

动态日历

1.外在架构

```
(function (global) {  
  }(typeof window !== "undefined" ? window : this));
```

姑且不论()中何意思，但typeof window是何意思？进行打印

```
console.log(typeof window );
```

object

[dycalendar.js:554](#)

全局变量：

```
// 指定代码在严格条件下执行。 严格模式下你不能使用未声明的变量  
"use strict";  
  
var  
  //this will be used by the user.  
  dycalendar = {},  
  
  // global是window吗？  
  //window document  
  document = global.document,  
  
  //starting year  
  START_YEAR = 1900,  
  
  //end year  
  END_YEAR = 9999,  
  
  //name of the months  
  monthName = {  
    full: ['January', 'February', 'March', 'April', 'May', 'June',  
    'July', 'August', 'September', 'October', 'November', 'December'],  
    mmm: ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep',  
    'Oct', 'Nov', 'Dec']  
  },  
  
  //name of the days  
  dayName = {  
    full: ['Sunday', 'Monday', 'Tuesday', 'Wednesday', 'Thursday',  
    'Friday', 'Saturday'],  
    d: ['S', 'M', 'T', 'W', 'T', 'F', 'S'],  
    dd: ['Su', 'Mo', 'Tu', 'We', 'Th', 'Fr', 'Sa'],  
    ddd: ['Sun', 'Mon', 'Tue', 'Wed', 'Thu', 'Fri', 'Sat']  
  };  
};
```

- console.log(global);

```
▼ Window {window: Window, self: Window, document: document, name: "", location: Location, ...} dycalendar.js:18
  ▶ alert: f alert()
  ▶ atob: f atob()
  ▶ blur: f blur()
  ▶ btoa: f btoa()
  ▶ caches: CacheStorage {}
  ▶ cancelAnimationFrame: f cancelAnimationFrame()
  ▶ cancelIdleCallback: f cancelIdleCallback()
  ▶ captureEvents: f captureEvents()
  ▶ chrome: f loadTimes: f ...
```

激活 Windows
转到“设置”以激活 Windows。

其内容是非常多的函数

```
//attach to global window object
global.dycalendar = dycalendar;
```

如果没有这一句，会显示

```
✖ ▶ Uncaught TypeError: dycalendar.draw is not a function (index):21
at (index):21
```

也就是说dycalendar.draw无法正常运行

```
//attach to global window object
console.log(dycalendar);
console.log( global.dycalendar);
global.dycalendar = dycalendar;
console.log( global.dycalendar);

▶ {draw: f} dycalendar.js:619
▶ <div id="dycalendar">...</div> dycalendar.js:620
▶ {draw: f} dycalendar.js:622
```

global.dycalendar由一个元素变成一个函数，什么意思呢？

2.我们在html文件中可以看到此js文件是这样被引用的

```
<script src="dycalendar.js"></script>
<script>
  dycalendar.draw({
    target: '#dycalendar',
    type: 'month',
    highlighttargetdate: true,
    prevnextbutton: 'show'
  })
</script>
```

我们在js文件中去寻找此函数

```
/**
 * this function will draw the calendar based on user preferences.
 *
 * option = {
 *   target : "#id|.class"  //(mandatory) for id use #id | for class use
 *   .class
```

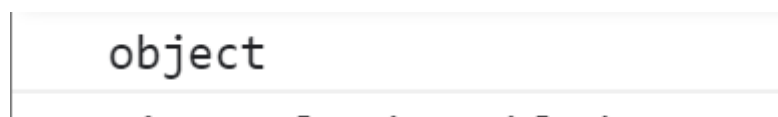
```

    * type : "calendar-type" // (optional) values: "day|month" (default "day")
    * month : "integer"      // (optional) value 0-11, where 0 = January, ...
    11 = December (default current month)
    * year : "integer"       // (optional) example 1990. (default current
    year)
    * date : "integer"       // (optional) example 1-31. (default current
    date)
    * monthformat : "full"   // (optional) values: "mmm|full" (default "full")
    * dayformat : "full"     // (optional) values: "ddd|full" (default "full")
    * highlighttoday : boolean // (optional) (default false) if true will
    highlight today's date
    * highlighttargetdate : boolean // (optional) (default false) if true will
    highlight targeted date of the month year
    * prevnextbutton : "hide" // (optional) (default "hide") (values:
    "show|hide") if set to "show" it will show the nav button (prev|next)
    * }
    *
    * @param object option    user preferences
    * @return boolean         true if success, false otherwise
    */
    dycalendar.draw = function (option) {

        //check if option is passed or not
        if (typeof option === "undefined") {
            global.console.error("Option missing");
            return false;
        }
        var
            self = this, //pointing at dycalendar object
            dateObj = new Date(), //current date
            //default settings
            defaults = {
                type: "day",
                month: dateObj.getMonth(),
                year: dateObj.getFullYear(),
                date: dateObj.getDate(),
                monthformat: "full",
                dayformat: "full",
                highlighttoday: false,
                highlighttargetdate: false,
                prevnextbutton: "hide"
            };
        //extend user options with predefined options
        option = extendSource(option, defaults);
        drawCalendar(option);
    };

```

- 可以看到在此函数的参数是一个数组，但在js文件中的变量名是看不出变量类型
- typeof option是何意呢？



- 可以看到会有默认的设置与个性化设置进行合并得到最终的设置

3.extendSource(option, defaults)

```
/**
 * this function will extend source object with defaults object.
 *
 * @param object source    this is the source object
 * @param object defaults  this is the default object
 * @return object
 */
function extendSource(source, defaults) {
    var property;
    for (property in defaults) {
        /*如果个性化设置里没有设置某一属性，那么此属性的设置值就采用默认设置*/
        if (source.hasOwnProperty(property) === false) {
            source[property] = defaults[property];
        }
    }
    return source;
}
```

