There's a newer version of Bootstrap!

| Search docs... | Ctrl + / | v5.1 ▾ | ↕ |

View on GitHub

Validation

Provide valuable, actionable feedback to your users with HTML5 form validation, via browser default behaviors or custom styles and JavaScript.

On this page

How it works
Custom styles
Browser defaults
Server side
Supported elements
Tooltips
Sass
    Variables
    Mixins
    Map
    Loop
    Customizing

We are aware that currently the client-side custom validation styles and tooltips are not accessible, since they are not exposed to assistive technologies. While we work on a solution, we'd recommend either using the server-side option or the default browser validation method.

How it works

Here's how form validation works with Bootstrap:

- HTML form validation is applied via CSS's two pseudo-classes, `:invalid` and `:valid`. It applies to `<input>`, `<select>`, and `<textarea>` elements.
- Bootstrap scopes the `:invalid` and `:valid` styles to parent `.was-validated` class, usually applied to the `<form>`. Otherwise, any required field without a value shows up as invalid on page load. This way, you may choose when to activate them (typically after form submission is attempted).
- To reset the appearance of the form (for instance, in the case of dynamic form submissions using AJAX), remove the `.was-validated` class from the `<form>` again after submission.
- As a fallback, `.is-invalid` and `.is-valid` classes may be used instead of the pseudo-classes for [server-side validation](#). They do not require a `.was-validated` parent class.
- Due to constraints in how CSS works, we cannot (at present) apply styles to a `<label>` that comes before a form control in the DOM without the help of custom JavaScript.
- All modern browsers support the [constraint validation API](#), a series of JavaScript methods for validating form controls.
- Feedback messages may utilize the [browser defaults](#) (different for each browser, and unstylable via CSS) or our custom feedback styles with additional HTML and CSS.
- You may provide custom validity messages with `setCustomValidity` in JavaScript.

With that in mind, consider the following demos for our custom form validation styles, optional server-side classes, and browser defaults.

Custom styles

For custom Bootstrap form validation messages, you'll need to add the `novalidate` boolean attribute to your `<form>`. This disables the browser default feedback tooltips, but still provides access to the form validation APIs in JavaScript. Try to submit the form below; our JavaScript will intercept the submit button and relay feedback to you. When attempting to submit, you'll see the `:invalid` and `:valid` styles applied to your form controls.

Custom feedback styles apply custom colors, borders, focus styles, and background icons to better communicate feedback. Background icons for `<select>`s are only available with `.form-select`, and not `.form-control`.

First name

Mark

Last name

Otto

Username

@

City

State

Choose...

Zip

☐ Agree to terms and conditions

Submit form

```
<form class="row g-3 needs-validation" novalidate>
  <div class="col-md-4">
    <label for="validationCustom01" class="form-label">First name</label>
    <input type="text" class="form-control" id="validationCustom01" value="Mark" req
    <div class="valid-feedback">
      Looks good!
    </div>
  </div>
  <div class="col-md-4">
    <label for="validationCustom02" class="form-label">Last name</label>
    <input type="text" class="form-control" id="validationCustom02" value="Otto" req
    <div class="valid-feedback">
      Looks good!
    </div>
  </div>
  <div class="col-md-4">
    <label for="validationCustomUsername" class="form-label">Username</label>
    <div class="input-group has-validation">
      <span class="input-group-text" id="inputGroupPrepend">@</span>
      <input type="text" class="form-control" id="validationCustomUsername" aria-des
      <div class="invalid-feedback">
        Please choose a username.
      </div>
    </div>
  </div>
```

```html
    </div>
    <div class="col-md-6">
      <label for="validationCustom03" class="form-label">City</label>
      <input type="text" class="form-control" id="validationCustom03" required>
      <div class="invalid-feedback">
        Please provide a valid city.
      </div>
    </div>
    <div class="col-md-3">
      <label for="validationCustom04" class="form-label">State</label>
      <select class="form-select" id="validationCustom04" required>
        <option selected disabled value="">Choose...</option>
        <option>...</option>
      </select>
      <div class="invalid-feedback">
        Please select a valid state.
      </div>
    </div>
    <div class="col-md-3">
      <label for="validationCustom05" class="form-label">Zip</label>
      <input type="text" class="form-control" id="validationCustom05" required>
      <div class="invalid-feedback">
        Please provide a valid zip.
      </div>
    </div>
    <div class="col-12">
      <div class="form-check">
        <input class="form-check-input" type="checkbox" value="" id="invalidCheck" req
        <label class="form-check-label" for="invalidCheck">
          Agree to terms and conditions
        </label>
        <div class="invalid-feedback">
          You must agree before submitting.
        </div>
      </div>
    </div>
    <div class="col-12">
      <button class="btn btn-primary" type="submit">Submit form</button>
    </div>
```

```javascript
// Example starter JavaScript for disabling form submissions if there are invalid fi
(function () {
  'use strict'

  // Fetch all the forms we want to apply custom Bootstrap validation styles to
  var forms = document.querySelectorAll('.needs-validation')
```

```
  // Loop over them and prevent submission
  Array.prototype.slice.call(forms)
    .forEach(function (form) {
      form.addEventListener('submit', function (event) {
        if (!form.checkValidity()) {
          event.preventDefault()
          event.stopPropagation()
        }

        form.classList.add('was-validated')
      }, false)
    })
```

