

Numeric Validation JavaScript

Series I : Validation (AJAX Presentation CWP)

Several people have asked me to discuss some of the details from the AJAX presentation from our last Campus Web Publishers (CWP) meeting.

In this article, which is part of the AJAX presentation series, I will share some validation functions I have used to validate the grid. In addition to validating the field, notice how it maps the arrows keys to give the user a Microsoft Excel like feel.

Validating a numeric field

```
function NumbersOnly(Object) {if(CurrentCellClicked) {if((event.shiftKey && event.keyCode ==
```

Validating a currency field

```
function CurrencyOnly(Object) {  
  if(CurrentCellClicked) {  
    if((event.shiftKey && event.keyCode == 9) || event.keyCode==37) {  
      event.returnValue=true;  
      return ;  
    }  
    else if(event.keyCode9 || event.keyCode39) {  
      event.returnValue=true;  
      return ;  
    }  
  }  
}
```

```
if(event.keyCode == 36 || event.keyCode == 35 || event.keyCode == 46 || event.keyCode == 8)  
event.returnValue=true;  
else if((event.shiftKey && event.keyCode == 9) || event.keyCode37) { var t =  
parseInt(Object.nextField); if(t > 2) { t = GlobalForm['c'+(t-2)]; if(t.disabled) {  
GlobalForm['c'+(t.nextField-2)].select(); GlobalForm['c'+(t.nextField-2)].focus(); } else { t.select();  
t.focus(); } event.returnValue = false; } } else if(event.shiftKey) event.returnValue = false; else  
if(event.keyCode9 || event.keyCode==39) {  
var t = GlobalForm['c'+Object.nextField];  
if(t.disabled) {  
GlobalForm['c'+t.nextField].select();
```

```

GlobalForm['c'+t.nextField].focus();
}
else {
t.select();
t.focus();
}
event.returnValue = false;
}
else {
event.returnValue = (event.keyCode == 110 || event.keyCode == 190 || event.keyCode == 109 ||
event.keyCode == 189 || (event.keyCode != 47 && ((event.keyCode>=45 &&
event.keyCode<=57) || (event.keyCode>=96 && event.keyCode<=105))));
}
}

```

Validating a Date Field

```

function DateOnly(Object) {
  if(CurrentCellClicked) {
    if((event.shiftKey && event.keyCode == 9) || event.keyCode==37) {
      event.returnValue=true;
      return ;
    }
    else if(event.keyCode9 || event.keyCode39) {
      event.returnValue=true;
      return ;
    }
  }
}

```

```

if(event.keyCode == 36 || event.keyCode == 35 || event.keyCode == 46 || event.keyCode == 8)
event.returnValue=true;
else if((event.shiftKey && event.keyCode == 9) || event.keyCode37) { var t =
parseInt(Object.nextField); if(t > 2) { t = GlobalForm['c'+(t-2)]; if(t.disabled) {
GlobalForm['c'+(t.nextField-2)].select(); GlobalForm['c'+(t.nextField-2)].focus(); } else { t.select();
t.focus(); } event.returnValue = false; } } else if(event.keyCode9 || event.keyCode==39) {
var t = GlobalForm['c'+Object.nextField];
if(t.disabled) {
GlobalForm['c'+t.nextField].select();
GlobalForm['c'+t.nextField].focus();
}
else {
t.select();
t.focus();
}
event.returnValue = false;
}

```

```

else {
event.returnValue = (event.keyCode == 109 || event.keyCode == 189 || event.keyCode == 191 ||
event.keyCode == 111 || (event.keyCode >= 47 && event.keyCode <= 57) || (event.keyCode >= 96
&& event.keyCode <= 105));
}
}

```

Of course, date validation also requires:

```

function IsDate(DateObject) {
    var month, year, day;
    if(DateObject.value.length < 6) {
        var datePat = /^(\d{1,2})(\v|-)(\d{1,2})$/;
        var matchArray = DateObject.value.match(datePat); // is the format ok?
        if (matchArray == null)
            return false;
    }
}

```

```

month = matchArray1; // parse date into variables
day = matchArray3;
year = (new Date()).getFullYear();
}
else {
var datePat = /^(\d{1,2})(\v|-)(\d{1,2})(\v|-)(\d{2,4})$/;
var matchArray = DateObject.value.match(datePat); // is the format ok?
if (matchArray == null)
return false;
}

```

```

month = matchArray1; // parse date into variables
day = matchArray3;
year = matchArray5;
}

```

```

if (month < 1 || month > 12) { // check month range
throw (new Exception("Month must be between 1 and 12."));
}
if (day < 1 || day > 31) {
throw (new Exception("Day must be between 1 and 31."));
}
if ((month4 || month6 || month9 || month11) && day==31) {
throw (new Exception("Month " + month + " doesn't have 31 days"));
}
if (month == 2) { // check for february 29th
var isleap = (year % 4 == 0 && (year % 100 != 0 || year % 400 == 0));
if (day > 29 || (day==29 && !isleap)) {
throw (new Exception("February " + year + " doesn't have " + day + " days!"));
}
}
}

```

```
if(year.toString().length>2) {  
year = year = year.toString().substring(2);  
}  
DateObject.value = (parseInt(month)) + '/' + (parseInt(day)) + '/' + year;  
return true; // date is valid  
}
```

Revision #2

Created 2006-05-26 00:33:08 UTC by Quazi, Salman

Updated 2006-05-26 00:39:14 UTC by Quazi, Salman