- 这里有一个想法，参数不是数组，而是哈希表

4.drawCalendar(option);

```
/*为方便观察函数，我们不妨把option中的内容打印出来
{
    "target": "#dycalendar",
    "type": "month",
    "highlighttargetdate": true,
    "prevnextbutton": "show",
    "month": 0,
    "year": 2023,
    "date": 19,
    "monthformat": "full",
    "dayformat": "full",
    "highlighttoday": false
}*/
//option中的日期有两种情况，一种是当前日期，另一种是由点击事件而导致的其他日期
function drawCalendar(option) {

    var
        //variables for creating calendar
        calendar,
        calendarHTML,
        targetedElementBy = "id",
        targetElem,

        //other variables
        i, len, elemArr;

    //find target element by
    if (option.target[0] === "#") {
        targetedElementBy = "id";
    } else if (option.target[0] === ".") {
```

```

        targetedElementBy = "class";
    }
    targetElem = option.target.substring(1);

    //get calendar HTML
    switch (option.type) {
        case "day":
            //get calendar detail
            calendar = getCalendar(option.year, option.month, option.date);
            //get calendar html
            calendarHTML = drawCalendarDay(calendar, option);
            break;

        case "month":
            //get calendar detail
            calendar = getCalendar(option.year, option.month, option.date);
            //get calendar html
            calendarHTML = drawCalendarMonthTable(calendar, option);
            break;

        default:
            global.console.error("Invalid type");
            return false;
    }

    //draw calendar
    if (targetedElementBy === "id") {

        document.getElementById(targetElem).innerHTML =
calendarHTML.outerHTML;

    } else if (targetedElementBy === "class") {


        elemArr = document.getElementsByClassName(targetElem);
        for (i = 0, len = elemArr.length; i < len; i = i + 1) {
            elemArr[i].innerHTML = calendarHTML.outerHTML;
        }

    }

}

```

- 打印option:

▼ Object 

```
date: 19
dayformat: "full"
highlighttargetdate: true
highlighttoday: false
month: 0
monthformat: "full"
prevnextbutton: "show"
target: "#dycalendar"
type: "month"
year: 2023
▶ [[Prototype]]: Object
```

发现最后有Prototype(原型)

```
▶ constructor: f Object()
▶ hasOwnProperty: f hasOwnProperty()
▶ isPrototypeOf: f isPrototypeOf()
▶ propertyIsEnumerable: f propertyIsEnumerable()
▶ toLocaleString: f toLocaleString()
▶ toString: f toString()
▶ valueOf: f valueOf()
▶ __defineGetter__: f __defineGetter__()
▶ __defineSetter__: f __defineSetter__()
▶ __lookupGetter__: f __lookupGetter__()
▶ __lookupSetter__: f __lookupSetter__()
▶ get __proto__: f __proto__()
▶ set __proto__: f __proto__()
```

这是一系列函数

- calendarHTML本身是一个class，如果它要复制给有id的元素的话，那么直接复制即可；但若是复制给只有class属性的元素的话，要以数组形式逐一复制

5.getCalendar(year, month, date)

```
/**
 * This function will return calendar detail.
 *
 * @param integer year          1900-9999 (optional) if not set will consider
```

```

    * the current year.
    * @param integer month 0-11 (optional) 0 = Jan, 1 = Feb, ... 11 =
Dec,
    * if not set will consider the current month.
    * @param integer date 1-31 (optional)
    * @return boolean|object if error return false, else calendar detail
    */
    // result.today.dayIndex
    // result.today.dayName
    // result.today.dayFullName
    // result.today.monthIndex
    // result.today.monthName
    // result.today.monthNameFull
    // result.today.date
    // result.today.year

    // result.year
    // result.month
    // result.date
    // result.firstDayIndex first day of the month
    // result.firstDayName
    // result.firstDayFullName
    // result.monthIndex
    // result.monthName
    // result.monthNameFull
    // result.totaldays
    // result.targetedDayIndex
    // result.targetedDayName
    // result.targetedDayFullName

function getCalendar(year, month, date) {

    // example: year:2023 month:1 date:17
    var
        dateObj = new Date(), //Tue Jan 17 2023 18:42:57 GMT+0800 (中国标准时间)
        dateString,
        result = {},
        idx;

    if (year < START_YEAR || year > END_YEAR) {
        global.console.error("Invalid Year");
        return false;
    }
    if (month > 11 || month < 0) {
        global.console.error("Invalid Month");
        return false;
    }
    if (date > 31 || date < 1) {
        global.console.error("Invalid Date");
        return false;
    }

    result.year = year;
    result.month = month;
    result.date = date;

```

```

//today
result.today = {};
dateString = dateObj.toString().split(" ");//[ "Tue", "Jan", "17",
"2023", "18:42:57", "GMT+0800", "(中国标准时间)"]
// console.log(dateObj);

//the index of ddd formate of the current week date
idx = dayName.ddd.indexOf(dateString[0]); //idx :2
result.today.dayIndex = idx; //2
result.today.dayName = dateString[0]; //"Tue"
result.today.dayFullName = dayName.full[idx]; //"Tuesday"

idx = monthName.mmm.indexOf(dateString[1]); //0
result.today.monthIndex = idx; //0
result.today.monthName = dateString[1]; //'Jan'
result.today.monthNameFull = monthName.full[idx]; //'January'

result.today.date = dateObj.getDate(); //17

result.today.year = dateString[3]; //'2023'

//get month-year first day
dateObj.setDate(1);
dateObj.setMonth(month);
dateObj.setFullYear(year);
dateString = dateObj.toString().split(" ");
// console.log(dateString); // ["Sun", "Jan", "01", "2023", "19:21:49",
"GMT+0800", "(中国标准时间)"]

idx = dayName.ddd.indexOf(dateString[0]); //0
result.firstDayIndex = idx; //0
result.firstDayName = dateString[0]; //'Sun'
result.firstDayFullName = dayName.full[idx]; //'Sunday'

idx = monthName.mmm.indexOf(dateString[1]); //0
result.monthIndex = idx; //0
result.monthName = dateString[1]; //'Jan'
result.monthNameFull = monthName.full[idx]; //'January'

//get total days for the month-year
dateObj.setFullYear(year);
dateObj.setMonth(month + 1);
dateObj.setDate(0);
//month 2, date 0 = totaldays of month 1
result.totaldays = dateObj.getDate(); //31

//get month-year targeted date
dateObj.setFullYear(year);
dateObj.setMonth(month);
dateObj.setDate(date);
dateString = dateObj.toString().split(" ");//[ "Tue", "Jan", "17",
"2023", "18:42:57", "GMT+0800", "(中国标准时间)"]

idx = dayName.ddd.indexOf(dateString[0]); //2

```



```

    result.targetedDayIndex = idx;//2
    result.targetedDayName = dateString[0];//'Tue'
    result.targetedDayFullName = dayName.full[idx];//'Tuesday'

    return result;
}

```

- console.log(dateObj);

