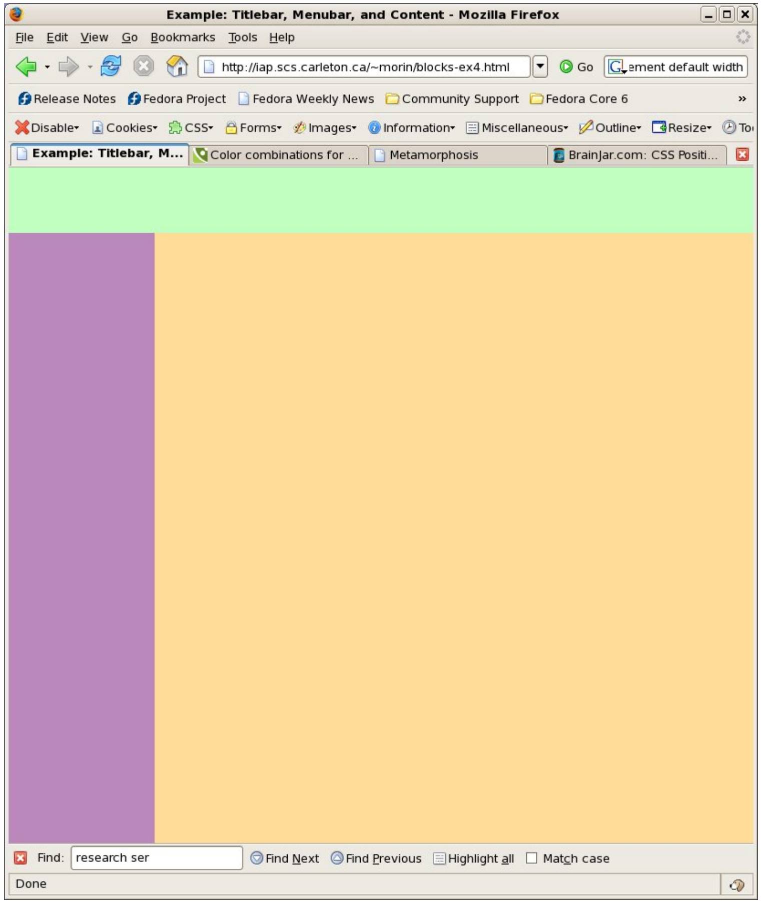
Carleton UNIVERSITY

Canada's Capital University

# Box Size

- The total width of a box is
  - margin-left + border-left-width + padding-left + width + padding-right + border-right-width + margin-right
- The total height of a box is
  - margin-top + border-top-width + padding-top + height + padding-bottom + border-bottom-width + margin-bottom
- **Tip:** Unless, the height or width is fixed, don't specify it
  - Instead, specify the top, left, right, or bottom

# Example: Titlebar, Menubar, and Content



Carleton UNIVERSITY
Canada's Capital University

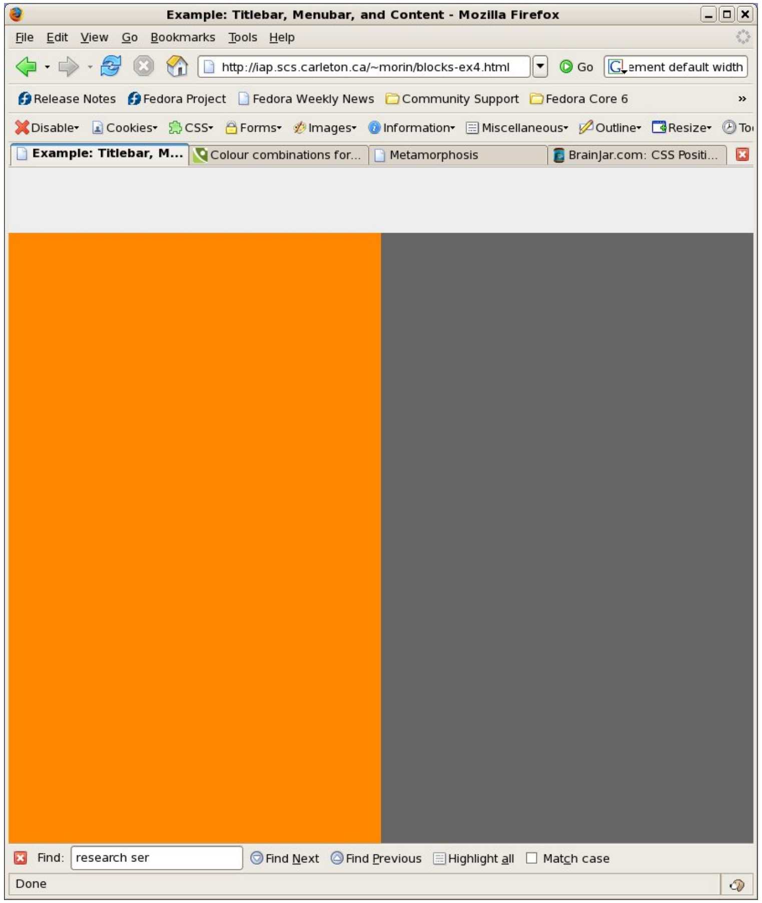
# *Example: Titlebar, Menubar, and Content*

