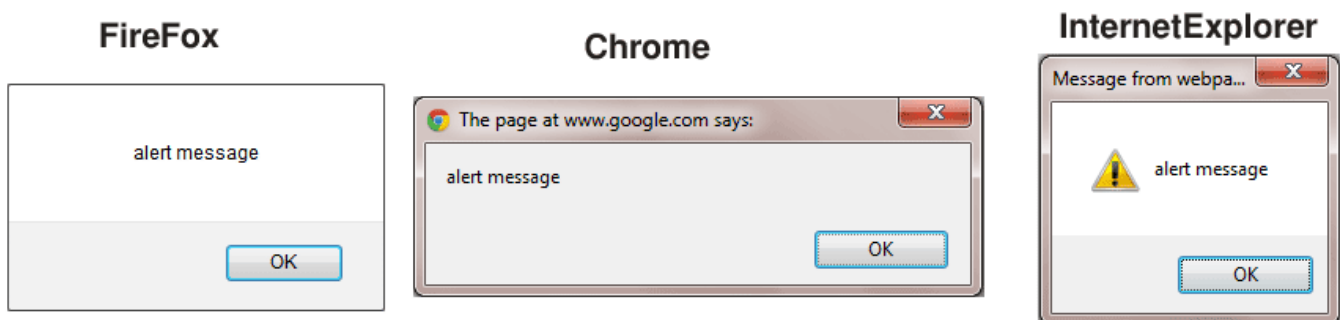
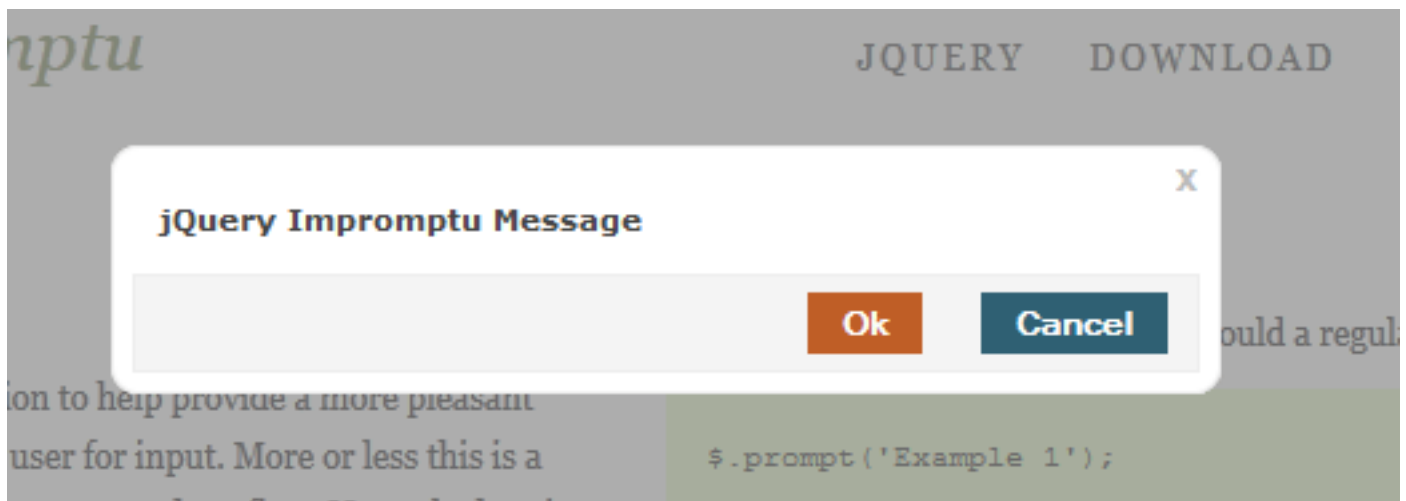


alert,confirm,prompt سه متد توکار JavaScript هستند که برای نمایش پیام، دریافت تایید و دریافت مقدار از کاربر هستند.

گرافیک این پیام‌ها هم وابسته به مرورگر هستند و قابل تغییر نیستند. متن عنوان و دکمه‌ها هم با توجه به زبان سیستم عامل تعیین می‌شوند و قابل تغییر توسط برنامه نویسی نیستند.



حال مواقعی پیش می‌آید که نیاز داریم پیام‌هایی با گرافیک و عبارات متفاوت نمایش دهیم. برای رفع این نیاز می‌توانیم از پلاگین [jQuery Impromptu](#) استفاده کنیم. البته این پلاگین قابلیت‌های دیگری هم دارد که در این مقاله با آنها آشنا می‌شویم.



از ویژگی‌های این پلاگین می‌توان حجم کم (حدود 11 کیلوبایت) و قدرت شخصی سازی بالا اشاره کرد.