Fri Jan 20 2023 08:29:20 GMT+0800 (中国标准时间)	dycalendar.js:359
--	-------------------

- console.log(dateObj.toString());

console.log(dateObj.toString().split(" "));

Fri Jan 20 2023	dycalendar.js:379
▶ (7) ["Fri", "Jan", "20", "2023", "08:31:19", "GMT+0800", "(中国标准时间)"]	dycalendar.js:380

- ```

console.log(dateObj[0]);
console.log(dateObj.toString()[0]);
console.log(dateObj.toString().split(" ")[0]);
console.log(dayName.ddd.indexOf(dateObj[0]));
console.log(dayName.ddd.indexOf(dateObj.toString()[0]));
console.log(dayName.ddd.indexOf(dateObj.toString().split(" ")[0]));

```

|           |                   |
|-----------|-------------------|
| undefined | dycalendar.js:379 |
| F         | dycalendar.js:380 |
| Fri       | dycalendar.js:381 |
| -1        | dycalendar.js:382 |
| -1        | dycalendar.js:383 |
| 5         | dycalendar.js:384 |

由上易知为何要使用toString().split(" ");

- 如何求这个月有多少天：下一个月的0号就是这个月的总天数

## 6. function drawCalendarDay(data, option)

```

function drawCalendarDay(data, option) {

 var
 div, container, elem;

 //calendar container
 container = document.createElement("div");
 container.setAttribute("class", "dycalendar-day-container");

 //----- Header -----

 //header div
 div = document.createElement("div");
 div.setAttribute("class", "dycalendar-header");

 //day span
 elem = document.createElement("span");
 elem.setAttribute("class", "dycalendar-span-day");
 if (option.dayformat === "ddd") {
 elem.innerHTML = dayName.ddd[data.targetedDayIndex];
 }
}

```

```

 } else if (option.dayformat === "full") {
 elem.innerHTML = dayName.full[data.targetedDayIndex];
 }

 //add day span to footer div
 div.appendChild(elem);

 //add header div to container
 container.appendChild(div);

 //----- Body -----

 //body div
 div = document.createElement("div");
 div.setAttribute("class", "dycalendar-body");

 //date span
 elem = document.createElement("span");
 elem.setAttribute("class", "dycalendar-span-date");
 elem.innerHTML = data.date;

 //add date span to body div
 div.appendChild(elem);

 //add body div to container
 container.appendChild(div);

 //----- Footer -----

 //footer div
 div = document.createElement("div");
 div.setAttribute("class", "dycalendar-footer");

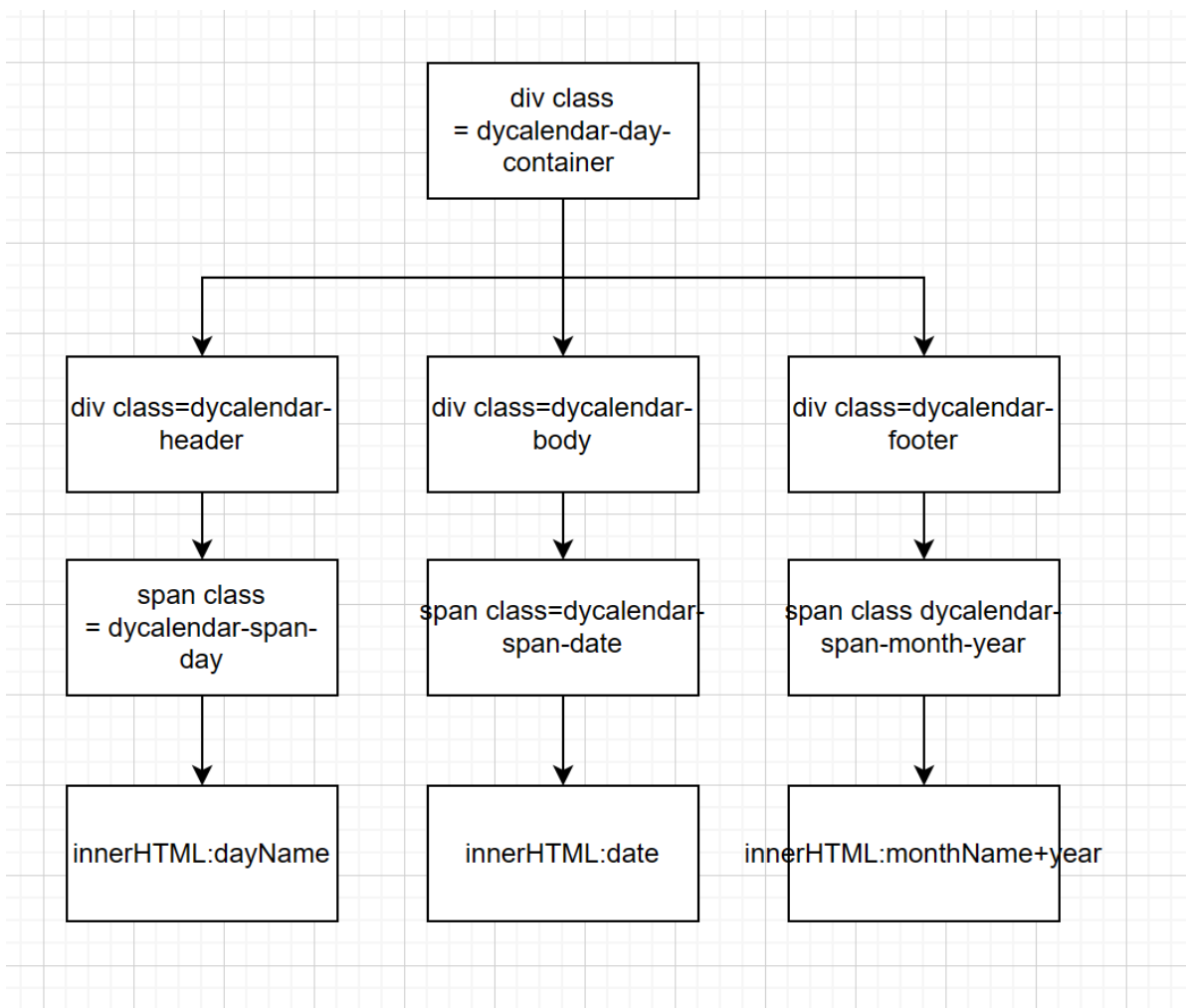
 //month span
 elem = document.createElement("span");
 elem.setAttribute("class", "dycalendar-span-month-year");
 if (option.monthformat === "mmm") {
 elem.innerHTML = data.monthName + " " + data.year;
 } else if (option.monthformat === "full") {
 elem.innerHTML = data.monthNameFull + " " + data.year;
 }

 //add month span to footer div
 div.appendChild(elem);

 //add footer div to container
 container.appendChild(div);

 //return container
 return container;
}

