ANNOUNCEMENTS

- You must implement programming projects by yourself.
CONVENTIONS TO USE FROM NOW ON

- **Variable/Function names**
  - We will use lowercase
  - If multiple words are associated with a variable name then capitalize the first letter of second word on
    - waterTemperature
    - globalWarmingIndex

- **Curly Brackets**
  - Use a particular style

- **Comparisons**
  - Use `===` rather than `==`

- **No global variables!**

- **Good indentation**
TESTING/DEBUGGING

- **Testing**
  - Remember to test your code as you develop it.
  - First use simple data sets.

- **Debugging**
  - Make sure your input is correct.
  - Use Error Console to track down problems.
  - Use lint to check the syntax.
  - Use alert to display values and identify the execution path.
  - Use trace tables.
GLOBAL VARIABLES

- Global Variables – variables defined outside of any function
- We want to avoid using global variables. Why?
FUNCTIONS AS DATA

- In JavaScript functions are considered data.
- That means they can be assigned to variables, passed as arguments to functions, etc.
- Example: FunctionsAsData.html
As we saw we can access data in forms by using 

\[
document.getElementById("elementId")
\]

g getElementByld returns a reference to an element that we can use to:

- Retrieve the value of the element (e.g., text field in a form)
  
  \[
  var login = document.getElementById("loginId").value;
  \]

- Set the function to call when an element is clicked on (e.g., button)
  
  \[
  document.getElementById("processButton").onclick = functionDoesProcessing;
  \]

- Get/Set Attributes
  
  \[
  var imageElement = document.getElementById("myImage");
  var imageName = imageElement.getAttribute("src");
  imageElement.setAttribute("src", "imageFile.jpg");
  \]

**Example:** PhotoViewer.html

- Will it work with [http://www.cs.umd.edu/assets/images/frontpage/vm.jpg](http://www.cs.umd.edu/assets/images/frontpage/vm.jpg) ?

**Example:** Animation.html