




# 가

Last update: 2023/05/11

가	1
.....	1
.....	1

# 가

: /HTML5/UTIL/CALENDAR/calendar\_tableview

-  [calendar\\_tableview.xml](#)
-  [calendar\\_tableview.js](#)
- 

```
var CALENDAR_UTIL = {
  objSelectDay: null,           //
  nSelectDay: "01",           //

  arrSaturdayForeColor: [0, 142, 185], //
  arrHolidayDayForeColor: [206, 0, 0], //
  arrWorkDayForeColor: [120, 120, 120], //

  arrInvalidDayForeColor: [195, 195, 195], //
  arrSelectDayForeColor: [255, 255, 255], //
  arrSelectDayBackColor: [96, 154, 185], //
  bShowOnlyThisMonth: false, //
  strFontName: " ",

//
  setDayStyle: function(objBtnDay, strYYYYMMDD, nDayOfWeek) {
    var objUserData;

    if (nDayOfWeek === undefined) {
      objUserData = objBtnDay.getuserdata();
      nDayOfWeek = objUserData.nDayOfWeek;
    }

    switch (nDayOfWeek) {
      case 0: this.setHolidayDayStyle(objBtnDay); break;
      case 6: this.setSaturdayStyle(objBtnDay); break;
      default: this.setWorkDayStyle(objBtnDay); break;
    }
  }
}
```

```
        if (strYYYYMMDD !== undefined) {
            objBtnDay.setuserdata({ strMonthType: "THIS_MONTH", strYYYYMMDD:
strYYYYMMDD, nDayOfWeek: nDayOfWeek });
        }
    },

    //
    setSaturdayStyle: function(objBtnDay) {
        objBtnDay.setforecolor(this.arrSaturdayForecolor[0],
this.arrSaturdayForecolor[1], this.arrSaturdayForecolor[2]);
        objBtnDay.setbackcolor(255,255,255);
        objBtnDay.setfont(this.strFontName, 8, false, false, false);
        objBtnDay.settooltiptext("");
        objBtnDay.setvisible(true);
        objBtnDay.setenable(true);
    },

    //
    /
    setHolidayDayStyle: function(objBtnDay, strTooltip) {
        objBtnDay.setforecolor(this.arrHolidayDayForecolor[0],
this.arrHolidayDayForecolor[1], this.arrHolidayDayForecolor[2]);
        objBtnDay.setbackcolor(255,255,255);
        objBtnDay.setfont(this.strFontName, 8, false, false, false);
        if (strTooltip === undefined) { objBtnDay.settooltiptext(""); }
        else { objBtnDay.settooltiptext(strTooltip); }
        objBtnDay.setvisible(true);
        objBtnDay.setenable(true);
    },

    //
    setWorkDayStyle: function(objBtnDay) {
        objBtnDay.setforecolor(this.arrWorkDayForecolor[0],
this.arrWorkDayForecolor[1], this.arrWorkDayForecolor[2]);
        objBtnDay.setbackcolor(255,255,255);
        objBtnDay.setfont(this.strFontName, 8, false, false, false);
        objBtnDay.settooltiptext("");
        objBtnDay.setvisible(true);
        objBtnDay.setenable(true);
    },

    //
    /
    setInvalidDayStyle: function(objBtnDay, strYYYYMMDD, strMonthType) {
        objBtnDay.setforecolor(this.arrInvalidDayForecolor[0],
this.arrInvalidDayForecolor[1], this.arrInvalidDayForecolor[2]);
        objBtnDay.setbackcolor(255,255,255);
        objBtnDay.setfont(this.strFontName, 8, false, false, false);
        objBtnDay.settooltiptext("");
        objBtnDay.setvisible(true);
        objBtnDay.setenable(true);

        objBtnDay.setuserdata({ strMonthType: strMonthType, strYYYYMMDD:
strYYYYMMDD });
    }
};
```

```

    },

    //
    setSelectDayStyle: function(objBtnDay, strYYYYMMDD, nDayOfWeek) {
        //
        if (this.objSelectDay) { this.setDayStyle(this.objSelectDay); }

        objBtnDay.setforecolor(this.arrSelectDayForeColor[0],
this.arrSelectDayForeColor[1], this.arrSelectDayForeColor[2]);
        objBtnDay.setbackcolor(this.arrSelectDayBackColor[0],
this.arrSelectDayBackColor[1], this.arrSelectDayBackColor[2]);
        objBtnDay.setfont(this.strFontName, 8, true, false, true);
        objBtnDay.settooltiptext("    ");
        objBtnDay.setvisible(true);
        objBtnDay.setenable(true);

        if (strYYYYMMDD !== undefined) {
            objBtnDay.setuserdata({ strMonthType: "THIS_MONTH", strYYYYMMDD:
strYYYYMMDD, nDayOfWeek: nDayOfWeek });
        }

        this.objSelectDay = objBtnDay;
    }
};

//
function showCalendarButton(nYear, nMonth, nDay)
{
    var i, j, strBtnIndex, objBtnDay, nTempDay, nDayOfWeekTemp;
    var strCurrYYYYMMDD, strCurrYYYYMM, strPrevYYYYMMDD, strPrevYYYYMM,
strNextYYYYMMDD, strNextYYYYMM;

    nYear = parseInt(nYear, 10);
    nMonth = parseInt(nMonth, 10);

    //
    this.fldYear.settext(nYear.toString(10));
    this.fldMonth.settext(nMonth.toString(10));

    strCurrYYYYMMDD = strSelectDate =
factory.fillstring(this.fldYear.gettext(), "0", 4, true) +
    factory.fillstring(this.fldMonth.gettext(), "0", 2, true) + "01";
    strCurrYYYYMM = strCurrYYYYMMDD.substring(0, 6);

    //
    // (28/29/30/31)
    nCurrLastDay =
parseInt(factory.datelastdayofmonth(strCurrYYYYMMDD).substring(6), 10);

    //
    // /
    strPrevYYYYMMDD = factory.datefrom("month", -1, strCurrYYYYMMDD);
    strPrevYYYYMM = strPrevYYYYMMDD.substring(0, 6);
    strNextYYYYMMDD = factory.datefrom("month", 1, strCurrYYYYMMDD)