```



```
/*非常简易的格式*/
.dycalendar-day-container {
 padding : 10px;
 text-align : center;
 font-family : Arial;
}

.dycalendar-day-container div{
 padding : 0;
 margin-bottom : 10px;
}

.dycalendar-day-container .dycalendar-span-day {
 font-size : 110%;
}

.dycalendar-day-container .dycalendar-span-date {
 font-size : 250%;
}

.dycalendar-day-container .dycalendar-span-month-year {
 font-size : 90%
}
```

## 7.function drawCalendarMonthTable(data, option)

```
/**
 * this function will draw Calendar Month Table
 *
 * @param object data this contains the calendar data
 * @param object option this is the settings object
 * @return html
 */
function drawCalendarMonthTable(data, option) {

 var
 table,
 div, container, elem;

 //get table
 table = createMonthTable(data, option);

 //calendar container
 container = document.createElement("div");
 container.setAttribute("class", "dycalendar-month-container");

 //----- Header -----

 //header div
 div = document.createElement("div");
 div.setAttribute("class", "dycalendar-header");
 div.setAttribute("data-option", JSON.stringify(option));
 // JSON.stringify(option):
 // {"target":"#dycalendar",
 // "type":"month",
 // "highlighttargetdate":true,
 // "prevnextbutton":"show",
 // "month":0,
 // "year":2023,
 // "date":17,
 // "monthformat":"full",
 // "dayformat":"full",
 // "highlighttoday":false}

 //prev button
 if (option.prevnextbutton === "show") {
 elem = document.createElement("span");
 elem.setAttribute("class", "dycalendar-prev-next-btn prev-btn");
 elem.setAttribute("data-date", option.date);
 elem.setAttribute("data-month", option.month);
 elem.setAttribute("data-year", option.year);
 elem.setAttribute("data-btn", "prev");
 elem.innerHTML = "<";
 //add prev button span to header div
 div.appendChild(elem);
 }

 //month span
 elem = document.createElement("span");
```

```

 elem.setAttribute("class", "dycalendar-span-month-year");
 if (option.monthformat === "mmm") {
 elem.innerHTML = data.monthName + " " + data.year;
 } else if (option.monthformat === "full") {
 elem.innerHTML = data.monthNameFull + " " + data.year;
 }
}

//add month span to header div
div.appendChild(elem);

//next button
if (option.prevnexbutton === "show") {
 elem = document.createElement("span");
 elem.setAttribute("class", "dycalendar-prev-next-btn next-btn");
 elem.setAttribute("data-date", option.date);
 elem.setAttribute("data-month", option.month);
 elem.setAttribute("data-year", option.year);
 elem.setAttribute("data-btn", "next");
 elem.innerHTML = ">";
 //add prev button span to header div
 div.appendChild(elem);
}

//add header div to container
container.appendChild(div);

//----- Body -----

//body div
div = document.createElement("div");
div.setAttribute("class", "dycalendar-body");
div.appendChild(table);

//add body div to container div
container.appendChild(div);

//return container
return container;
}

```

- js文件构成的框架



```
.dycalendar-month-container .dycalendar-body table tr td {
 padding : 5px;
}

.dycalendar-month-container .dycalendar-today-date,
.dycalendar-month-container .dycalendar-target-date {
 background-color: #111;
 color : #fff;
 border-radius: 2px;
}
```

|    |              |    |    |    |    |    |   |
|----|--------------|----|----|----|----|----|---|
| <  | January 2023 |    |    |    |    |    | > |
| S  | M            | T  | W  | T  | F  | S  |   |
| 1  | 2            | 3  | 4  | 5  | 6  | 7  |   |
| 8  | 9            | 10 | 11 | 12 | 13 | 14 |   |
| 15 | 16           | 17 | 18 | 19 | 20 | 21 |   |
| 22 | 23           | 24 | 25 | 26 | 27 | 28 |   |
| 29 | 30           | 31 |    |    |    |    |   |

- ```
console.log(option);
console.log(JSON.stringify(option));
```



The screenshot shows a web browser's developer console. At the top, there are two lines of JavaScript code: `console.log(option);` and `console.log(JSON.stringify(option));`. Below the code, the console displays a log message: `{target: "#dycalendar", type: "month", highlighttargetdate: true, prevnextbutton: "show", month: 0, ...}`. Below this, the JSON stringified version of the object is shown: `{"target": "#dycalendar", "type": "month", "highlighttargetdate": true, "prevnextbutton": "show", "month": 0, "year": 2023, "date": 20, "monthformat": "full", "dayformat": "full", "highlighttoday": false}`. At the bottom, it says "Live reload enabled." and there is a link "(index):54".

后者相比于前者缺少了*Prototype*

- 可以看到css中一个元素除了class属性，还可以认为设置其他属性

8.function createMonthTable(data, option)

```
/**
 * this function will create month table.
 *
 * @param object data this contains the calendar data
 * @param object option this is the settings object
 * @return html
 */
function createMonthTable(data, option) {

    var
        table, tr, td,
        r, c, count;

    table = document.createElement("table");
```

```

tr = document.createElement("tr");

//create 1st row for the day letters
for (c = 0; c <= 6; c = c + 1) {
    td = document.createElement("td");
    // 后面表示字符串数组的第几个元素
    td.innerHTML = "SMTWTFS"[c];
    tr.appendChild(td);
}
table.appendChild(tr);

//create 2nd row for dates
tr = document.createElement("tr");

//blank td
for (c = 0; c <= 6; c = c + 1) {
    if (c === data.firstDayIndex) {
        break;
    }
    td = document.createElement("td");
    tr.appendChild(td);
}

//remaing td of dates for the 2nd row
count = 1;
//There is no need to worry about the excessive number
//of td,because tr will not be loaded with extra td.
while (c <= 6) {
    td = document.createElement("td");
    td.innerHTML = count;
    if (data.today.date === count && data.today.monthIndex ===
data.monthIndex && option.highlighttoday === true) {
        td.setAttribute("class", "dycalendar-today-date");
    }
    if (option.date === count && option.month === data.monthIndex &&
option.highlighttargetdate === true) {
        td.setAttribute("class", "dycalendar-target-date");
    }
    tr.appendChild(td);
    count = count + 1;
    c = c + 1;
}

table.appendChild(tr);

//create remaining rows
for (r = 3; r <= 7; r = r + 1) {
    tr = document.createElement("tr");
    for (c = 0; c <= 6; c = c + 1) {
        if (count > data.totaldays) {
            table.appendChild(tr);
            return table;
        }
        td = document.createElement('td');

