Once Is Not Enough

Repeating instructions is the source of great power in computing

© 2006, Lawrence Snyder
Announcements

Project 2, Part A, Step 3. Aligning the `<input ...>` items in the center is OK.
“Iteration” is another term for “repeat”

- Iteration doesn’t suffer from the question of whether the first item is counted … in iteration it always is. (Use “repeat” and “iterate” interchangeably unless it matters.)
- Iterating is usually called “looping” in programming
- Programming languages have many kinds of statements to help program loops
- In JS we will use the `for`-statement
Sample for-statement

```c
for ( i=0; i<7; i++ ) {
    <stuff to be repeated>
}
```

- Where to start counting
- Where to stop counting. Number of “reps”
- Add 1
Sample for-statement

for-statements repeat

for (i=0; i<7; i++) {
  <stuff to be repeated>
}

For example ...

for (i=0; i<7; i++) {
  document.write("<img src=RedBox.gif>");
}


Anatomy of \texttt{for}

The \texttt{for}-statement syntax

\begin{verbatim}
for ( \langle initialize\rangle; \langle continue\ test\rangle; \langle next\ iteration\rangle ) 
  \langle statement\ list\rangle
\end{verbatim}

\texttt{for}'s 3 control specifications -- the "control trio" -- are connected by an iteration variable

\begin{itemize}
  \item \langle initialize\rangle -- gives iteration variable its first value
  \item \langle continue\ test\rangle -- this test is performed before starting each cycle of loop; if false, quit
  \item \langle next\ iteration\rangle -- the change to the iteration variable after each cycle
\end{itemize}
An Iteration

iterations can count ...

```html
<html><head><title>Test Page</title></head> <body>
<script language="JavaScript">
    var i, text = ""; // Initialize text to empty string
    for (i=1; i<=5; i=i+1) {
        text = text + "Iteration no.: " + i + "\n";
    }
    alert(text);
</script></body>
</html>
```

Newline in JS
Iterations can replicate other things...

It is possible to make it a lot funnier by changing the limit variable to, say, \(i \leq 1000\)
Key Points of Loops

The most important features of loops:

• The starting value of the iteration variable
• The ending value of the iteration variable
• The amount the iteration variable changes

* As explained in the book, it is possible to completely control these features by properly setting the “control trio,” but programmers have gotten in the habit of writing a single kind of iteration: WFI
World Famous Iteration

To loop \( n \) times the WFI has this form:

```java
for ( i=0; i<n; i++) {
    <statement list>
}
```

Advantages:
- Fast to type
- The number of iterations is the number after `<
- 0-origin makes it handy for most computations

WFI starts at 0, steps by 1, stops (before) \( n \) 0, 1, 2, ..., \( n-1 \)

Same as \( i=i+1 \)
"Off By 1" Error

The most common error when working with iterations is to miscount by 1

- Everyone makes this mistake
- A common place where the "off by 1" error matters is in how many times a loop loops
- The importance of the WFI is it tells exactly

```c
for ( i=0; i<n;  i++) {
    <statement list>
}
```
Using Iteration In JS

Print out a row of things

```html
<html>
<head>
<title>Test Page</title>
</head>
<body>
<script language="JavaScript">
    var j;
    for (j=0; j<5; j++) {
        document.write('[' + j + ']');
    }
</script>
</body>
</html>
```
Doubly Nested Loop

A loop within a loop repeats repetitions

```
<html><head><title>Test Page</title></head><body>
<script language="JavaScript">
  var i, j;
  for (i=0; i<3; i++) {
    for (j=0; j<5; j++) {
      document.write('[ ' + i + ',' + j + ' ]');
    }
  }
</script></body></html>
```
Arrays and Indexes

We know about names with multiple instances: Rocky 3, QE 2, John Paul 2

- The number is called the name’s index
- The least index is called the index origin
- In programming, variables that can be indexed are called arrays
- Declare arrays in JavaScript:
  `var <identifier> = new Array (<num elements>);`
- JavaScript arrays are 0-origin
- Reference array elements w/ brackets: A[0]
Arrays and Loops

Loops and arrays work together

• Declare an array and initialize elements to 8
  ```javascript
  var j, A = new Array(5);
  for (j=0; j<5; j++) {
    A[j] = 8;
  }
  ```

  WFI and array’s indices both start at 0

  Notice what would change to have 1000 elements -- arrays and loops give power
Iteration is very powerful because a small amount of code specifies a lot of computation

- `for` gives full range of looping limits, steps
- Use any form of `for` that works, but using the WFI is a good habit to adopt
- In a doubly nested loop one iteration has another iteration as its `<statement list>`
- Arrays are variables with many elements that are referred to by their index