**Parsley.js**

# Data Attributes

Parsley's default DOM API is data-parsley-.

That means that if in config you see a foo property, it can be set/modified via DOM with data-parsley-foo="value".

# Extra Plug-in to Parsley

## Words

Validates that the value is within a certain range of words

data-parsley-words="[200, 600]"

# Form Properties

## Priority-Testing

data-parsley-priority-enabled

Yüksek önceliği olan validasyondan düşük önceliği olan validasyona doğru tek tek gerçekleştirir ve biri fail ettiği anda durup sadece onu gösterir.(Default hali bu) Tüm failları görebilmek için priority-enabled false olmalı.

**Default : True**

## Field selector

data-parsley-inputs

Form field’larını seçmek için kullanır. Eğer field’ın kullanılmasını istemiyorsak excluded kullanmamız gerekir.

**Default: input,textarea,select**

## Exclude

data-parsley-excluded

Validasyona dahil olmayacak alanları filtremede kullanılır.

**Default : input[type=button],input[type=submit],input[type=reset],input[type=hidden]**

data-parsley-excluded="input[type=button], input[type=submit], input[type=reset], input[type=hidden], [disabled], :hidden"

# Form Methods

## whenValid()

whenValid({group, force})

Form validasyonu geçerse tamamlanacak olan jQuery Promise döner.User interface’i etkilemez.Group parametre olarak geçilmişse sadece o gruba ait field’ları validate eder. Force parametre olarak geçilmişse tüm her şeyi validate eder.

**Return: Promise**

## isValid()

User Interface’e etki etmez yeni eventlar tetiklemez.Promise tamamlanmışsa true , reject yediyse false , validasyon bekleniyorsa null  döner.

isValid({group, force})

**Return: boolean or null**

## whenValidate()

Form validasyonu yapar. User interface’i değiştirebilir, yeni event’lar tetikleyebilir.

whenValidate({group, force})

**Return: Promise**

## validate()

User Interface’e etki edebilir. Promise tamamlanmışsa true , reject yediyse false , validasyon bekleniyorsa null  döner.

validate({group, force})

**Return: Boolean or null**

## reset()

Form içeriğini temizler

reset() #2.0

## destroy()

destroy() #2.0

Verilen form ve field’ları için parsleyi disable eder.

# Field Options

## Field Value

Parsley validasyonu için field’e spesifik bir değer set eder.

data-parsley-value

## Grouping

Field’e grup validasyonu için bir grup belirler.Böylelikle formun bir kısmını validate edebiliriz. Birden çok group ismi verebiliriz.

data-parsley-group='["foo", "bar"]'

## Checkbox/Radio Grouping

## Checkbox/Radio’ları gruplamak için kullanılır.

## data-parsley-multiple="mymultiplelink"

## Empty Validation

Opsiyonel bir fieldimizin boş olmasının sıkıntı yaratmaması için kullanılır. Bu sorunu by-pass etmek için eklemeliyiz.

data-parsley-validate-if-empty

## Whitespace Clear

Value’da yer alan boşlukları Parsley validation için siler. Min karakter 10’sa boşluk ekleyerek 8 karakteri 2 karakter yapmasını istemiyorsak kullanabiliriz.

data-parsley-whitespace

Baştaki ve sondaki boşlukları silmek için : data-parsley-whitespace="trim"

Ardarda boşlukları tek boşluğa indirmek için : data-parsley-whitespace="squish"

## Enable

Eğer false olarak verilirse, Parsley bu field için sayfada bir etki yapmaz.

data-parsley-ui-enabled

## No Error Message

Parsley operasyonları gerçekleşir ama hata mesajı gözükmez.

data-parsley-errors-messages-disabled

## Exclude

True set edildiği zaman parsley o field’i yok sayar.

data-parsley-excluded

# Field Methods

## isValid()

UI’yı etkilemez ve event tetiklemez.

isValid({force})

**Return: true if all ok  
 null if some validations are still pending  
 [Violation, ...]] if fails**

## Validate()

UI’yı etkileyebilir ve de event tetikleyebilir.

isValid({force})

## **Return: true if all ok null if some validations are still pending [Violation, ...]] if fails**

## reset()

Field içeriğini temizler

reset() #2.0

## destroy()

destroy() #2.0

Verilen field için parsleyi disable eder.

## getErrorsMessages()

Validate olduğu zaman çıkacak hata mesajlarına döner

## Example

* <input id="first" data-parsley-maxlength="42" value="hello"/> <input id="second" value="world"/> [...] <script> **var** instance = $('#first').parsley(); console.log(instance.isValid()); *// maxlength is 42, so field is valid* $('#first').attr('data-parsley-maxlength', 4); console.log(instance.isValid()); *// No longer valid, as maxlength is 4* *// You can access and override options in javascript too:* instance.options.maxlength++; console.log(instance.isValid()); *// Back to being valid, as maxlength is 5* *// Alternatively, the options can be specified as:* **var** otherInstance = $('#second').parsley({ maxlength: 10 }); console.log(otherInstance.options); *// Shows that maxlength will be 10 for this field*

# Not :

* Parsley relies on [**jQuery**](http://jquery.com/)**(>= 1.8)**, and it will need to be included before including Parsley.
* You will also need to include [**es5-shim**](https://github.com/es-shims/es5-shim) if you want need to support IE8.
* Dil Seçeneklerinde en alttaki dilin mesajları otomatik olarak atanıyor:Alttakinde mesala İtalyanca oluyor. Bunlar da dil seçenekleri : [See the available localizations here](https://github.com/guillaumepotier/Parsley.js/tree/master/src/i18n).

To load a different locale and its messages, include them after Parsley:

<script src="jquery.js"></script>

<script src="parsley.min.js"></script>

<script src="i18n/fr.js"></script>

<script src="i18n/it.js"></script>

#### Checkbox, radio and select multiple

These fields are a bit different than regular input, textarea or simple select. They need to have either a name or anid attribute to be correctly bound and validated by Parsley. Otherwise, they will be ignored and a warning will be put in the console.

* <http://parsleyjs.org/doc/index.html>