طرز استفاده به این شکل است:

```
$.prompt( msg , options )
```

msg می‌تواند یک string یا یک شیء از states باشد . string ارسال شده می‌تواند شامل کدهای html باشد .

یک شیء states هم شامل مجموعه ای از وضعیت‌های prompt است . برای مثال می‌توان یک prompt ایجاد کرد که مثل یک Wizard شامل چند مرحله باشد .

options هم تنظیماتی است که می‌توان مشخص کرد . تنظیماتی مثل : ...,prefix,classes,persistent,timeout

msg :

گفتیم شیء states شامل وضعیت‌های مختلف prompt است . هر وضعیت (state) می‌تواند شامل بخش‌های زیر باشد :

html : مقدار Html وضعیت

buttons : یک شیء شامل متن و مقدار دکمه‌هایی که کاربر می‌تواند کلیک کند

focus : ایندکس دکمه‌ی focus شده در وضعیت

submit : تابعی که زمانی که یکی از دکمه‌های وضعیت انتخاب شود فراخوانی می‌شود .

اگر در این تابع false بازگشت داده شود یا متد preventDefault از event فراخوانی شود ، prompt باز می‌ماند . (روشی برای جلوگیری از بسته شدن prompt هنگام تغییر state یا اعتبار سنجی فرم)
همچنین شیء event شامل state (المنت state) و stateName (نام state) می‌باشد .

پیشفرض :

```
function(event, value, message, formVals){}
```

value مقدار دکمه ای است که بروی آن کلیک شده ، message مقدار html تعریف شده برای state است ، formVals هم در صورتی که در html تعریف شده برای state ، المنت‌های فرم وجود داشته باشد ، شامل نام/مقادیر آنها می‌باشد . (برای دریافت مقادیر فرم ، باید از نام المنت استفاده نمایید .)

position : مشخص کننده‌ی موقعیت state که شامل موارد زیر است :

```
position: { container: '#container', x: 0, y: 0, width: 0, arrow: 'lm' }
```

container : selector المنتی است که state باید در آن مکان قرار بگیرد .

x/y : موقعیت نسبی prompt نسبت به container

arrow : جهت نمایش فلش prompt است که می‌تواند یکی از این مقادیر باشد : tl, tc, tr, rt, tm, tb, br, bc, bl, lb, lm, lt

نحوه تعریف یک wizard ساده با شیء states :

```
var tourSubmitFunc = function (e, v, m, f) {
  if (v === -1) {
    $.prompt.prevState();
  }
}
```

```

        return false;
    } else if (v === 1) {
        $.prompt.nextState();
        return false;
    }
};

var states =
{
    state0:
    {
        html: "State1",
        buttons: { Next: 1 },
        //position: { container: '#container', x: 10, y: 0, width: 350, arrow: 'lm' },
        submit: tourSubmitFunc
    },
    state1:
    {
        html: "State2",
        buttons: { Prev: -1, Next: 1 },
        submit: tourSubmitFunc
    },
    state2:
    {
        html: "State3",
        buttons: { Prev: -1, Done: 0 },
        submit: tourSubmitFunc
    }
};

$.prompt(states);

```

تا به اینجا با پارامتر اول prompt آشنا شدیم و فهمیدیم که می‌توانیم یک رشته یا یک شی states به عنوان message به prompt ارسال کنیم .

options :

اکنون با option های prompt (پارامتر دوم) آشنا خواهیم شد .

توجه کنید که زمانی که یک رشته به prompt ارسال کنید ، مقادیر buttons, focus, submit از این تنظیمات دریافت می‌شود . به عبارت دیگر ، زمانی که یک شی states به prompt ارسال کنید ، از مقادیر فوق که در تنظیمات است ، استفاده نمی‌شود .

loaded

یک تابع که زمانی که prompt کامل بارگزاری شده فراخوانی می‌شود .

```

$.prompt("Message",
{
    loaded: function() {
        alert("Prompt Loaded !");
    }
});

```

submit

یک تابع که زمانی که یکی از دکمه‌های state کلیک شود ، فراخوانی می‌شود . (زمانی اتفاق می‌افتد که یک رشته به عنوان متن به prompt ارسال کرده باشید و زمانی که یک شی states ارسال می‌کنید ، هنگام کلیک دکمه‌های آنها ، این تابع فراخوانی نمی‌شود .) پیشفرض :

```
function(event){}
```

statechanging

یک تابع که زمانی که یک state در حال تعویض شدن هست فراخوانی می‌شود .
پیشفرض :

```
function(event, fromStateName, toStateName){}
```

برای لغو تغییر state ، مقدار return false کنید یا متد preventDefault از event را فراخوانی کنید .

statechanged

یک تابع که زمانی که یک state در حال تعویض شدن هست فراخوانی می‌شود .
پیشفرض :

```
function(event, toStateName){}
```

callback

یک تابع که زمانی که (یکی از دکمه‌های prompt کلیک شود و) prompt بسته شود ، فراخوانی می‌شود .
پیشفرض :

```
function(event[, value, message, formVals]){} 
```

