ANNOUNCEMENTS

- No posting of code in the forum
- Check class announcements daily
JAVASCRIPT (IF STATEMENT)

- What is the difference between === and ==?
- Let’s compare
  - 20 === “20” vs. 20 == “20”
JAVASCRIPT (LOGICAL OPERATORS)

- Used with comparison operators to create more complex expressions
- Operators
  - Logical and (&&) \( \rightarrow \) expr1 && expr2
    - Expression is true if and only if both expressions are true otherwise is false
    - Notice you need two &
    - You can have more than two expressions
    - Example: LogicalOp1.html
  - Logical or (||) \( \rightarrow \) expr1 || expr2
    - Expression is false if and only if both expressions are false otherwise is true
    - The | character can be found on top of the Enter key in the keyboard (same key as \( \backslash \))
    - You can have more than two expressions
    - Example: LogicalOp2.html
  - Logical Not (!) \( \rightarrow \) !expr
    - Inverts the boolean value of the 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:

```plaintext
If (expr1) {
    // Statements executed if expr1 is true
} else if (expr2) {
    // Statements executed if expr2 is true
} else if (expr3) {
    // Statements 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 don’t need to use {} if you only have one statement
More efficient than having multiple if statements
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(s)  // executed as long as expression is true
}
```

- `{ }` not needed if you only have one statement
- You can have other types of statements (including whiles) in a while
- Common mistake: to add a semicolon after closing parenthesis

**Example:** Numbers.html
**Example:** EvenNumbers.html
**Example:** NumbersTable.html
**Example:** SqrtTable.html

- To develop this program we first generated a list of values and square root values (no HTML)
- Let’s see this version of the program without HTML
ESCAPE SEQUENCE

- In the previous examples we had:
  
  "<table border="1">"

- In the previous examples we had:
  
  If we use “ before the 1 rather than \” JavaScript will think the end of the string occurs before the 1
  
  By using \” the “ is treated as a normal character (e.g. letter a)
  
  \ before a special character creates a escape sequence
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 EvenNumbers.html
COMBINATION OF STATEMENTS

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