



The Basics

Select an HTML Element by ID

```
$(sId)
```

Get the Value of a Form Field by ID

```
$F(sId)
```

HTML Events

Attaching an Event Handler to an HTML Element

```
nitobi.html.attachEvent($( "id" ), "click", handleClick);
```

Attaching Multiple Event Handlers to One HTML Element

```
var evts = [{type:"mouseover",handler:handleMouseover},  
            {type:"mouseout",handler:handleMouseout}];  
nitobi.html.attachEvents($( "id" ), evts);
```

Detaching those same Events

```
nitobi.html.detachEvents($( "id" ), evts);
```

Detach events to
avoid memory
leaks

CSS

Add a CSS Class to an Object (by ID)

```
nitobi.html.Css.addClass($( "id" ), "myClass");
```

Remove a CSS Class from an Object (by ID)

```
nitobi.html.Css.removeClass($( "id" ), "myClass");
```

Precache All Images Referenced in a StyleSheet

```
nitobi.html.Css.precacheImages();
```

Get the Computed Style Object for an Element

```
nitobi.html.Css.getStyle($( "id" ), "background-color");
```

Get the Style Object for the CSS Selector

```
nitobi.html.Css.getStyleBySelector(".myClass");
```

Browser Checking

Check the Browser

```
nitobi.browser.IE  
nitobi.browser.IE6  
nitobi.browser.IE7  
nitobi.browser.MOZ
```

XML and XSLT

Create a Browser Native XML Document

```
nitobi.xml.createXmlDoc("<root></root>");
```

Create a Browser Native XSLT Processor

```
nitobi.xml.createXslProcessor("<xsl:stylesheet ... />");
```

Transform XML Using XSLT

```
var result = nitobi.xml.transform(xmlDoc, xslDoc);
```

Use XPath to Quickly Access XML Nodes

```
var xmlDoc = nitobi.xml.createXmlDoc("<root>...</root>");  
var custs = nitobi.xml.selectNodes("//Customer");  
var cust = nitobi.xml.selectSingleNode("//Customer[@name='Fred']");
```

This is fast!

Ajax

Create a Browser Native XMLHttpRequest Object

```
var xhr = nitobi.ajax.createXmlHttpRequest();
```

Use the More Friendly Nitobi HttpRequest Object

```
var hr = new nitobi.ajax.HttpRequest();
```

GET Data from the Server

```
var hr = new nitobi.ajax.HttpRequest();  
ht.handler = "myscript.php";  
ht.onGetComplete.subscribe(myFunction);  
ht.get();
```

Object Oriented Programming

Basic Inheritance

```
ns.Bird = function() {  
    this.flightless = false;  
};  
ns.Ostrich = function() {  
    ns.Ostrich.baseConstructor.call(this);  
    this.flightless = true;  
};  
nitobi.lang.extend(ns.Ostrich, ns.Bird);
```

Mixins are like
multiple inheritance

Use Interfaces to Make Mixins!

```
ns.Raven = function() {}  
ns.IFlyingBird = function() {}  
ns.IFlyingBird.prototype.fly = function() {  
    alert("I am flying!");  
};  
nitobi.lang.implement(ns.Raven, ns.IFlyingBird);
```

JavaScript Events

Create a JavaScript New Event

```
this.onReady = new nitobi.base.Event();
```

Subscribe a Handler to the Event

```
this.onReady.subscribe(this.handleReady, this);
```

Subscribe a Handler Only Once

```
this.onReady.subscribeOnce(this.handleReady,  
this);
```

Keep the execution scope
in subscribed functions!

Effects BETA

Instantiate a New Effect Using an HTML Element

```
var fx = new nitobi.effects.BlindUp($( "id" ));
```

Execute a Function When the Effect is Complete

```
fx.onFinish.subscribeOnce(effectComplete);
```

Run the Effect

```
fx.start();
```

Troubleshooting Resources

Nitobi Forums - <http://forums.nitobi.com>

Nitobi Knowledgebase - <http://www.nitobi.com/kb/>

Email Support - support@nitobi.com

Fiddler Tool - <http://www.fiddlertool.com>

Firebug - <http://www.getfirebug.com>