```

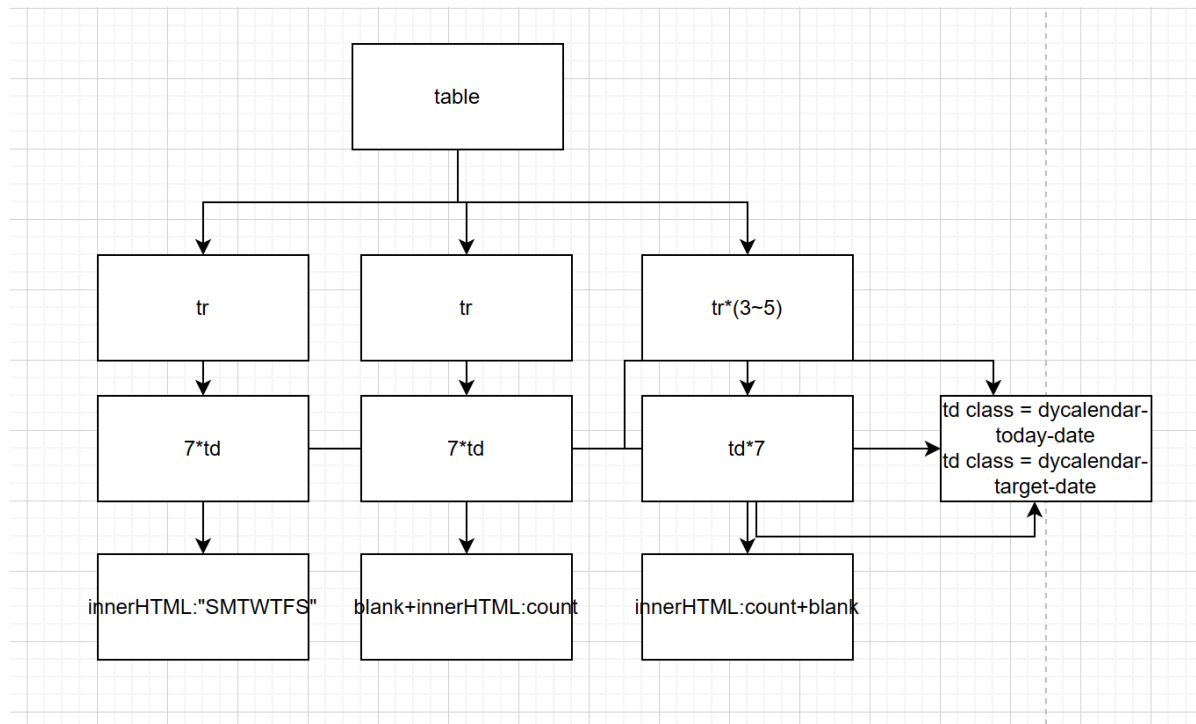


```

        td.innerHTML = count;
        if (data.today.date === count && data.today.monthIndex ===
data.monthIndex && option.highlighttoday === true) {
            td.setAttribute("class", "dycalendar-today-date");
        }
        if (option.date === count && option.month === data.monthIndex &&
option.highlighttargetdate === true) {
            td.setAttribute("class", "dycalendar-target-date");
        }
        count = count + 1;
        tr.appendChild(td);
    }
    table.appendChild(tr);
}

return table;
}

```



9.function onClick()

```

/**
 * this function will handle the on click event.
 */
function onClick() {

    document.body.onclick = function (e) {
        // console.log(e);
        // console.log(global.event);
        //get event object (window.event for IE compatibility)
        e = global.event || e;
        // console.log(e);
        // console.log(e.target);
        // console.log(e.srcElement);
        var

```

```

        //get target dom object reference
        targetDomObject = e.target || e.srcElement,

        //other variables
        date, month, year, btn, option, dateObj;
    // console.log(targetDomObject.classList);
    //prev-next button click
    //extra checks to make sure object exists and contains the class of
interest
    if ((targetDomObject) && (targetDomObject.classList) &&
(targetDomObject.classList.contains("dycalendar-prev-next-btn"))) {
        date = parseInt(targetDomObject.getAttribute("data-date"));
        month = parseInt(targetDomObject.getAttribute("data-month"));
        year = parseInt(targetDomObject.getAttribute("data-year"));
        btn = targetDomObject.getAttribute("data-btn");
        option =
JSON.parse(targetDomObject.parentElement.getAttribute("data-option"));
        // console.log(targetDomObject.parentElement);
        if (btn === "prev") {
            month = month - 1;
            if (month < 0) {
                year = year - 1;
                month = 11;
            }
        }
        else if (btn === "next") {
            month = month + 1;
            if (month > 11) {
                year = year + 1;
                month = 0;
            }
        }
        }

        option.date = date;
        option.month = month;
        option.year = year;

        drawCalendar(option);
    }

    //month click
    //extra checks to make sure object exists and contains the class of
interest
    if ((targetDomObject) && (targetDomObject.classList) &&
(targetDomObject.classList.contains("dycalendar-span-month-year"))) {
        option =
JSON.parse(targetDomObject.parentElement.getAttribute("data-option"));
        dateObj = new Date();

        option.date = dateObj.getDate();
        option.month = dateObj.getMonth();
        option.year = dateObj.getFullYear();

        drawCalendar(option);
    }

```

```
};  
}
```

- ```
console.log(e);
console.log(global.event);
//get event object (window.event for IE compatibility)
e = global.event || e;
console.log(e);
```

当我进行对页面进行鼠标点击时，控制台如下：

```

▶ PointerEvent {isTrusted: true, pointerId: 1, width: 1, height: 1, pressure: 0, ...} dycalendar.js:441
▶ PointerEvent {isTrusted: true, pointerId: 1, width: 1, height: 1, pressure: 0, ...} dycalendar.js:442
▶ PointerEvent {isTrusted: true, pointerId: 1, width: 1, height: 1, pressure: 0, ...} dycalendar.js:445
.
```

