

# آموزش KnockoutJS در textInput binding

## هدف

این binding یک text box (<input>) یا یک text area (<textarea>) را به یک مشخصه ی view model متصل می کند. و در این صورت بین مشخصه ی view model و مقدار المان امکان آپدیت های دو طرفه را فراهم می کند. برخلاف value binding ، textInput برای انواع ورودی های کاربر (اعم از drag-and-drop و clipboard event ها) امکان آپدیت های فوری را از DOM فراهم می کند.

## مثال:

```
<p>Login name: <input data-bind="textInput: userName" /></p>
<p>Password: <input type="password" data-bind="textInput: userPassword" /></p>
```

ViewModel:

```
<pre data-bind="text: ko.toJSON($root, null, 2)"></pre>
```

```
<script>
  ko.applyBindings({
    userName: ko.observable(""), // Initially blank
    userPassword: ko.observable("abc") // Prepopulate
  });
</script>
```

## پارامترها

- پارامتر اصلی
- KnockoutJS محتوای متن المان را بر روی مقدار پارامتر شما قرار می دهد. تمامی مقادیر قبلی overwrite می شوند.
- اگر این پارامتر مقدار قابل مشاهده ای باشد، در این صورت binding هر زمان که این مقدار تغییر کند مقدار المان را آپدیت می کند. اما اگر این پارامتر قابل مشاهده نباشد، تنها یک بار مقدار المان را تنظیم می کند و دیگر آن را آپدیت نخواهد کرد.
- اگر شما چیزی غیر از عدد یا رشته (برای مثال شیء یا آرایه) را در کدتان قرار دهید، در این صورت متن نمایش داده شده با `yourParameter.toString()` برابری می کند (استفاده از مقادیر غیر عددی یا غیر رشته ای اغلب چندان مفید نیست به همین دلیل بهترین کار استفاده از مقادیر رشته ای و عددی است).

هر زمان که کاربر مقدار را در form control مربوطه ویرایش می کند، KnockoutJS مشخصه ی آن را در view model شما آپدیت می کند. زمانی که value توسط کاربر یا دیگر رویدادهای DOM تغییر می کند KnockoutJS همواره سعی خواهد کرد تا view model شما را آپدیت کند.

- پارامترهای اضافی
- ندارد

## آموزش KnockoutJS-آموزش textInput binding در برابر value binding

هرچند که [value binding](#) می تواند بین text box ها و مشخصه های view model به اجرای binding دو طرفه بپردازد، بهتر است که در صورتی که نیاز به آپدیت های فوری و زنده دارید، از textInput استفاده کنید. تفاوت های اصلی این دو binding در زیر آمده اند:

- آپدیت های فوری  
به صورت پیش فرض value تنها زمانی که کاربر focus را از text box بر می دارد (بر روی جایی خارج از text box کلیک می کند - مترجم - )، textInput مدل شما را بلافاصله بعد از هر کلیک یا دیگر مکانیزم های ورود متن (مانند drag & cut متن، که لزوما باعث بروز هیچ focus change event ای نمیشوند) آپدیت می کند.
- مدیریت رفتارهای عجیب رویداد مرورگر  
مرورگرها در برابر رویدادهایی که در پاسخ به مکانیزم های غیر معمول ورود متن مانند cut کردن، drag کردن یا قبول کردن پیشنهادهایی که به صورت خودکار کامل می شوند (autocomplete suggestions) fire می شوند، رفتار ناپایداری از خود نشان می دهند. value binding با وجود اینکه برای دریافت آپدیت های مربوط به رویدادهای مشخص دارای option های اضافی مانند valueUpdate: afterkeydown است، تمام حالات ورود متن را نمی تواند در تمام مرورگرها پوشش دهد.
- عجیب مرورگرها در مدیریت کرد. و به کمک آن بتوان حتی در پاسخ به روش های غیرمعمول ورود متن آپدیت های فوری و پایداری را در مدل شاهد بود.
- سعی کنید این دو binding را در کنار یکدیگر و در المانی یکسان استفاده نکنید زیرا نتیجه ی مفیدی از این کار حاصل نخواهد شد.

### وابستگی ها

به غیر از کتابخانه ی اصلی KnockoutJS وابستگی دیگری ندارد.