ANNOUNCEMENTS

- No posting of code in the forum.
- You must implement programming projects by yourself.
JAVASCRIPT (LOGICAL OPERATORS)

- Used with comparison operators to create more complex expressions.
- Operators
  - Logical and (&&) – expr1 && expr2
    - Expression is true if and only if both expressions are true otherwise is false.
    - **Example:** LogicalOp1.html
  - Logical or (||) – expr1 || expr2
    - Expression is false if and only if both expressions are false otherwise is true.
    - **Example:** LogicalOp2.html
  - Logical Not (!) – !expr
    - Inverts the boolean value of the expression.
Remember you can use parenthesis to impose a particular order for the evaluation of an expression.
You can combine if statements to handle different cases.
This approach to organize if statements to handle different cases is called the **Cascaded If Statement**.
Cascaded If statement general form:

```
If (expr1)
    // Statement is executed if expr1 is true
else if (expr2)
    // Statement is executed if expr2 is true
else if (expr3)
    // Statement is executed if expr3 is true
else
    // If none of the above expressions is true
```

Notice it is not a JavaScript statement.
Once one of the cases is executed no other case will be executed.
You can use {} to enclose more than one statement.
**Example:** See CascadedIf.html
WHILE STATEMENT

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

- **Basic Form**

  ```javascript
  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
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