سه پارامتر آخر تنها زمانی که یک دکمه‌ی prompt کلیک شده باشد موجود هستند .

buttons

یک شیء شامل مجموعه ای از دکمه‌ها .
پیشفرض :

```
{ Ok : true }
```

شکل دیگر تعریف دکمه به این شکل است :

```
[
  {title: 'Hello World',value:true},
  {title: 'Good Bye',value:false}
]
```

prefix

یک پیشوند برای همه css class ها و id های المنت‌های html که توسط prompt ایجاد می‌شود .
پیشفرض : jq_

classes

یک css class که به بالاترین سطح prompt داده می‌شود .
در حالت پیشفرض مقداری ندارد .

focus

ایندکس دکمه‌ی focus شده
پیشفرض : 0

zIndex

zIndex اعمال شده بروی prompt .

پیشفرض : 999

useiframe

استفاده از یک iframe برای overlay در IE6

پیشفرض : false

top

فاصله‌ی prompt از بالای صفحه

پیشفرض : 15%

opacity

میزان شفافیت لایه‌ی ای که صفحه را پوشانده است .

پیشفرض : 0.6

overlayspeed

سرعت نمایش افکت fadeIn , fadeOut لایه‌ی پوشاننده .

مقادیر قابل قبول : "slow", "fast", number(milliseconds)

پیشفرض : "slow"

promptspeed

سرعت نمایش prompt .

مقادیر قابل قبول : "slow", "fast", number(milliseconds)

پیشفرض : "fast"

show

نام یک متد jQuery برای animate کردن نمایش prompt .

مقادیر قابل قبول : "show", "fadeIn", "slideDown", ...

پیشفرض : "promptDropIn"

persistent

بسته شدن prompt زمانی که بروی لایه‌ی fade کلیک شود .

بسته شدن : false

پیشفرض : true

timeout

مدت زمانی که پس از آن , prompt بصورت خودکار بسته می‌شود . (milliseconds)

پیشفرض : 0

: returns

مقدار بازگشتی متد prompt , یک شیء jQuery , شامل همه‌ی محتویات تولید شده توسط prompt است .

: Events using Bind

استفاده از Event ها با بایند کردن

promptloaded

معادل loaded در option ها .

promptsubmit

هنگام submit شدن (کلیک شدن یکی از دکمه های) state فعال می شود .

promptclose

معادل callback در option ها .

promptstatechanging

معادل statechanging در option ها .

promptstatechanged

معادل statechanged در option ها .

ظرز بایند کردن یک event به شئی prompt :

```
var myPrompt = $.prompt( msg , options );  
myPrompt.bind('promptloaded', function(e){});
```

: Helper Functions

(jQuery.prompt.setDefaults(options

تعیین مقادیر پیشفرض برای همه ی prompt ها .

```
jQuery.prompt.setDefaults({  
  prefix: 'myPrompt',  
  show: 'slideDown'  
});
```

(jQuery.prompt.setStateDefaults(options

تعیین مقادیر پیشفرض برای state ها .

```
jQuery.prompt.setStateDefaults({  
  buttons: { Ok:true, Cancel:false },  
  focus: 1  
});
```

(jQuery.prompt.getCurrentState

یک شئی jQuery از state جاری برمی گرداند .

(jQuery.prompt.getCurrentStateName

نام state جاری را برمی گرداند .

(jQuery.prompt.getStateContent(stateName

یک شیء jQuery از state مشخص شده برمی‌گرداند .

(jQuery.prompt.goToState(stateName, callback

state مشخص شده را در prompt نمایش می‌دهد .

callback تابعی است که بعد از تغییر state فراخوانی می‌شود .

(jQuery.prompt.nextState(callback

prompt را به state بعدی منتقل می‌کند .

(jQuery.prompt.prevState(callback

prompt را به state قبلی منتقل می‌کند .

(jQuery.prompt.close

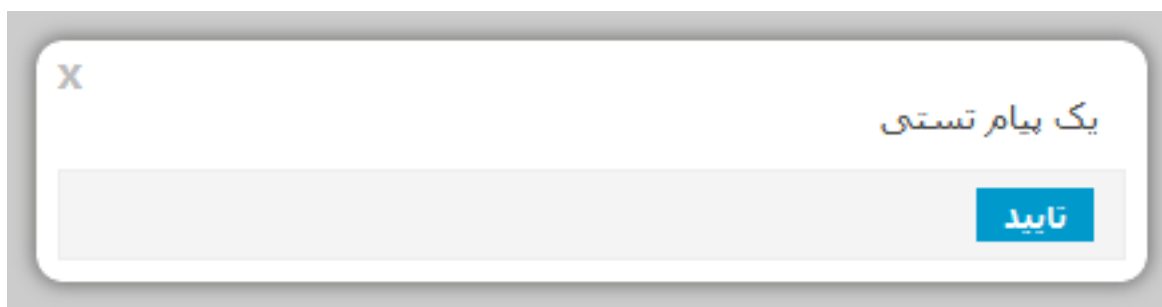
prompt را می‌بندد .

