ANNOUNCEMENTS

- Project #2 posted. Do not wait to start working on it.
- No posting of code in the forum.
- You must implement programming projects by yourself.

WHILE STATEMENT

- **while statement** – Control statement which allows JavaScript to repeat a set of statements.
- **Basic Form**

  while (expression)

  statement // executed as long as expression is true
- If you want to execute more than one statement then use a set of { } to enclose the statements.
- You can have other types of statements (including whiles) in a while.
- **Example:** SqrtTable.html

COMBINATION OF STATEMENTS

- Keep in mind that you can have any combination of conditionals, and iteration (while) statements.
- For example:
  - Conditionals insides of loops.
  - Conditionals inside conditionals.
  - Loops inside conditionals.
  - Loops inside of loops.

TRACE TABLES

- Mechanism to keep track of values in a program.
- Allows you to understand the program behavior.
- We could create a trace table for SqrtTable.html

INFINITE LOOPS

- An infinite loop occurs when the expression controlling the loop never becomes false.
- **Example1**

```
int x = 30;
while(x > 0)
   document.writeln("<li>Element</li>");
```

- **Example2**

```
 int x = 7;  // how about x = 8
 while (x != 0)  {
    document.writeln("<li>Element</li>");
    x=x – 2;
 }
```

- How can we detect infinite loops?

PROGRAMMING ERRORS

- **Syntax Error**: (Compile-time error) The program violates the language's grammar.
- **Semantic Error:** The program fails to accomplish what we want.
- **Debugging:** The process of finding and fixing errors. Extremely hard for large software systems. Tools for debugging:
  - Trace tables
  - Output statements
  - Debuggers
- **Analogy:**
  - Taco tom ate.→ Syntactically therefore semantically incorrect.
  - A taco ate tom.→ Syntactically correct however semantically incorrect.
  - Tom ate a taco → Syntactically and semantically correct (what we want!)

HOW TO FIND PROBLEMS IN YOUR CODE

- Computer programming is not about writing code and letting someone else find the problems with it.
- You have to learn how to find problems in your code.
- Approach
  - Use Error Console to see possible error
  - Use JavaScript Lint
- How about logical errors?

INTRODUCTION TO DEBUGGING

- How to debug your code?
- Introducing the alert() function

SUGGESTIONS FOR SOLVING PROBLEMS USING A PROGRAMMING LANGUAGE

- **Design-** Make sure you first come up with a plan/design for your code (e.g., using pseudocode).
- **Do not wait until the last minute** – Code implementation can be unpredictable.
- **Incremental code development** – Fundamental principle in computer programming. Write a little bit of code, and make sure it works before you move forward.
- **Don't make assumptions** – If you are not clear about a language construct write a little program to familiarize yourself with the construct.
- **Good Indentation** – From the get-go use good indentation as it will allow you to understand your code better.

SUGGESTIONS FOR SOLVING PROBLEMS USING A PROGRAMMING LANGUAGE

- **Good variable names** – Use good variable names from the beginning (do not use x and y and then change them to meaningful names before submitting the project).
- **Testing**
  - Test your code with simple cases first.
  - Test as your develop your code.
- **Keep backups** – As you make significant progress in your development, make the appropriate backups
- **Trace your code**
- **Use a debugger**
- **Take breaks** – If you cannot find a bug take a break and come back later.