

# Announcements

- ❖ Check class announcements daily

# Writing Computer Programs

❖ Please read the document:

<http://www.cs.umd.edu/~nelson/documents/SuggestionsForWritingComputerPrograms.htm>

# 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

# 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 and width determined in the same way

- ❖ **height property** → sets the content height

- ❖ **Example:** widthHeight.html/widthHeight.css

# Block/Inline Elements

## ❖ **Block Element**

- ❖ Displayed as a block of content starting and ending with a new line.
- ❖ Examples: `<p>`, `<div>`, `<h1>` → `<h6>`, `<table>`
- ❖ Listed one after another VERTICALLY down the page.
- ❖ They will stretch across the whole page unless constrained in some way
- ❖ **Block elements** generate a **principal block box**

## ❖ **Inline Element**

- ❖ Does not start new lines
- ❖ Contained within the flow of text.
- ❖ Appear one after another in a line, HORIZONTALLY across the page
- ❖ **Inline elements** generate an **inline box**
- ❖ **Example:** `blockInline.html`
- ❖ **display property** → Enables you to change the type of an element.
- ❖ **Example:** `changingDisplayProp.html`

# Position Property

- ❖ The **position** property along with the **top**, **right**, **bottom**, **left** properties allow us to manipulate the position of elements
- ❖ Four possible positioning schemes
  - ❖ ***fixed positioning*** → the element is fixed (does not move) and it is placed with respect to the viewport (e.g., browser window)
  - ❖ ***static positioning*** → default positioning where each element is laid out one after another (“normal flow”)
  - ❖ ***relative positioning*** → the element is placed in a position relative to the position where it will appear in static positioning
  - ❖ ***absolute positioning*** → the element is placed in an absolute position within the containing block (positioning context)

# Fixed Positioning

- ❖ **Example:** fixed.html
- ❖ top/right/bottom/left → Distance from appropriate side
- ❖ In the example the navigation menu and the paragraph in red are fixed
- ❖ If you reduce the window size the paragraph (in red) does not move and sometimes you will not be able to see it
- ❖ Useful when you want to have an element in a fixed position of the web page (e.g., menu or back button).

# Relative Positioning

- ❖ **CSS Normal Document Flow** → Placing of elements one after another or within another element based on the document structure and whether the element is an inline or block element
- ❖ **Relative Positioning**
  - ❖ You can move the element relative to its normal position in normal flow
  - ❖ Space occupied by the element in normal flow is still retained
- ❖ **Example:** relative.html

# Absolute Positioning

- ❖ **Containing Block (Positioning Context)**
  - ❖ Element with respect you are positioning the current element
- ❖ Default containing block is the body block
- ❖ **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 different from static
  - ❖ If no ancestor has a position property set then the containing block is the body block
- ❖ Setting the position property of an element to relative and not providing any offsets (e.g., top, etc.) makes the element a containing block for its descendants
- ❖ **Example:** absolute.html

# float Property

- ❖ float property
  - ❖ A box is laid out according to normal document flow
  - ❖ The box is then taken out of the flow and shifted to the left or right as far as possible
- ❖ Values:
  - ❖ right → floats box to the right and content to the left
  - ❖ left → floats box to the left and content to the right
- ❖ **Example:** float.html, float.css (floating inline element img)
- ❖ Using the float property instead of align attribute in img
- ❖ The float property can be used with elements other than images
- ❖ Provides an alternative to table-based layout
- ❖ Float property and block elements
  - ❖ **Example:** floatBlock.html, floatBlock.css

# Favicon

- ❖ Icon that appears next to the websites' URL in the address bar.
- ❖ Example: <http://www.umd.edu/>
- ❖ You also see them when you bookmark a page
- ❖ The icon is represented by a file named favicon.ico
- ❖ In the resources section of the class web page you can find a link to a favicon generator
- ❖ To place the favicon in your page insert the following html tag in the head section of your html file (assuming file name for favicon is favicon.ico)

`<link rel="shortcut icon" href="favicon.ico" />`

- ❖ **Example:** FaviconExample.html

# Like Facebook Button

- Just cut and paste the following in your html document:

```
<iframe  
  src="http://www.facebook.com/widgets/like.php?href=http://PutAW  
ebAddressHere" scrolling="no" frameborder="0"  
  style="border:none; width:250px; height:75px">  
  
</iframe>
```