حال که با این پلاگین آشنا شدیم ، سه دستور جاوا اسکریپتی که در ابتدای مقاله ذکر کردیم را با این پلاگین پیاده سازی می‌کنیم .

alert

```
$.prompt("یک پیام تستی",  
{  
  prefix: 'dnt',  
  buttons: { 'تایید': true }  
});
```

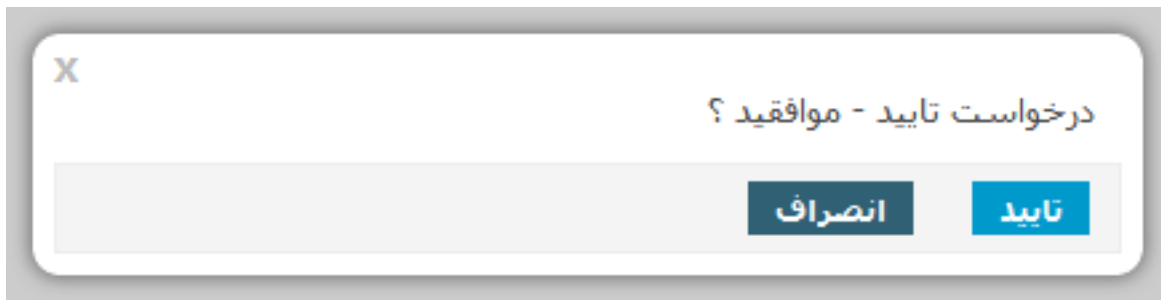
نتیجه :



confirm

```
$.prompt("درخواست تایید - موافقید ؟",  
{  
  prefix: 'dnt',  
  buttons: { 'تایید': true, 'انصراف': false }  
});
```

نتیجه :



prompt

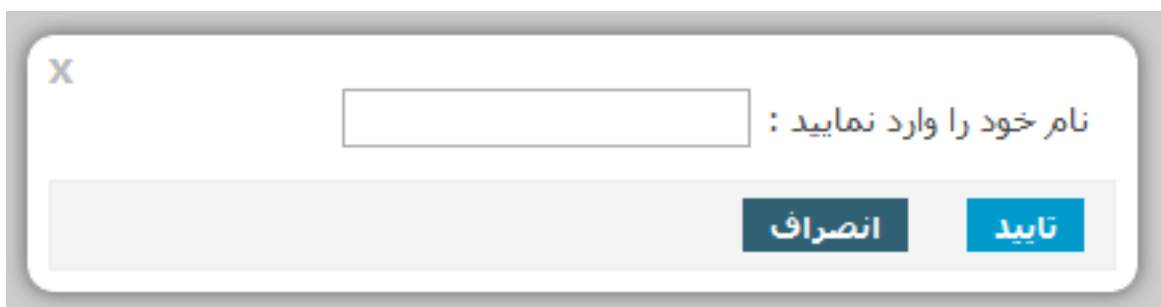
یک فرم html مخفی برای نمایش در prompt :

```
<div class="prompt-content" style="display: none;">
  <span>نام خود را وارد نمایید</span>
  <span>
    <input type="text" name="name" />
  </span>
</div>
```

نمایش prompt :

```
$.prompt(
  $(".prompt-content").html(),
  {
    prefix: 'dnt',
    buttons: { 'تایید': true, 'انصراف': false }
  });
```

نتیجه :



در این سه مثال آخر ، از یک css متفاوت استفاده کردیم . این پلاگین یک سری کلاس دارد که نام این کلاس ها از ترکیب مقدار prefix که در option مشخص کردیم حاصل می شود .
برای مثال اگر مقدار prefix را برابر با dnt قرار بدهیم ، برای استایل دهی متن پیام باید از کلاس div.dnt .dntmessage استفاده کنید .
همانطور که در سه مثال آخر مشاهده کردید ، تغییری در استایل prompt داشتیم که با تغییر دادن استایل های مورد نظر انجام شد .

برای تهیه ی یک قالب سفارشی باید selector المنت هایی که نیاز به تغییر دارند را از [قالب پیشفرض](#) پیدا کنید ، سپس قسمت prefix از selector را به نام قالب خودتان تغییر بدید و استایل را ویرایش کنید . سپس هنگام ایجاد prompt ، مقدار prefix را برابر نام قالب قرار دهید .