- We use three `div` elements
- the `menubar` is fixed-width and its left and top sides are specified
- the `titlebar` is fixed-height and its top side is specified
- the `content` is auto height and width but its left and top sides match the sizes of the `menubar` and `titlebar`, respectively

```
#titlebar {
        position: absolute;
        left: 0;
        top: 0;
        right: 0;
        height: 10ex;
}
#menubar {
        position: absolute;
        width: 12em;
        top: 10ex;
        left: 0; bottom: 0;
}
#content {
        position: absolute;
        left: 12em;
        top: 10ex;
        right: 0; bottom: 0;
}
```

Canada's Capital University

# Example: Two Columns with Headline



Carleton
UNIVERSITY
Canada's Capital University

# Example: Two Columns with Headline

- Use three `div` elements
- The `header` has a fixed-height, its left, top, and right sides at distance 0
- `column1` has left side at distance 0 and right-side at 50%
- `column2` has left side at 50% right-side at distance 0
- The bottom of `column1` and `column2` are at 0 so that they line up

Carleton UNIVERSITY
Canada's Capital University

```
#titlebar {
      position: absolute;
      left: 0; right: 0; top: 0;
      height: 10ex;
}
#column1 {
      position: absolute;
      left: 0; right: 50%;
      top: 10ex; bottom: 0;
}
#column2 {
      position: absolute;
      left: 50%; right: 0;
      top: 10ex; bottom: 0;
}
```

# *Changing the display Attribute*

- The display attribute of tags can be modified
- Two common uses:
  - `display: none` can be used to hide information without deleting it from the HTML
  - The `display` attribute of list items can be used to turn lists into menus
- **Tip:** By setting `a:hover` properties you can get beautiful horizontal or vertical menu without using images

Carleton UNIVERSITY
Canada's Capital University

# A Menubar List

```
ul.menu {
      width: 6em;
      margin: 0;
      padding: 0;
      border-left: solid 1px black;
      border-right: solid 1px black;
      border-bottom: solid 1px black;
}
ul.menu li {
      list-style-type: none;
      background-color: gray;
      border-top: solid 1px black;
      text-align: left;
}
```

# A Menubar List (Cont'd)

```
<ul class=”m enu”>
  <li><a href=”e 1/”> Eggs and ham</a></li>
  <li><a href=”e 2/”> Sausage and bacon</a></li>
  <li><a href=”e 3/”> Pancakes</a></li>
</ul>
```

Carleton UNIVERSITY
Canada's Capital University

# A Horizontal List

```
ul.hmenu li {
  display: inline;
  list-style: none;
  padding-left: .5em;
  padding-right: .5em;
  margin: 0;
}
....

<ul class=“h menu”>
  <li><a href=“e 1/”> Eggs</a></li>
  <li><a href=“e 2/”> Sausage</a></li>
  <li><a href=“e 3/”> Pancakes</a></li>
</ul>
```

Carleton
UNIVERSITY
Canada’s Capital University

# *Summary*

- HTML elements are (mostly) either `inline` or `block`
  - `DIV` is the prototype `block` element
  - `SPAN` is the prototype `inline` element
- This can be changed with the `display` property
- Uses include:
  - Using `DIV` elements to do page layout
  - Using lists (`UL`) as (vertical or horizontal) menus
- **Tip:** Even recent browsers can have problems with some CSS properties