커스텀 일자 달력 가이드

Last update: 2023/05/11

www.softbase.co.kr

목차

커스텀 일자 달력 가이드

이 화면은 사용자 정의 달력을 개발하기 위한 샘플화면이다.

예시

템플릿 위치: /HTML5/UTIL/CALENDAR/calendar_button

템플릿 파일

- calendar_button.xml
- calendar_button.js
- 새창으로 실행

화면 스크립트

```
var CALENDAR_UTIL = {
    objSelectDay: null,                             // 현재 선택된 날짜 버튼
    nSelectDay: "01",                               // 현재 선택된 일

    arrSaturDayForeColor: [0, 142, 185],            // 토요일 전경색
    arrHolidayDayForeColor: [206, 0, 0],            // 일요일, 공유일의 전경색
    arrWorkDayForeColor: [120, 120, 120],           // 토요일, 일요일, 공휴일을 제외한
날짜의 전경색
    arrInvalidDayForeColor: [195, 195, 195],        // 현재달이 아닌 날짜의 전경색
    arrSelectDayForeColor: [255, 255, 255],         // 선택일 전경색
    arrSelectDayBackColor: [96, 154, 185],          // 선택일 배경색
    bShowOnlyThisMonth: false,                      // 선택월 날짜만 보여줄지 여부
    strFontName: "맑은 고딕",

  // 요일에 따라 스타일 적용
    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 });
        }
    },

  // 토요일 스타일 적용
    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);
    },

  // 일요일/휴일 스타일 적용
    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);
    },

  // 주중 스타일 적용
    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);
    },

  // 이전달/다음달 스타일 적용
    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 });
```

```
    },

  // 선택한 날짜 스타일 적용
    setSelectDayStyle: function(objBtnDay, strYYYYMMDD, nDayOfWeek) {
     // 현재 선택되어 있는 일자에 대한 칼라를 원상태로 복귀
        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("선택일");
        objBtnDay.setvisible(true);
        objBtnDay.setenable(true);

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

        this.objSelectDay = objBtnDay;
    }
};

// 달력 날짜 표시
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);

  // 년도 콤보 및 월 콤보에 값 설정
    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);

  // 달력 표시 년월의 마지막 일을 구함(28/29/30/31)
    nCurrLastDay =
parseInt(factory.datelastdayofmonth(strCurrYYYYMMDD).substring(6), 10);

  // 달력 표시 년월의 한달전/한달후 날짜 오브젝트를 구함
    strPrevYYYYMMDD = factory.datefrom("month", -1, strCurrYYYYMMDD);
    strPrevYYYYMM = strPrevYYYYMMDD.substring(0, 6);
    strNextYYYYMMDD = factory.datefrom("month", 1, strCurrYYYYMMDD)
```

```
    strNextYYYYMM = strNextYYYYMMDD.substring(0, 6);

  // 달력 표시 년월의 이전달 마지막 일을 구함(28/29/30/31)
    nPrevLastDay =
parseInt(factory.datelastdayofmonth(strPrevYYYYMMDD).substring(6), 10);

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

  // 이전월 날짜 표시
    for (i = 0; i < nDayOfWeek; i++) {
    // 버튼 인덱스를 기준으로 버튼 오브젝트를 구함
        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); }

    // 이전월 또는 다음월의 날짜 버튼에 대한
        CALENDAR_UTIL.setInvalidDayStyle(objBtnDay, strPrevYYYYMM +
factory.fillstring(nTempDay, "0", 2, true), "PREV_MONTH");
    }

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

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

       // 버튼 텍스트가 현재 날짜와 동일한 경우, 폰트 설정
            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);
            }
        }
    // 다음달 날짜 표시 정보 생성
        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;

  // 변경된 콤보 박스의 년도 및 월을 구함
    nYear = parseInt(this.fldYear.gettext(), 10);
    nMonth = parseInt(this.fldMonth.gettext(), 10);

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

// 년도 및 월 이전, 이후 버튼 클릭 이벤트 처리
function handleYearMonthPrevNextButton(isYear, isNext)
{
    var nYear, nMonth;

  // 변경된 콤보 박스의 년도 및 월을 구함
    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 처리 함수 시작
 */
function screen_on_load()
{
    var i, strCurrYYYYMMDD, nCurrentYear, nCurrentMonth, nCurrentDay;

    strCurrYYYYMMDD = factory.datetoday();

  // 변경된 콤보 박스의 년도 및 월을 구함
    nCurrentYear = parseInt(strCurrYYYYMMDD.substring(0, 4), 10);
    nCurrentMonth = parseInt(strCurrYYYYMMDD.substring(4, 6), 10);
    nCurrentDay = parseInt(strCurrYYYYMMDD.substring(6), 10);

  // 달력 표시
    this.showCalendarButton(nCurrentYear, nCurrentMonth, nCurrentDay);
}

// 다음달 버튼 클릭
function btnMonthNext_on_mouseup()
{
    this.handleYearMonthPrevNextButton(false, true);
}

// 이전달 버튼 클릭
function btnMonthBefore_on_mouseup()
{
    this.handleYearMonthPrevNextButton(false, false);
}

// 다음년도 버튼 클릭
function btnYearNext_on_mouseup()
{
    this.handleYearMonthPrevNextButton(true, true);
}

// 이전년도 버튼 클릭
function btnYearBefore_on_mouseup()
{
    this.handleYearMonthPrevNextButton(true, false);
}

// 날짜 버튼 클릭 이벤트
```

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

  // 클릭된 버튼의 일자 및 정보를 구함
    strDay = objInst.gettext();
    if (strDay == "") { return; }

    objUserData = objInst.getuserdata();

    CALENDAR_UTIL.nSelectDay = strDay;

  // 이전달이거나, 다음달인 경우
    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);
}
```