

# CMSC335

---

## Web Application Development with JavaScript



### CSSIII

Department of Computer Science  
University of MD, College Park

Slides material developed by Ilchul Yoon, Nelson Padua-Perez

# Review: Box Model

---

- A block-level element (e.g., p) contains four edges (top, bottom, right, and left) defining a box
- Every box has three available properties that can be adjusted to control its appearance:

**border**

margin

padding



# Review: Box Model

---

- The width and height of an element with CSS is only for the **content area**. Add padding, borders, and margins to calculate the full size of an element

border

margin

padding



- Tutorial
  - [https://www.w3schools.com/css/css\\_boxmodel.asp](https://www.w3schools.com/css/css_boxmodel.asp)

# Width and Height Properties

---

- Box width
  - left + right padding, left + right border, left + right margin, content width
  - **width** property - sets the **content** width
- Box height is determined in the same way
  - **height** property - sets the **content** height
- **Example:** WidthHeight.html

# Border/Padding/Margins are NOT inherited

---

- The following CSS properties are not inherited by default
  - margin
  - border
  - padding
  - background
- Property values for a block element are **not** inherited to their child block-level elements

# display property

---

- **display property**
  - Defines the type of rendering box (e.g., block, inline) of an element
  - **Values**
    - » **inline** - causes a block-level element to act like an inline one
    - » **block** - causes an inline element to act like a block-level one
    - » **inline-block** - causes a block-level element to flow like an inline one while retaining other features of a block-level element
    - » **none** - hides an element from the page
    - » **flex** - displays element as a block-level flex container
    - » **grid** - displays element as a block-level grid container
  - **Reference:**  
[https://www.w3schools.com/cssref/pr\\_class\\_display.php](https://www.w3schools.com/cssref/pr_class_display.php)
- **Example:** DisplayInlineBlockNone.html

# float property

---

- **CSS Normal document flow/normal position** - Placing of elements one after another based on the document structure and whether the element is an inline or block element
- **float**
  - Places an element on the left or right side of the container, enabling text and inline elements to wrap around it
  - **Values**
    - » none, left, right
- **Example:** FloatI.html
- **Example:** FloatII.html
  - Creating a layout using floats

# position property

---

- “**position**” property can be set to:
  - **sticky** (complicated, not supported by many browsers)
  - **static** (normal position—no effect)
    - » By default, the position property is set to **static**
    - » **Example:** PositionStatic.html
  - **relative** (adjust relative to normal position)
    - » Space occupied by the element in normal flow is still retained
    - » If you make the window smaller, you will see scroll bars
    - » **Example:** PositionRelative.html
  - **fixed** (fixed position in the viewport (browser window), even when scrolling)
    - » Space occupied by the element in normal flow is still retained
    - » If you make the window smaller, you will NOT see scroll bars
    - » **Example:** PositionFixed.html



# Absolute Positioning

---

- **absolute** positioning
  - Current element is placed in relation to the containing block
  - The containing block is not necessarily the immediate parent block
- Rules for determining the containing block:
  - Nearest ancestor of the element that has a position property value set to something other than **static**
  - If no ancestor has a position property set to something other than static, then the containing block is the body block
- **Example:** PositionAbsoluteContainingBlockBody.html
- **Example:** PositionAbsoluteContainingBlockDiv.html

## z-index property

---

- “**z-index**” property is used to specify which elements are “in front” when they overlap (default value = 0). You can use any values (e.g., 5, 15, 20); the largest one represents the element on top
- **Example:** PositioningZIndex.html

# flex display property

---

- **Example:** DisplayFlex.html

# Miscellaneous Properties

---

- **text-indent** - specifies the indentation to use for the first line of a block
- **line-height** - height of a line of text
- **letter-spacing** - space in-between letters
- **word-spacing** - space in-between words