Control Structures

Chapter 3
Objectives

• Use iteration statements to make repetition
• Use **while** statements, **do**...**while** statements, and **for** statements to execute code repeatedly
• Use **continue** statements to restart a looping statement
Repeating Code

- Loop statement repeatedly executes a statement, or a series of statements
  - while a specific condition is true or until it becomes true
- while statements
- do...while statements
- for statements
while Statements

• Repeats a statement or series of statements as long as conditional expression evaluates to true

while (conditional expression) {
  statement(s);
}

Example: `while`

- Counter is a variable that increments or decrements with each iteration of a loop statement

```javascript
var count = 10;
while (count > 0) {
    document.write(count + "<br />");
    --count; // or count = count - 1;
}
document.write("<p>Liftoff!</p>");
```
Example `while`... 

**Figure 3-10**
Output of a `while` statement using a decrement operator

```
10
9
8
7
6
5
4
3
2
1

We have liftoff.
```
Infinite loop

• A loop statement that never ends because its conditional expression is never false

• Possible uses:
  – Events, better handled by raising an event
  – Force user to abort program
do...while Statements

• Executes statements once
• Repeats the execution as long as conditional expression evaluates to true

```java
do {
    statement(s);
} while (conditional expression);
```
Example: do...while

```javascript
var count = 2;
do {
    document.write("<p>The count is equal to " + count + "</p>");
    ++count;
} while (count < 2);
```
for Statements

• Repeats a statement or series of statements as long as conditional expression evaluates to true

• Can also include code that initializes a counter and changes its value with each iteration
for Statements...

for (counter declaration and initialization; condition; update statement) {
    statement(s);
}


Example for

```javascript
for (var count = 0; count < 10; count++) {
    document.write("<p>" + count + "</p>");
}
```
More elaborate example

This text appears in black
This text appears in green
This text appears in red
This text appears in grey
```javascript
var colors = new Array(4);
colors[0] = "black";
colors[1] = "green";
colors[2] = "red";
colors[3] = "grey";
for (var i=0; i < colors.length; i++) {
    document.write("<h1 style="color:"+colors[i]+"">"+ "This text appears in "+colors[i]+"</h1>");
}
```
Example 2

• Prompt user for number
• Print that number of lines
var numberStr = window.prompt("Enter the number of iterations");
var number = parseInt(numberStr);
if (isNaN(number))
    window.alert("ERROR! Not a number!");
else
    for (var i=0; i < number; i++)
        document.write("Counting:" + (i+1) + "<br />");
continue Statements

• Skips remainder of a command block in looping statement and restarts the loop with a new iteration

• When you want to stop a loop for the current iteration, but want the loop to continue with a new iteration
**break** Statement in Loops

- Possible to use `break` statement in loops
- Interrupts loop
- Should be *avoided* since "breaking" should be done based on the condition statement
- `break` vs `continue`
  - `continue` skips only current iteration
  - `break` also skips all following iterations
while vs. for

• No difference in iteration between do, while and for
• But do executes a minimum of once, while and for may execute zero times
• while preferable to for where loops do not need to declare, initialize, or update a counter
Exercise

On paper write code that does the following:

• Create an array with 5 bank balances
• Calculate and add a yearly interest rate of 4.3% to each of the bank balances (using a loop)
• Display the resulting balances
Summary

• Loop statement is a control structure that repeatedly executes a series of statements
  – while statements
  – do...while statements
  – for statements
Summary

• Each repetition is called an iteration
• An infinite loop never ends
• The `continue` statement halts a looping statement and restarts the loop with a new iteration