Control Flow

INFO/CSE 100, Autumn 2004
Fluency in Information Technology

http://www.cs.washington.edu/100

Readings and References

• Reading
  » Fluency with Information Technology
    • Chapter 21, Iteration Principles

if statement in Simple Sample GUI

```javascript
function setResults(resultString) {
  var tempString = resultString;
  if (document.getElementById("radioLC").checked) {
    tempString = tempString.toLowerCase();
  } else if (document.getElementById("radioUC").checked) {
    tempString = tempString.toUpperCase();
  }
  document.getElementById("resultField").value = tempString;
}
</script>
```

- the `setResults(string)` function is called by several event processors
- in every case, it takes the string that it is given, decides if upper or lower case is desired, and sets the `resultField` accordingly

The if / else statement

The if statement is a conditional statement

  » a conditional expression is evaluated as being true or false
  » the expression is a boolean expression (ie, returns true or false)
  » if the condition is true, then one set of statements is executed
  » if the statement is false, then a different set of statements is executed

```javascript
if (<boolean expression>) {
  <statements>
} else {
  <statements>
}
```
Examples

```javascript
if (count == 0) {
    ready = false;
} else {
    ready = true;
    count = count-1;
}
```

What is the conditional expression?
What statements are part of the true block?
Which statements are part of the false block?
What happens when count is 21? 0? -1?

```javascript
if (pageCount >= 100) {
    alert("This may take a few minutes.");
}
```

What is the conditional expression?
Which statements are part of the false block?
What statements are part of the true block?
What happens when pageCount is 21? 100? 200?

W3Schools TryIt Editor

A Fancier Example of a GUI program

a single tall latte, what a great way to start the morning
An if statement from bean.html

```html
<html>
<head>
<title>Interactive Coffee Cost Calculator</title>
<script type="text/javascript">
function refresh() { 
    var shotCount; // number of espresso shots
    var cupSize; // size of the cup in ounces
    var drink; // name of the requested drink
    var price; // calculated price of the drink
    var taxRate = 0.087; // Seattle retail tax
    var element; // the current gui element (radio button)

    for (var i=0; i<document.getElementById("shotForm").elements.length; i++) {
        element = document.getElementById("shotForm").elements[i];
        if (element.checked) {
            shotCount = parseInt(element.value,10);
        }
    }
    ...
}
</script>
</head>
<body>

Iteration

• Iteration or looping is a way to execute a block of program statements more than once
• we will use the for statement to create loops
  » The for loop is generally controlled by counting
  » There is an index variable that you increment or decrement each time through the loop
  » When the index reaches some limit condition, then the looping is done and we continue on in the code

Why do we want loops in our code?

• Do something for a given number of times or for every object in a collection of objects
  » for every radio button in a form, see if it is checked
  » for every month of the year, charge $100 against the balance
  » calculate the sum of all the numbers in a list
  » etc.
• Many loops are counting loops
  » they do something a certain number of times

The for loop

A counting loop is usually implemented with for

```javascript
for (var i=0; i < count; i++) {
    document.writeln("<br>index value is : "+i);
}
```
**for example**

- **i++ is a shortcut**
  - *for (i=0; i < count; i++)*
  - at the end of every pass through the *for* loop body, do the following:
    - get the value of *i*
    - increment *i*
    - store the incremented value
  - Used as it is here, this is the same as writing
    - *i = i + 1*

**body of loop may not execute at all**

- Notice that depending on the values of the control variables, it is quite possible that the body of the loop will not execute at all

```javascript
var itemCount = 0;
...
for (var i=0; i < itemCount; i++) {
    document.writeln("<br>..processing item "+i);
}
```

**loop body skip**

- check for limit condition
  - `itemCount` is 0 when we get here, so
  - `i<itemCount` is immediately false and the loop body is skipped completely
“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
  - One advantage of a simple loop control statement is that it's easier to tell how many loops there will be

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

Number of iterations

Another Example from the iCCC

A double tall Cappuccio, what a great way to start the afternoon

A for loop from bean.html

```html
<html>
<head>
<title>Interactive Coffee Cost Calculator</title>
<script type="text/javascript">
function refresh() {
  var shotCount; // number of espresso shots
  var cupSize; // size of the cup in ounces
  var drink; // name of the requested drink
  var price; // calculated price of the drink
  var taxRate = 0.087; // Seattle retail tax
  var element; // the current gui element (radio button)

  for (var i=0; i<document.getElementById("shotForm").elements.length; i++) {
    element = document.getElementById("shotForm").elements[i];
    if (element.checked) {
      shotCount = parseInt(element.value,10);
    }
  }
...
```

arrays

- On the previous page, we are selecting one element from a collection of elements
- this collection is an array named `elements`
  - one entry for each radio button in the shotForm
  - the length of this array is available
```javascript
document.getElementById("shotForm").elements.length
```
  - we retrieve an individual element using the index variable
```javascript
element = document.getElementById("shotForm").elements[i];
```
  - The index of the first element is 0