这三者区别应该是不大的

- ```
console.log(e.target);  
console.log(e.srcElement);
```

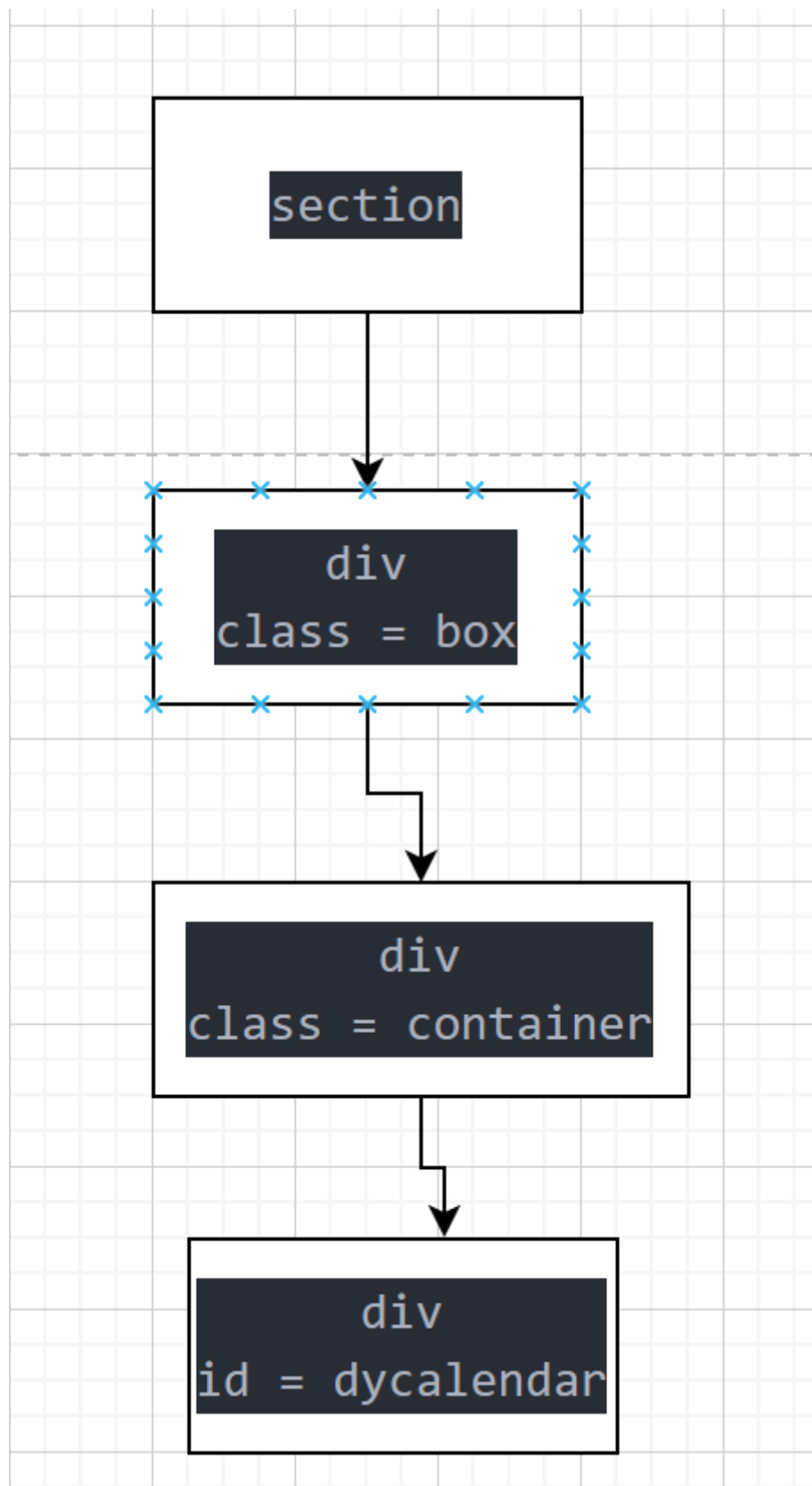
控制台如下：

```
<td>1</td> dycalendar.js:443  
<td>1</td> dycalendar.js:444  
  
<span class="dycalendar-prev-next-btn prev-btn" data-date="20" data-month="0" data-year="2023"  
data-btn="prev"><</span> dycalendar.js:443  
  
<span class="dycalendar-prev-next-btn prev-btn" data-date="20" data-month="0" data-year="2023"  
data-btn="prev"><</span> dycalendar.js:444  
  
<td>T</td> dycalendar.js:443  
<td>T</td> dycalendar.js:444
```