Browser defaults

Not interested in custom validation feedback messages or writing JavaScript to change form ▸ behaviors? All good, you can use the browser defaults. Try submitting the form below. Depending on your browser and OS, you'll see a slightly different style of feedback.

While these feedback styles cannot be styled with CSS, you can still customize the feedback text through JavaScript.

First name

Mark

Last name

Otto

Username

@

City

State

Choose...

Zip

Agree to terms and conditions

Submit form

```html
<form class="row g-3">
  <div class="col-md-4">
    <label for="validationDefault01" class="form-label">First name</label>
    <input type="text" class="form-control" id="validationDefault01" value="Mark" re
  </div>
  <div class="col-md-4">
    <label for="validationDefault02" class="form-label">Last name</label>
    <input type="text" class="form-control" id="validationDefault02" value="Otto" re
  </div>
  <div class="col-md-4">
    <label for="validationDefaultUsername" class="form-label">Username</label>
    <div class="input-group">
      <span class="input-group-text" id="inputGroupPrepend2">@</span>
      <input type="text" class="form-control" id="validationDefaultUsername"  aria-d
    </div>
  </div>
  <div class="col-md-6">
    <label for="validationDefault03" class="form-label">City</label>
    <input type="text" class="form-control" id="validationDefault03" required>
  </div>
  <div class="col-md-3">
    <label for="validationDefault04" class="form-label">State</label>
    <select class="form-select" id="validationDefault04" required>
      <option selected disabled value="">Choose...</option>
      <option>...</option>
    </select>
  </div>
  <div class="col-md-3">
    <label for="validationDefault05" class="form-label">Zip</label>
    <input type="text" class="form-control" id="validationDefault05" required>
  </div>
  <div class="col-12">
    <div class="form-check">
      <input class="form-check-input" type="checkbox" value="" id="invalidCheck2" re
      <label class="form-check-label" for="invalidCheck2">
        Agree to terms and conditions
      </label>
    </div>
  </div>
```

```
<div class="col-12">
  <button class="btn btn-primary" type="submit">Submit form</button>
</div>
```

Server side

We recommend using client-side validation, but in case you require server-side validation, you can indicate invalid and valid form fields with `.is-invalid` and `.is-valid`. Note that `.invalid-feedback` is also supported with these classes.

For invalid fields, ensure that the invalid feedback/error message is associated with the relevant form field using `aria-describedby` (noting that this attribute allows more than one `id` to be referenced, in case the field already points to additional form text).

To fix issues with border radii, input groups require an additional `.has-validation` class.

First name

Mark

Looks good!

Last name

Otto

Looks good!

Username

@	

Please choose a username.

City

Please provide a valid city.

State

Choose...

Please select a valid state.

Zip

Please provide a valid zip.

☐ Agree to terms and conditions

       You must agree before submitting.