مثلا می خواهید یک قالب به نام dnt داشته باشید . می خواهید متن قالب راست به چپ باشد .

```
div.dnt .dntmessage {
    color: #444444;
    line-height: 20px;
    padding: 10px;
    /*      Edited      */
    direction: rtl;
    text-align: right;
}
```

البته راه بهتری هم هست که نیاز به آشنایی با فایرباگ دارد . در این روش ابتدا کل قالب jqz (موجود در [قالب پیشفرض](#)) را در برنامه خود کپی می کنیم ، مقادیر jqz را با نام قالب جایگزین می کنیم ، مقدار prefix را در prompt برابر با نام قالب قرار می دهیم . اکنون در Firefox یک prompt ایجاد می کنیم و توسط فایرباگ استایل هایی که با نام قالب بروی prompt اعمال شده اند را مطابق سلیقه تغییر می دهیم . در مرحله آخر به تب CSS در فایرباگ می رویم و کل استایل های مربوط به قالب را کپی و جایگزین استایل قبلی در برنامه می کنیم .

قابلی که بنده برای سه دستور فوق استفاده کردم (dnt) ، به این شکل است :

```
/*      Start : DotNetTips Theme      */

.dntfade {
    background-color: #AAAAAA;
    position: absolute;
}

div.dnt {
    background-color: #FFFFFF;
    border-radius: 10px 10px 10px 10px;
    border: 1px solid #FFFFFF;
    box-shadow: 0px 0px 10px 1px #6D6D6C;
    font-family: tahoma;
    font-size: 11px;
    padding: 7px;
    position: absolute;
    text-align: left;
    width: 400px;
}

div.dnt .dntcontainer {
    font-size: small;
}

div.dnt .dntclose {
    color: #BBBBBB;
    cursor: pointer;
    font-weight: bold;
    position: absolute;
    top: 4px;
    width: 18px;
}

div.dnt .dntmessage {
    color: #444444;
    line-height: 20px;
    padding: 10px;
}

div.dnt .dntbuttons {
```

```
background-color: #F4F4F4;
border: 1px solid #EEEEEE;
padding: 5px 0px;
text-align: right;
}

div.dnt button {
background-color: #2F6073;
border: 1px solid #F4F4F4;
color: #FFFFFF;
font-size: 12px;
font-weight: bold;
margin: 0px 10px;
padding: 3px 10px;
}

div.dnt button:hover {
background-color: #728A8C;
}

div.dnt button.dntdefaultbutton {
background-color: #0099CC;
}

.dnt_state {
direction: rtl;
text-align: right;
}

.dnt_state button {
font-family: tahoma;
}

.dntwarning .dnt .dntbuttons {
background-color: #CCDDFF;
}

.dnt .dntarrow {
border: 10px solid transparent;
font-size: 0px;
height: 0px;
line-height: 0;
position: absolute;
width: 0px;
}

.dnt .dntarrowtl {
border-bottom-color: #FFFFFF;
left: 10px;
top: -20px;
}

.dnt .dntarrowtc {
border-bottom-color: #FFFFFF;
left: 50%;
margin-left: -10px;
top: -20px;
}

.dnt .dntarrowtr {
border-bottom-color: #FFFFFF;
right: 10px;
top: -20px;
}

.dnt .dntarrowbl {
border-top-color: #FFFFFF;
bottom: -20px;
left: 10px;
}

.dnt .dntarrowbc {
border-top-color: #FFFFFF;
bottom: -20px;
left: 50%;
margin-left: -10px;
}

.dnt .dntarrowbr {
border-top-color: #FFFFFF;
bottom: -20px;
```

```
    right: 10px;
}

.dnt .dntarrowlt {
    border-right-color: #FFFFFF;
    left: -20px;
    top: 10px;
}

.dnt .dntarrowlm {
    border-right-color: #FFFFFF;
    left: -20px;
    margin-top: -10px;
    top: 50%;
}

.dnt .dntarrowlb {
    border-right-color: #FFFFFF;
    bottom: 10px;
    left: -20px;
}

.dnt .dntarrowrt {
    border-left-color: #FFFFFF;
    right: -20px;
    top: 10px;
}

.dnt .dntarrowrm {
    border-left-color: #FFFFFF;
    margin-top: -10px;
    right: -20px;
    top: 50%;
}

.dnt .dntarrowrb {
    border-left-color: #FFFFFF;
    bottom: 10px;
    right: -20px;
}

/*      End : DotNetTips Theme      */
```

برای مشاهده مثال‌های بیشتر به صفحه‌ی اصلی [jQuery Impromptu](#) مراجعه نمایید .