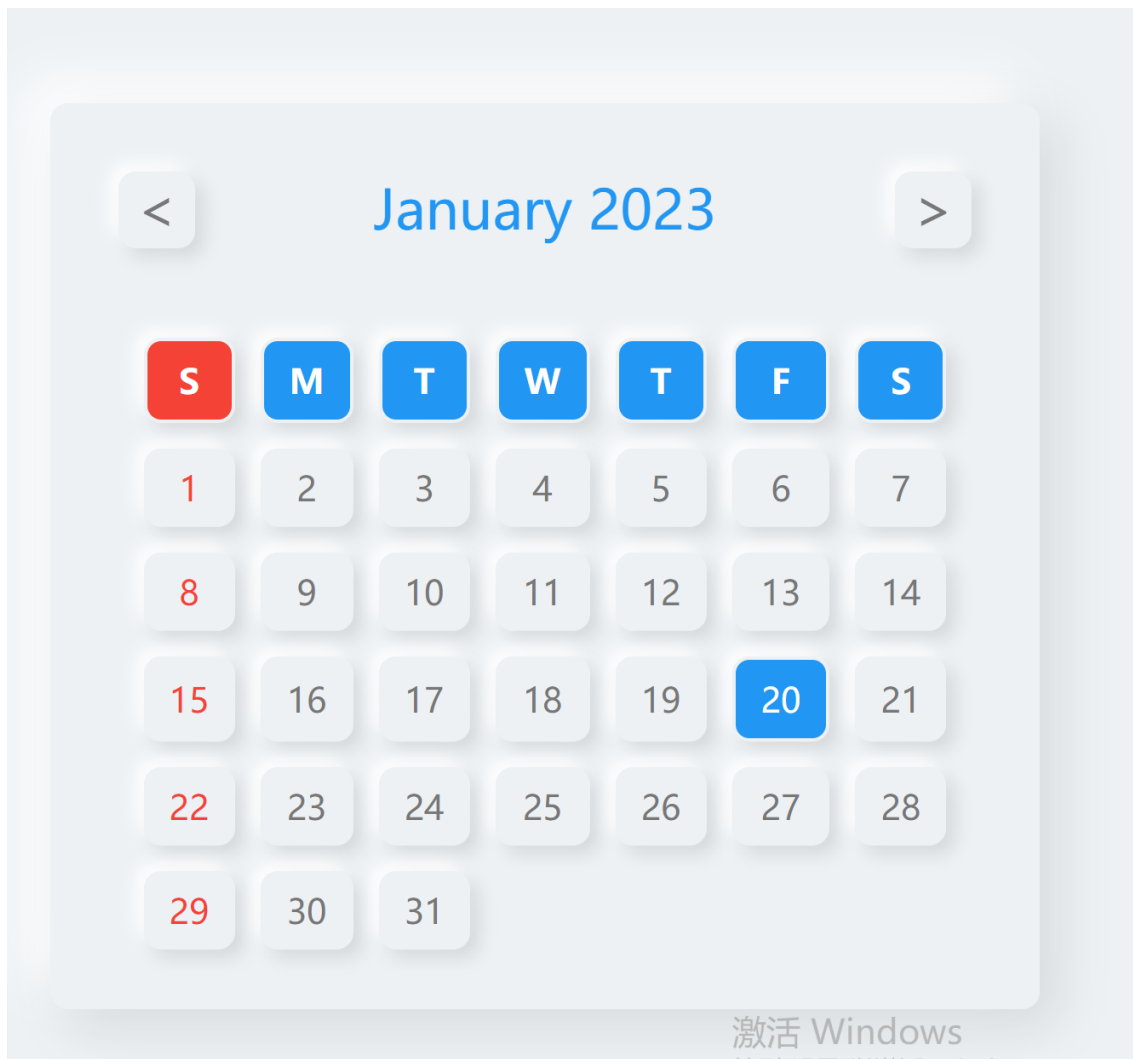
会显示我鼠标点击位置的元素，这两者区别应该也是不大的

9.复杂格式的设置

9.1全局结构



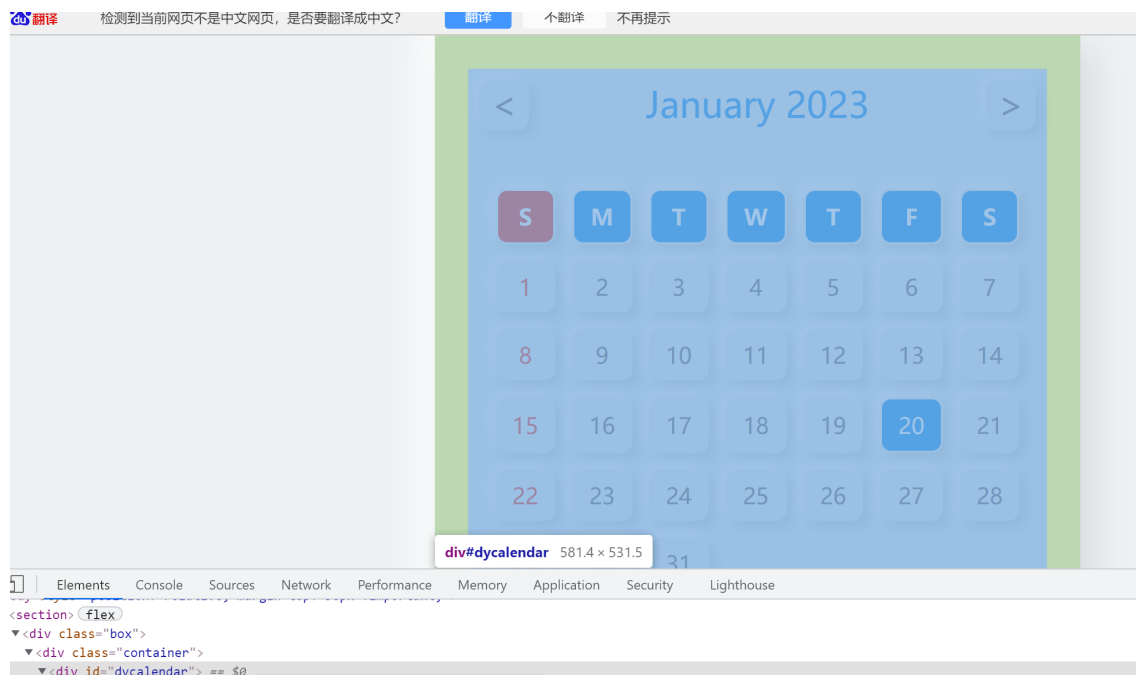
- ```
.container{
 position: relative;
 min-width: 400px;
 border-radius: 10px;
 box-shadow: 10px 10px 30px rgba(0,0,0,0.1), -15px
 -15px 30px #fff;
}
```



