

# CSS

Making Layouts with CSS

# Where does it go?

- CSS can be included in
  - The style attribute of an element
  - The <head> tag of a HTML document
  - An external file specified in the <head> tag
- ➔ This order indicates the dominance scheme
  - Dominance absolutely applies only in certain cases

# Things to know

- CSS rules applied by specificity
  - Selectors are not overridden simply by order
  - Most specific selector for an element's property wins
- Comments `/* */`
- Not every browser renders content the same

# CSS

## Basic Layout Properties

# Layout & HTML

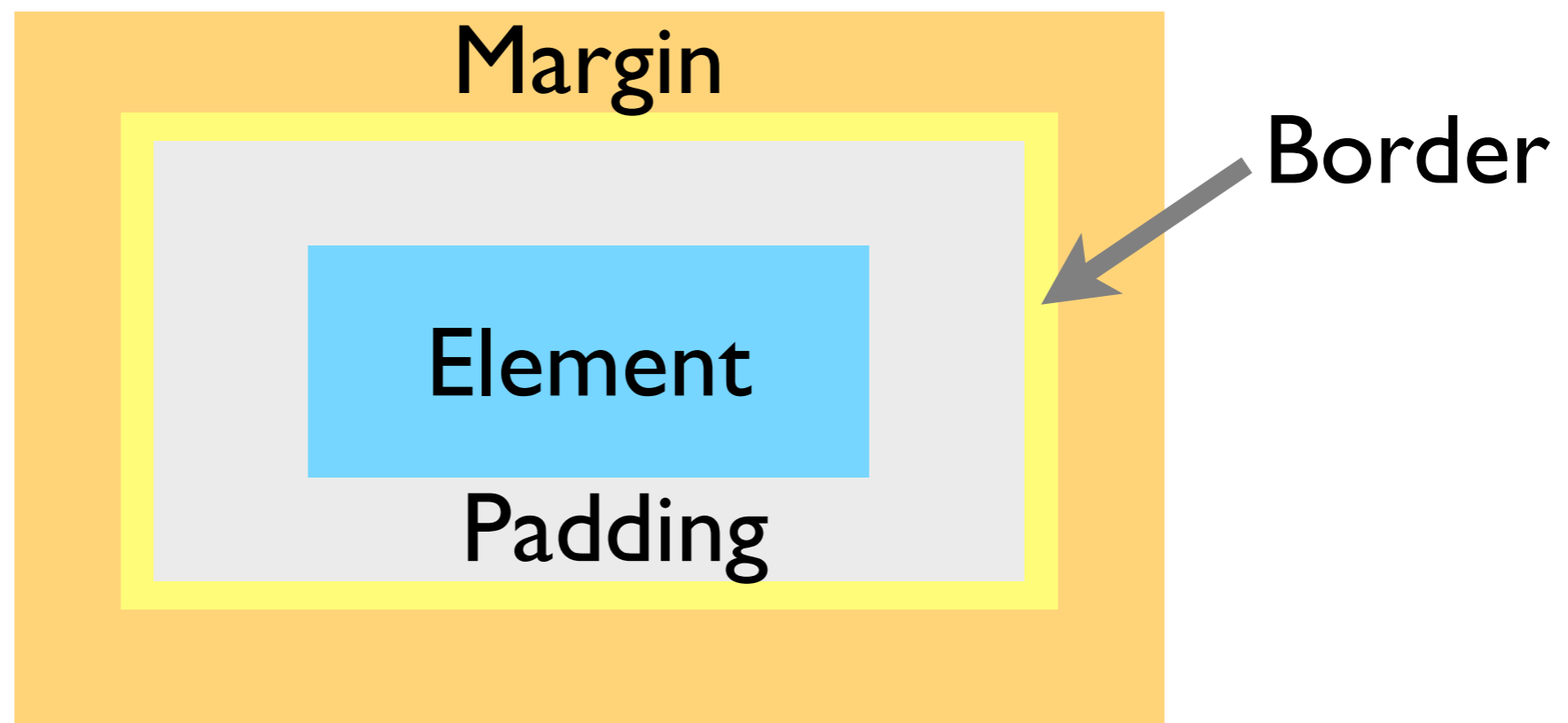
- By default, elements are rendered top to bottom, left to right
- Elements are divided into two layout categories
  - **inline:** width & height determined by contents
    - ex: `<a>`, `<img>`, `<span>`, `<strong>`
  - **block:** height determined by contents; width extends to edge of parent container
    - ex: `<div>`, `<p>`, `<table>`, `<h1>`