نظرات خوانندگان

نویسنده: بهمن خلفی
تاریخ: ۱۳۹۱/۰۶/۱۹ ۱۰:۳۶

با سلام و تشکر از شما

نمونه پست ارسالی شما در این [آدرس](#) نیز موجود است لطفا تفاوت کارایی های این دو را بفرمائید و از طرفی اگر امکان دارد نحوه استفاده آن در دات نت را برای تأیید و دریافت ورودی توضیح دهید باتشکر

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۶/۱۹ ۲۳:۵۷

- تفاوت مهم آن در حجم کمتر است. jQuery-UI نسبتا حجم بالایی دارد و اگر صرفا قرار است پیام کوتاهی به کاربری نمایش داده شود، روش فوق حداقل از لحاظ حجم (11K) مقرون به صرفه تر است.
- استفاده از نتایج آن هم مطابق توضیحات مفصل مطلب جاری، در قسمت callback آن باید صورت گیرد. [برای مثال](#)

نویسنده: سیروان عقیفی
تاریخ: ۱۳۹۱/۰۶/۲۰ ۱۴:۵۹

ممنون،

یه سوال بنده یه فرم ثبت نام دارم که با کلیک بر روی Button اطلاعات در دیتابیس درج می شود حالا می خوام بعد از ثبت اطلاعات یه alert به کاربر نشون بده که اطلاعات با موفقیت ثبت شد ولی مشکل اینجاست که وقتی کاربر روی Button کلیک میکنه چون صفحه PostBack میشه دیگه پیغام به کاربر نمایش داده نمیشه، اگه امکان داره بنده رو راهنمایی بفرمائید. دقیقا مثل همین قسمت ارسال نظر سایت.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۶/۲۰ ۱۵:۵۹

[این مطلب](#) می تونه مفید و مرتبط باشه.

نویسنده: رضا
تاریخ: ۱۳۹۱/۰۷/۱۲ ۲:۳۲

سلام. یک linkbutton دارم تو گرید ویو که کار حذف رو انجام میده. در حالت پیش فرض با دستور

```
return confirm('')
```

و انتساب اون به خاصیت onClick می شد یک confirm ساده قبل حذف داشت. حالا وقتی میخام از این کنترل برای این کار استفاده کنم کار نمیکنه؟ میشه راهنمایی کنید؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۷/۱۲ ۹:۴۶

مثالی که در [این کامنت](#) معرفی شد مرتبط به GridView هم هست (انتهای مقاله).

نویسنده: omid

تاریخ: ۸:۱۹ ۱۳۹۱/۱۰/۱۲

سلام

با تشکر از راهنمایی شما برای کنترل‌ها من از این کنترل می‌خواستم استفاده کنم ولی اجرا نشد اگه راهنمایی کنید ممنون میشم .

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html lang="en" xmlns="http://www.w3.org/1999/xhtml">

<head>
<meta content="text/html; charset=UTF-8" http-equiv="Content-Type" />
<title></title>

<link href="jquery-impromptu.css" media="all" rel="stylesheet" type="text/css" />
<script src="jquery-1.8.3.min.js" type="text/javascript"></script>
<script src="jquery-impromptu.js" type="text/javascript"></script>
<script type="text/javascript">

$(function(){
$show.click(function(e){
$.prompt("Hello World!");
});
});
});

</script>
</head>

<body>
<button class="show">ShowPrompt</button>
</body>
</html>
```

ممنون

نویسنده: وحید نصیری
تاریخ: ۱۰:۲۴ ۱۳۹۱/۱۰/۱۲

- کد شما تعداد {} متوازی ندارد. دو {} بازه شده، سه تا بسته.
- به نظر با آخرین نگارش jQuery سازگار نیست. به صفحه خطای مرورگر مراجعه کنید تا خطای شناخته نشدن نوع مرورگر قابل مشاهده باشد. با نگارش‌های پایین‌تر [مشکلی ندارد](#) .

نویسنده: omid
تاریخ: ۱۰:۵۶ ۱۳۹۱/۱۰/۱۲

سلام

خیلی ممنون بابت راهنمایی‌تون و آشنایی با سایت <http://jsfiddle.net>

نویسنده: neda
تاریخ: ۱۲:۲۶ ۱۳۹۲/۰۱/۰۹

ممنون از سایت خیلی خوب و مفیدتون
ممنون میشم اگه امکانش هست یه سمپل ساده و کوچیک از این کار هم در اختیار بگذارید
من از کدها استفاده کردم ولی موفق نشدم خروجی صحیح بگیرم
ممنونم

نویسنده: احمد احمدی
تاریخ: ۱۷:۱۷ ۱۳۹۲/۰۱/۰۹

