

General Guidelines

- *Do not* restate the code and *do not* include what is obvious.
ex: `var firstName; // declares a variable named firstName`
- *Do* include knowledge that is assumed or not inherently obvious. Do use variable, function and object names that show meaning, while still remaining succinct. In the following example we are storing the user's mother's first name:
Ok: `var fName; // Global, String.`
Better: `var fName; // Global, String. Stores the first name of the user's mother`
Best: `var motherFName; // Global, String. Stores the first name of the user's mother`
- *Do* comment as you code. *Do not* wait until you are finished.
- *Do* format your code. Insert a blank line before multiline comments and use tabbing to increase readability.

File Headers

For the purposes of class, include following header information:

- the author's name
- the name of the class
- the week the work was assigned
- the location of the file on the web (if applicable)
- a *brief* statement of the contents of the file

```
/*  
 * @author Philip Likens  
 * @description Programming Logic  
 * Week 6  
 * http://deptof.com/aid/index.php  
 * A Javascript Calculator  
 */
```

Functions

When commenting functions include a summary (what purpose does the function serve?), what parameters the function takes (input), and what the function will return (output), including data types.

```
/*  
 * @description Generates a random value between 0 and n, not including n  
 * @param n is the number of possible random values, starting with 0  
 * @returns an integer  
 */  
function rand ( n )  
{  
    return Math.floor( Math.random() * n );  
}
```

Variables

When commenting variables, always describe the role of the variable in the program or function. Additionally, include the scope and data type of the variable being declared.

```
var mathSymbol; // Global, Number. Stores the current math symbol being used in the calculation
```

Other Code

Comment any other code that does not intuitively make sense. As a rule, if you might forget the function of the line(s) of code within the next week, add comments.

```
// if the client's username and their mom's first  
// name match, show the user an error  
if ( firstName == userName )  
{  
    alert("hello world");  
}
```

Commenting Code

Philip Likens

Revisions

If you make revisions to a previously saved file, include commentary in the header detailing the nature of the revisions, and be sure to update any relevant commentary in the body of your code. *Note: If using a Revision Control System, this is considered bad practice. But since we are not using a Revision Control System, these comments will help document how the code has changed.*

Tags

Tags provide a way for a programmer to quickly communicate the status of a line or section of code.

FIXME to mark potential problematic code that requires special attention and/or review.

NOTE to document inner workings of code and indicate potential pitfalls.

TODO to indicate planned enhancements.

XXX to warn other programmers of problematic or misleading code.

See Also

http://class.ee.iastate.edu/cpre211/labs/commenting_guide.html

<http://particletree.com/features/successful-strategies-for-commenting-code/>

<http://klauskomenda.com/code/my-javascript-coding-guidelines-and-standards/>

<http://dev.opera.com/articles/view/efficient-javascript/>

<http://www.oracle.com/technetwork/java/javase/documentation/index-137868.html>

Thank you to Erik Shreve, Chad Hardy and the Art Institute of Dallas WDIM PAC for their input.