# Layout & HTML

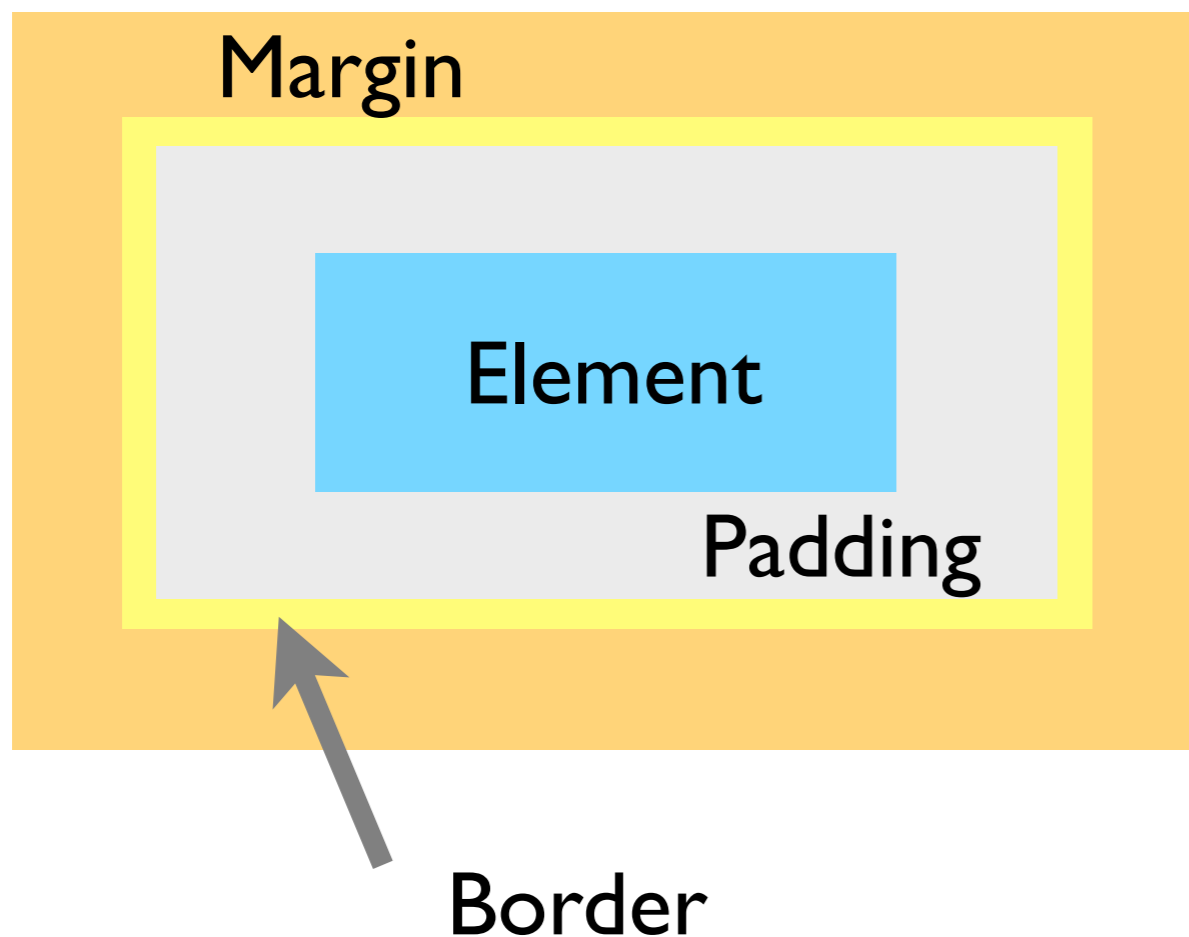
- Elements are rendered from L-to-R until a block is encountered
  - Blocks break flow of inline elements
  - ex: `<br />`
- CSS Box Model explains spacing between objects rendered according to this scheme

# Layout: The Box Model

- Describes how layout properties alter rendering
- Well-illustrated in real-time with Firebug