```
<!DOCTYPE html>
<html>
```

```
<head>
<link rel="stylesheet" media="all" type="text/css" href="http://trentrichardson.com/Impromptu/jquery-impromptu.css" />
<script type="text/javascript" src="http://code.jquery.com/jquery-1.9.0.min.js"></script>
<script type="text/javascript" src="http://trentrichardson.com/Impromptu/jquery-impromptu.js"></script>
</head>
<body>
<button class="show">ShowPrompt</button>
<script type="text/javascript">
$(function(){
    $(".show").click(function(e){
        $.prompt("Hello World!");
    });
});
</script>
</body>
</html>
```

نویسنده: سمیرا

تاریخ: ۱۳:۴۴ ۱۳۹۲/۰۳/۰۸

سلام؛ وقتی با استفاده از jQuery Impromptu یک inputbox ایجاد می‌کنیم و کاربر اطلاعاتی رو توی inputbox وارد می‌کنه، چجوری می‌تونیم مقداری که کاربر وارد کرده رو برگردونیم و ارزش استفاده کنیم ؟

نویسنده: مژده

تاریخ: ۱۴:۲۳ ۱۳۹۲/۰۳/۰۸

سلام

من هم همین مشکلو داشتم

از این روش استفاده کردم جواب داد :

```
var myval = m.find('#name').val();
alert(myval);
```

نویسنده: احمد احمدی

تاریخ: ۱۸:۴۴ ۱۳۹۲/۰۳/۰۸

سلام؛

طبق مطلب ارائه شده در مقاله:

تابع submit :

```
function(event, value, message, formVals){}
```

value مقدار دکمه ای است که بروی آن کلیک شده ، message مقدار html تعریف شده برای state است ، formVals هم در صورتی که در html تعریف شده برای state ، المنت‌های فرم وجود داشته باشد ، شامل نام/مقادیر آنها می‌باشد . (برای دریافت مقادیر فرم ، باید از نام المنت استفاده نمایید .)

خب حالا مشابه مثال زیر عمل کنید:

```
<div class="prompt-content" style="display: none;">
<span>ایمیل خود را وارد نمایید</span>
<span>
    <input type="text" name="user_email" />
</span>
```

</div>

```
$.prompt(
    $(".prompt-content").html(),
    {
        submit: function (e, v, m, f) {
            var userEmail = f["user_email"];
            console.log(userEmail);
        }
    }
);
```

نویسنده:

مژده

تاریخ:

۲۱:۱۳ ۱۳۹۲/۰۳/۰۹

سلام

خسته نباشید

من توی سایتیم از ajax استفاده کردم و برای حذف یه رکورد اول از کاربر تاییدیه گرفتم . اگه کاربر روی دکمه‌ی بله کلیک کنه ، عملیات شروع میشه و رکورد حذف میشه .

آخر سر توی تابعی که نتیجه برمیگرده توش (handleServerResponse)، بر اساس نتیجه‌ی برگشتی پیغام مناسبی باید نمایش داده بشه. اما هیچ پیغامی نشون داده نمیشه

خیلی چیزا رو آزمایش کردم و آخر سر به این نتیجه رسیدم که چون دو تا پیغام میخواد پشت سر هم نمایش داده بشه ، این مشکل پیش میاد :

حتی وقتی که پیغام تایید حذف رو برداشتم ، دیدم که پیغام دوم میاد !

والااا نمیدونم علتش چیه ! ممنون میشم کمک کنید.

اینم قسمتی از کدمه :

```
function deleteFile(location, filename)//حذف فیزیکی و منطقی فایل
{
    $.prompt("؟ آیا برای حذف فایل موجود اطمینان دارید", {
        title: '',
        buttons: { "بله": true, "خیر": false },
        focus: 2,
        submit: function (e, v, m, f) {
            if (v == true) {
                var getdate = new Date(); //Used to prevent caching during ajax call
                if (xmlhttp) {
                    $(".p#vtip").fadeOut("slow").remove(); //محو شدن tooltip
                    var id = document.getElementById("id").value;
                    var i = '3';
                    params = "id=" + id + "&i=" + i + "&filename=" + filename + "&location=" +
location;

                    xmlhttp.open("POST", "FetchData/dbSearchDocument1.php", true);
                    xmlhttp.onreadystatechange = handleServerResponse4;
                    xmlhttp.setRequestHeader('Content-Type', 'application/x-www-form-urlencoded');
                    xmlhttp.send(params);
                }
            }
            else {
            }
        }
    });
}
//-----
function handleServerResponse4() {
    if ((xmlhttp.readyState == 1) || (xmlhttp.readyState == 2) || (xmlhttp.readyState == 3))//loading
    {
        document.getElementById("content").innerHTML = "<img src='Images/bigloading.gif' />";
    }
    else if (xmlhttp.readyState == 4) //ready
    {
        if (xmlhttp.status == 200) {
            if (xmlhttp.responseText == 1) {
                $.prompt("، حذف فایل با موفقیت انجام شد", {
```