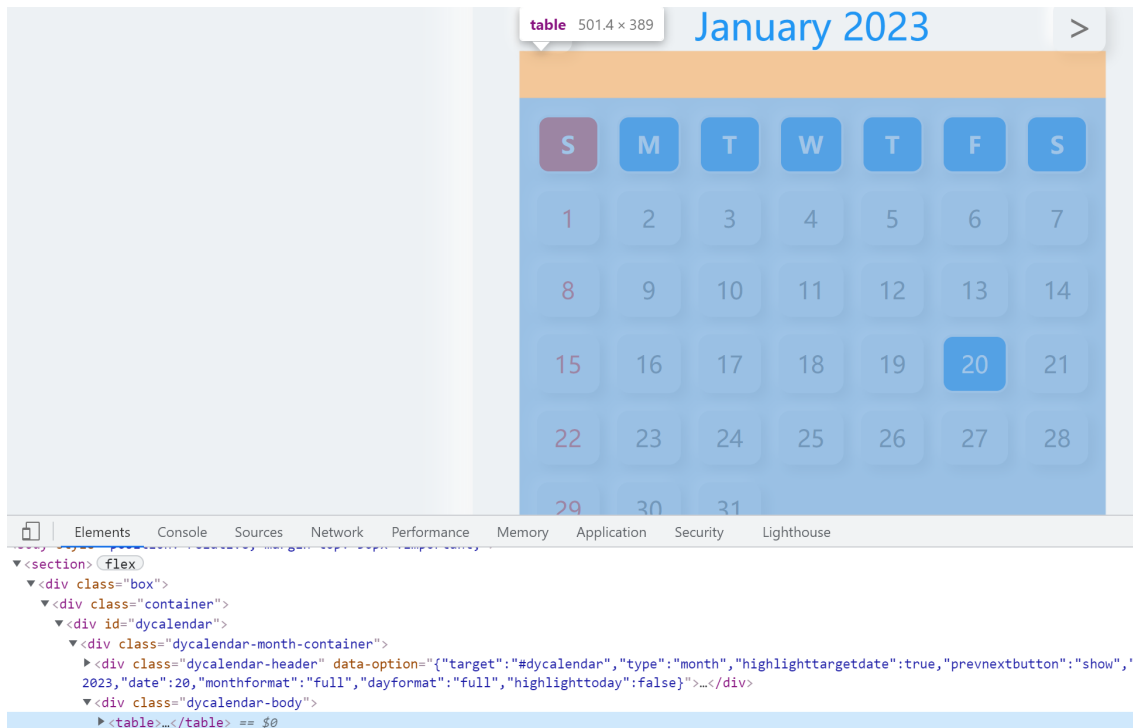
在左部和上部进行白色阴影处理，右侧和下侧进行黑色阴影处理

- ```
#dycalendar{
  width: 100%;
  padding: 30px 30px 0;
}
```

我们从图中更容易看到内边距的设置



- ```
#dycalendar table{
 width: 100%;
 border-spacing: 15px;
 margin-top: 40px;
}
```



可以看到上外边距的设置

- ```
#dycalendar table tr:nth-child(1) td{
  background: #2196f3;
  color: #fff;
  border-radius: 10px;
  font-weight: 600;
  box-shadow: 5px 5px 10px rgba(0,0,0,0.1),
    -5px -5px 10px #fff;
  border: 2px solid #edf1f4;
}

#dycalendar table tr:nth-child(1) td:nth-child(1){
  background: #f44336;
}

#dycalendar table td{
  color: #777;
  padding: 10px 15px;
  cursor: pointer;
  font-weight: 500;
  border-radius: 10px;
  font-size: 1.25em;
  box-shadow: 5px 5px 10px rgba(0,0,0,0.1),
    -5px -5px 10px rgba(255,255,255,1);
}

#dycalendar table td:hover{
  /* 使用 inset 关键字会使得阴影落在盒子内部，这样看起来就像是内容被压低了。
  此时阴影会在边框之内（即使是透明边框）、背景之上、内容之下。*/
  /*以第一个阴影为例，若没有inset属性，那么阴影的位置会在右外（右右）和下外（下下）；
  若有inset属性，那么阴影位置就会处在左内（左右）上内（上下）*/
}
```

```

        box-shadow: inset 5px 5px 10px rgba(0,0,0,0.1),
        -5px -5px 10px rgba(255,255,255,1);
    }
    /*会对每一个tr的第一个td生效*/
    #dycalendar table td:first-child{
        color: #f44336;
    }

```

- 注意到虽然#dycalendar table td在#dycalendar table tr:nth-child(1) td后，但后者所限制的范围更小，因此后者的优先级更高

- ```

.dycalendar-month-container .dycalendar-body .dycalendar-target-date
{
 background: #2196f3;
 /*此important何意？我发现去掉它之后#fff色不会生效
 所以说!important是让此属性生效？*/
 color: #fff !important;
 box-shadow: 5px 5px 10px rgba(0,0,0,0.1),
 -5px -5px 10px rgba(255,255,255,1) !important;
 border: 2px solid #edf1f4;
}
.dycalendar-month-container .dycalendar-header .dycalendar-prev-next-
btn.prev-btn{
 width: 45px;
 height: 45px;
 font-size: 2em;
 left: 15px;
 color: #777;
 background: #edf1f4;
 border-radius: 10px;
 box-shadow: 5px 5px 10px rgba(0,0,0,0.1),
 -5px -5px 10px rgba(255,255,255,1);
}
.dycalendar-month-container .dycalendar-header .dycalendar-prev-next-
btn.next-btn{
 width: 45px;
 height: 45px;
 font-size: 2em;
 right: 15px;
 color: #777;
 background: #edf1f4;
 border-radius: 10px;
 box-shadow: 5px 5px 10px rgba(0,0,0,0.1),
 -5px -5px 10px rgba(255,255,255,1);
}
.dycalendar-month-container .dycalendar-header .dycalendar-prev-next-
btn.prev-btn:active{
 box-shadow: inset 5px 5px 10px rgba(0,0,0,0.1),
 inset -5px -5px 10px rgba(255,255,255,1);
}
.dycalendar-month-container .dycalendar-header .dycalendar-prev-next-
btn.next-btn:active{
 box-shadow: inset 5px 5px 10px rgba(0,0,0,0.1),
 inset -5px -5px 10px rgba(255,255,255,1);
}

```

```
.dycalendar-span-month-year{
 color: #2196f3;
 font-size: 2em;
 font-weight: 400;
}
```

- 一件有意思的事情

```
.dycalendar-month-container .dycalendar-header .dycalendar-prev-nex
/* left : 0; */
}

.dycalendar-month-container .dycalendar-header .dycalendar-prev-nex
/* right : 0; */
}

.dycalendar-month-container .dycalendar-header .dycalendar-prev-nex
width: 45px;
height: 45px;
font-size: 2em;
left: 25px;
color: #777;
background: #edf1f4;
border-radius: 10px;
box-shadow: 5px 5px 10px rgba(0,0,0,0.1),
-5px -5px 10px rgba(255,255,255,1);
}

.dycalendar-month-container .dycalendar-header .dycalendar-prev-nex
width: 45px;
height: 45px;
font-size: 2em;
right: 15px;
color: #777;
background: #edf1f4;
border-radius: 10px;
```

```
<link rel="stylesheet" href="style.css">
<link rel="stylesheet" href="dycalendar.css">
```

在两个css中均有对btn相对位置的设置，生效的是靠后的css文件

- 什么情况下会触发元素的active状态？ 敲击