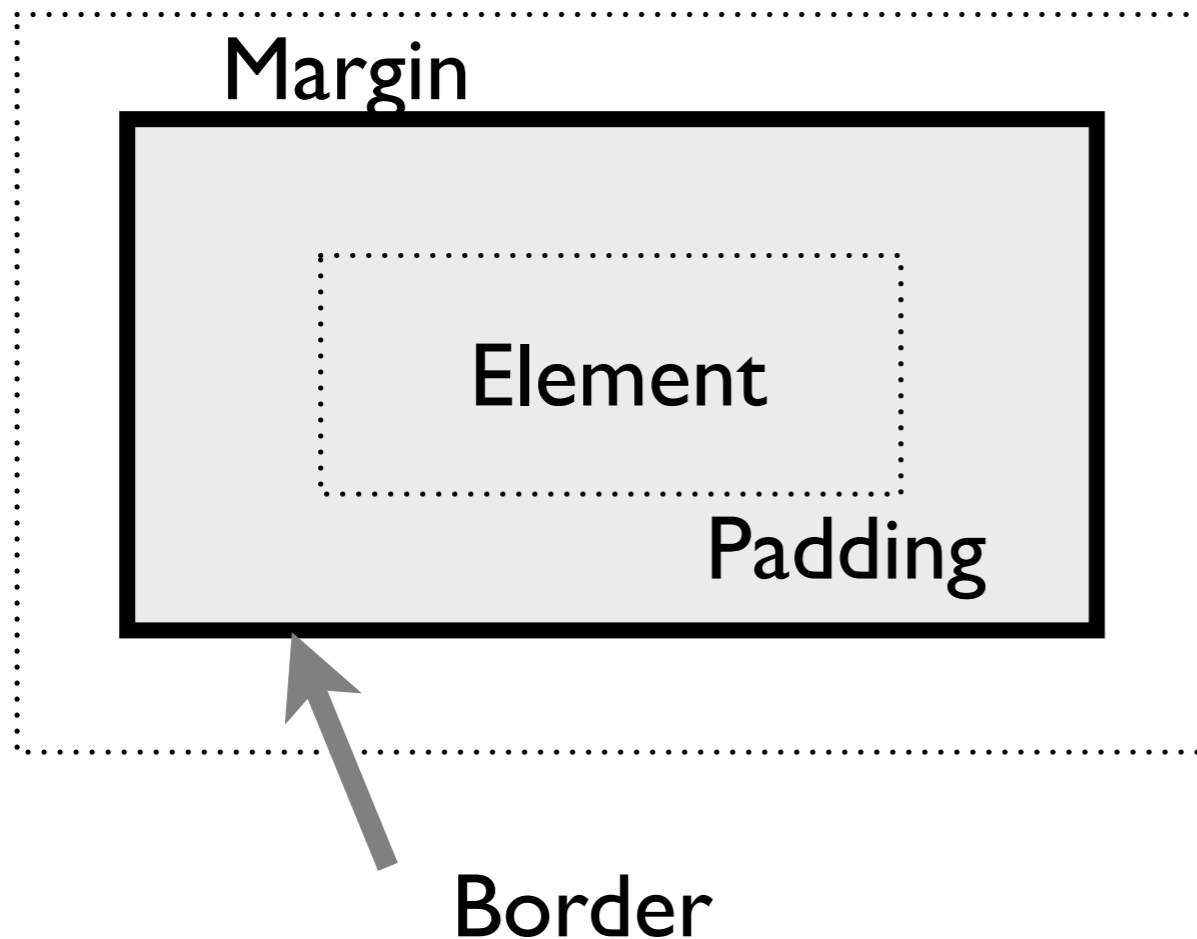
# How to change attributes



- Element
  - width, height
- Padding
  - padding macro
- Border
  - border macros
- Margin
  - margin macro



# How it's seen



- Background color extends to border
- Margin is invisible

# About Macro Properties

- Macro properties take multiple values, separated by spaces
- `border: <border-width> <border-style> <border-color>`
  - ex: `border: 1px solid #000000;`
- Each constituent value corresponds to a property that can be set itself

# CSS

## Advanced Layout Properties

# How?

- Standard layout model does not provide capabilities for side-by-side elements
  - × Use tables to layout page-- NO.
  - ✓ Use <div>s with creative CSS to layout page
    - ➔ Maintains semantics of document

! It is recommended that you view this section of the slides interactively:

[http://sipb.mit.edu/iap/webdesign/course\\_materials/lecture\\_2/floats.html](http://sipb.mit.edu/iap/webdesign/course_materials/lecture_2/floats.html)

# Using Floats

- What is it?
  - Any block element given
    - `float: left;` or `float: right;`
- Floats break from the normal layout flow
  - Rendered to either side of parent
  - Objects around them flow around it
  - ex: an inset figure in a paper

# Using Floats

- Floated `<div>`s can be used to place block content side-by-side
  - ➔ Enables traditional layout schemes
- ◆ Elements floated to the same side are rendered in order
- ◆ To interrupt aligned layout (next “row”), use
  - `clear: (left | right | both);`

# The Problem with Floats

- Usage is very common & valid, but feature was not designed for this purpose
  - ➔ Unexpected behavior (padding & margin)
- Floats must be given an explicit width
- ◆ A non-floated element containing only a float has 0 height
  - ➔ Floats (blocks) may also need a height