```

        title: '',
        buttons: { "بستن": true }
    });
}
else {
    $.prompt("حذف فایل با خطا مواجه شده است", {
        title: '',
        buttons: { "بستن": true }
    });
}
}
else {
    alert("Error during AJAX call. Please try again");
}
}
}
//-----

```

نویسنده:

مژده

۲۱:۵۴ ۱۳۹۲/۰۳/۰۹

تاریخ:

من الان یک بار دیگه این مطلبو خوندم و به نظرم رسید که باید از callback استفاده کنم.
اما حتی این توی سایتهم جواب نمیده :

```

function mycallbackfunc(v,m){
    $.prompt('i clicked ' + v);
}
$.prompt('Example 8',{ callback: mycallbackfunc });
});

```

نویسنده:

وحید نصیری

۰:۱۶ ۱۳۹۲/۰۳/۱۰

تاریخ:

زمانیکه از jQuery استفاده می‌کنید، دیگر نباید از Ajax خام استفاده کنید. خود jQuery تابع سازگار با انواع و اقسام مرورگرها رو برای کار با Ajax داره. این مثال تست شده و مشکلی نداره:

```

using System.Web.Mvc;

namespace jQueryImpromptu.Controllers
{
    public class HomeController : Controller
    {
        [HttpGet]
        public ActionResult Index()
        {
            return View();
        }

        [HttpPost]
        public ActionResult Index(string postId, string filename)
        {
            return Content("ok");
        }
    }
}

```

```

@{
    ViewBag.Title = "Index";
    var postUrl = Url.Action(actionName: "Index", controllerName: "Home");
}
<h2>
    Index</h2>
<input type="button" id="btnDelete" value="delete" />

```



```

<div id="infoDiv">
    Some Info ....
</div>
@section JavaScript
{
    <script type="text/javascript">
        $(function () {
            $('#btnDelete').click(function () {
                $.prompt("آیا برای حذف اطلاعات موجود اطمینان دارید ؟",
                {
                    title: 'Title',
                    buttons: { "بله": true, "خیر": false },
                    focus: 2,
                    submit: function (e, v, m, f) {
                        //debugger;
                        if (!v) {
                            return;
                        }
                        var postId = 10;
                        var fileName = "test.jpg"
                        $.ajax({
                            type: "POST",
                            url: '@postUrl',
                            data: JSON.stringify({ postId: postId, fileName: fileName }),
                            contentType: "application/json; charset=utf-8",
                            dataType: "json",
                            complete: function (xhr, status) {
                                var data = xhr.responseText;
                                if (status === 'error' || !data || data == "nok") {
                                    $.prompt("حذف فایل با خطا مواجه شده است !",
                                    {
                                        title: 'Title',
                                        buttons: { "بستن": true }
                                    });
                                }
                                else {
                                    $("#infoDiv").fadeOut("slow").remove();
                                }
                            }
                        });
                    }
                });
            });
        });
    </script>
}

```

نویسنده: مژده

تاریخ: ۱۱:۳۵ ۱۳۹۲/۰۳/۱۱

سلام

منمونی که جواب دادید. خیلی لطف کردید

من تونستم با استفاده از به جز دیگه ، با callback مشکلمو حل کنم . اما حالا این js ، عنوان (title) نداره ! D:

از این سایت استفاده کردم :

<http://www.shiguenori.com/material/jquery.impromptu>

نویسنده: وحید نصیری

تاریخ: ۱۱:۴۴ ۱۳۹۲/۰۳/۱۱

کدی که نوشته شد بر اساس [نسخه GitHub](#) بود.

نویسنده: رضا منصوری

تاریخ: ۲۱:۱۱ ۱۳۹۲/۱۱/۰۶

سلام ، اگه بخوایم مقدار ورودی‌های اکشنمون به صورت داینامیک باشه چیکار باید کنیم؟

منظورم به عنوان مثال مقادیر این خط در مثال شماست

```
var postId = 10;  
var fileName = "test.jpg"
```

چنین مدلی دارم که میخوام دکمه‌ی حذفش به صورت JQuery Ajax باشه

```
@foreach (var item in Model) {  
    <tr>  
        <td>  
            @Html.DisplayFor(modelItem => item.Title)  
        </td>  
        <td>  
            @Html.DisplayFor(modelItem => item.IsRead)  
        </td>  
        <td>  
            @Html.ActionLink("حذف پیام", MVC.User.ActionNames.DeleteMessageSend, MVC.User.Name, new {  
id=item.Id }, new { @class="btn btn-danger" })  
        </td>  
    </tr>  
}
```

ولی مقدار id رو باید به اکشن بفرستم ! اونم تو یه حلقه Foreach . باید کدهای جی کوئری تو حلقه باشه؟ به نظرتون راه حل چیه؟ ممنون

نویسنده: رضا منصوری
تاریخ: ۲۱:۳۶ ۱۳۹۲/۱۱/۰۶

ببخشید تو [این مطلب](#) توضیح دادید.