

Form Validation

Useful Methods and Properties

Strings

```
// Gets the length of a String  
string.length
```

```
// Gets a character at a specific index from a String  
string.charAt(index);
```

Regular Expression Object

```
// Create a regular expression object  
var regexObj = new RegExp(expression_pattern);  
  
// The test method of a regular expression object return  
// true if the argument matches  
// the expression pattern. It returns false if it doesn't.  
regex.test(valueToTest);
```

More about the RegExp object at http://www.w3schools.com/jsref/jsref_obj_regexp.asp

Note: expressions can be found by searching online. It's recommended that you not spend time trying to create your own.

Select Element

```
// Gets the selected option's index  
selectElement.selectedIndex;
```

```
// Gets a reference to an option within the select element  
selectElement.options[index]
```

Determine an Element's Type

```
// Gets the type of a form field element  
element.type;
```

For example, if the element reference is to a text field, its type attribute would be "text". To determine

the types of other form elements, output their type attribute in the console.

Radio Button Element

```
// The checked attribute of a radio button object can
// be used to determine if it has been selected
// checked equals true when it is selected
// checked equals false when it is not selected
radioElement.checked;
```

Modifying Element Styles

Syntax:

```
element.style.styleName = "value"
```

Example:

```
element.style.visibility = "hidden"; // (hidden | visible)
element.style.display = "none"; // (none | block)
```

Parent Node Reference

```
// Gets a reference to the element's parent node
element.parentNode
```

```
// Gets a reference to the parent of the element's parent
element.parentNode.parentNode
```

Remove A Child Node

```
// Removes a child element
element.removeChild(elementToRemove);
```

Note: elementToRemove is a reference to the child element object to be removed. element would be a reference to elementToRemove's parent.

Building Complex Markup with Javascript

Sample Markup

```
<div id="test">
  <p class="fancy">
    <span>Hello</span><br />
    My name is Dan Daman.
  </p>
  <div class="clear"></div>
</div>
```

Sample Javascript

```
var paragraphSpan = document.createElement("span");
paragraphSpan.innerHTML = "Hello";

var paragraph = document.createElement("p");
paragraph.setAttribute("class", "fancy");
paragraph.appendChild(paragraphSpan);
paragraph.appendChild(document.createElement("br"));
paragraph.appendChild(document.createTextNode("My name is Dan Daman.));

var clearDiv = document.createElement("div");
clearDiv.setAttribute("class", "clear");

var testDiv = document.createElement("div");
testDiv.setAttribute("id", "test");
testDiv.appendChild(paragraph);
testDiv.appendChild(clearDiv);
```

TIP: Create the lowest DOM level elements first and work your way out, adding the child elements to it's parents as you go.