```

```

strNextYYYYMM = strNextYYYYMMDD.substring(0, 6);

//          (28/29/30/31)
nPrevLastDay =
parseInt(factory.datelastdayofmonth(strPrevYYYYMMDD).substring(6), 10);

nDayOfWeek = new Date(nYear, nMonth - 1, 1).getDay();

//
for (i = 0; i < nDayOfWeek; i++) {
//
strBtnIndex = factory.fillstring(i, "0", 2, true);
objBtnDay = screen.getinstancebyname("btnDay" + strBtnIndex);

nTempDay = (nPrevLastDay - nDayOfWeek) + (i + 1);

if (CALENDAR_UTIL.bShowOnlyThisMonth) { objBtnDay.settext(""); }
else { objBtnDay.settext(nTempDay); }

//
CALENDAR_UTIL.setInvalidDayStyle(objBtnDay, strPrevYYYYMM +
factory.fillstring(nTempDay, "0", 2, true), "PREV_MONTH");
}

nTempDay = 1;
for (i = nDayOfWeek; i < 42; i++) {
//
strBtnIndex = factory.fillstring(i, "0", 2, true);
objBtnDay = screen.getinstancebyname("btnDay" + strBtnIndex);

//
if (nTempDay <= nCurrLastDay) {
nDayOfWeekTemp = i % 7;
objBtnDay.settext(nTempDay);

//          가
if (nDay == nTempDay) {
CALENDAR_UTIL.setSelectDayStyle(objBtnDay, strCurrYYYYMM +
+ factory.fillstring(nTempDay, "0", 2, true), nDayOfWeekTemp);
}
else {
CALENDAR_UTIL.setDayStyle(objBtnDay, strCurrYYYYMM + +
factory.fillstring(nTempDay, "0", 2, true), nDayOfWeekTemp);
}
}
//
else {
if (CALENDAR_UTIL.bShowOnlyThisMonth) { objBtnDay.settext(""); }
else { objBtnDay.settext(nTempDay - nCurrLastDay); }

CALENDAR_UTIL.setInvalidDayStyle(objBtnDay, strNextYYYYMM +
factory.fillstring((nTempDay - nCurrLastDay), "0", 2, true), "NEXT_MONTH");
}
}

```

```
    }

    nTempDay++;
}

return;
}

function btnShowCalendar_on_mouseup(objInst)
{
    this.updateCalendar();
}

function updateCalendar()
{
    var nYear, nMonth;

    //
    nYear = parseInt(this.fldYear.gettext(), 10);
    nMonth = parseInt(this.fldMonth.gettext(), 10);

    this.showCalendarButton(nYear, nMonth, CALENDAR_UTIL.nSelectDay);
}

//
function handleYearMonthPrevNextButton(isYear, isNext)
{
    var nYear, nMonth;

    //
    nYear = parseInt(this.fldYear.gettext(), 10);
    nMonth = parseInt(this.fldMonth.gettext(), 10);

    if (isYear) {
        if (isNext) { nYear = nYear + 1; }
        else { nYear = nYear - 1; }
    }
    else {
        if (isNext) {
            nMonth = nMonth + 1;
            if (nMonth > 12) {
                nMonth = 1;
                nYear += 1;
            }
        }
        else {
            nMonth = nMonth - 1;
            if (nMonth < 1) {
                nMonth = 12;
                nYear -= 1;
            }
        }
    }
}
```

```
    }  
  }  
  
  this.fldYear.setText(nYear);  
  this.fldMonth.setText(nMonth);  
  
  this.showCalendarButton(nYear, nMonth, CALENDAR_UTIL.nSelectDay);  
}  
  
/**  
 * Event  
 */  
function screen_on_load()  
{  
  var i, strCurrYYYYMMDD, nCurrentYear, nCurrentMonth, nCurrentDay;  
  
  strCurrYYYYMMDD = factory.datetoday();  
  
  //  
  nCurrentYear = parseInt(strCurrYYYYMMDD.substring(0, 4), 10);  
  nCurrentMonth = parseInt(strCurrYYYYMMDD.substring(4, 6), 10);  
  nCurrentDay = parseInt(strCurrYYYYMMDD.substring(6), 10);  
  
  //  
  this.showCalendarButton(nCurrentYear, nCurrentMonth, nCurrentDay);  
}  
  
//  
function btnMonthNext_on_mouseup()  
{  
  this.handleYearMonthPrevNextButton(false, true);  
}  
  
//  
function btnMonthBefore_on_mouseup()  
{  
  this.handleYearMonthPrevNextButton(false, false);  
}  
  
//  
function btnYearNext_on_mouseup()  
{  
  this.handleYearMonthPrevNextButton(true, true);  
}  
  
//  
function btnYearBefore_on_mouseup()  
{  
  this.handleYearMonthPrevNextButton(true, false);  
}  
  
//
```



```
function btnDay_on_mouseup(objInst)
{
    var strSelectDate, strDay, objUserData;

    //
    strDay = objInst.gettext();
    if (strDay == "") { return; }

    objUserData = objInst.getuserdata();

    CALENDAR_UTIL.nSelectDay = strDay;

    //
    if (objUserData) {
        if (objUserData.strMonthType == "NEXT_MONTH") {
            this.btnMonthNext_on_mouseup();
            return;
        }
        else if (objUserData.strMonthType == "PREV_MONTH") {
            this.btnMonthBefore_on_mouseup();
            return;
        }
    }

    factory.consoleprint("strSelectDate = [" + objUserData.strYYYYMMDD +
    "]);

    CALENDAR_UTIL.setSelectDayStyle(objInst);
}
```

From:

<https://technet.softbase.co.kr/wiki/> - **xFrame5 TechNet**

Permanent link:

[https://technet.softbase.co.kr/wiki/guide/util/calendar/calendar\\_tableview](https://technet.softbase.co.kr/wiki/guide/util/calendar/calendar_tableview)



Last update: **2023/05/11 16:21**