Submit form

```html
<form class="row g-3">
  <div class="col-md-4">
    <label for="validationServer01" class="form-label">First name</label>
    <input type="text" class="form-control is-valid" id="validationServer01" value="
    <div class="valid-feedback">
      Looks good!
    </div>
  </div>
  <div class="col-md-4">
    <label for="validationServer02" class="form-label">Last name</label>
    <input type="text" class="form-control is-valid" id="validationServer02" value="
    <div class="valid-feedback">
      Looks good!
    </div>
  </div>
  <div class="col-md-4">
    <label for="validationServerUsername" class="form-label">Username</label>
    <div class="input-group has-validation">
      <span class="input-group-text" id="inputGroupPrepend3">@</span>
      <input type="text" class="form-control is-invalid" id="validationServerUsernam
      <div id="validationServerUsernameFeedback" class="invalid-feedback">
        Please choose a username.
      </div>
    </div>
  </div>
  <div class="col-md-6">
    <label for="validationServer03" class="form-label">City</label>
    <input type="text" class="form-control is-invalid" id="validationServer03" aria-
    <div id="validationServer03Feedback" class="invalid-feedback">
      Please provide a valid city.
    </div>
  </div>
  <div class="col-md-3">
    <label for="validationServer04" class="form-label">State</label>
    <select class="form-select is-invalid" id="validationServer04" aria-describedby=
      <option selected disabled value="">Choose...</option>
      <option>...</option>
    </select>
```

```
      <div id="validationServer04Feedback" class="invalid-feedback">
        Please select a valid state.
      </div>
    </div>
    <div class="col-md-3">
      <label for="validationServer05" class="form-label">Zip</label>
      <input type="text" class="form-control is-invalid" id="validationServer05" aria-
      <div id="validationServer05Feedback" class="invalid-feedback">
        Please provide a valid zip.
      </div>
    </div>
    <div class="col-12">
      <div class="form-check">
        <input class="form-check-input is-invalid" type="checkbox" value="" id="invali
        <label class="form-check-label" for="invalidCheck3">
          Agree to terms and conditions
        </label>
        <div id="invalidCheck3Feedback" class="invalid-feedback">
          You must agree before submitting.
        </div>
      </div>
    </div>
    <div class="col-12">
      <button class="btn btn-primary" type="submit">Submit form</button>
    </div>
```

Supported elements

Validation styles are available for the following form controls and components:

- `<input>`s and `<textarea>`s with `.form-control` (including up to one `.form-control` in input groups)
- `<select>`s with `.form-select`
- `.form-check`s

Textarea

Required example textarea

Please enter a message in the textarea.

☐ Check this checkbox

Example invalid feedback text

○ Toggle this radio
○ Or toggle this other radio
   More example invalid feedback text

┌───┐
│ Open this select menu   │
└───┘
Example invalid select feedback

┌──────────────────┬──────────────────────────────────┐
│ Escolher ficheiro │ Nenhum ficheiro selecionado   │
└──────────────────┴──────────────────────────────────┘
Example invalid form file feedback

┌──────────────────┐
│ Submit form   │
└──────────────────┘

```html
<form class="was-validated">
  <div class="mb-3">
    <label for="validationTextarea" class="form-label">Textarea</label>
    <textarea class="form-control is-invalid" id="validationTextarea" placeholder="R
    <div class="invalid-feedback">
      Please enter a message in the textarea.
    </div>
  </div>

  <div class="form-check mb-3">
    <input type="checkbox" class="form-check-input" id="validationFormCheck1" requir
    <label class="form-check-label" for="validationFormCheck1">Check this checkbox</
    <div class="invalid-feedback">Example invalid feedback text</div>
  </div>

  <div class="form-check">
    <input type="radio" class="form-check-input" id="validationFormCheck2" name="rad
    <label class="form-check-label" for="validationFormCheck2">Toggle this radio</la
  </div>
  <div class="form-check mb-3">
    <input type="radio" class="form-check-input" id="validationFormCheck3" name="rad
    <label class="form-check-label" for="validationFormCheck3">Or toggle this other
    <div class="invalid-feedback">More example invalid feedback text</div>
  </div>

  <div class="mb-3">
    <select class="form-select" required aria-label="select example">
      <option value="">Open this select menu</option>
      <option value="1">One</option>
      <option value="2">Two</option>
      <option value="3">Three</option>
    </select>
    <div class="invalid-feedback">Example invalid select feedback</div>
```

```
    </div>

    <div class="mb-3">
      <input type="file" class="form-control" aria-label="file example" required>
      <div class="invalid-feedback">Example invalid form file feedback</div>
    </div>

    <div class="mb-3">
      <button class="btn btn-primary" type="submit" disabled>Submit form</button>
    </div>
```

Tooltips

If your form layout allows it, you can swap the `.{valid|invalid}-feedback` classes for `.{valid|invalid}-tooltip` classes to display validation feedback in a styled tooltip. Be sure to have a parent with `position: relative` on it for tooltip positioning. In the example below, our column classes have this already, but your project may require an alternative setup.

First name

Mark

Last name

Otto

Username

@

City

State

Choose...

Zip

Submit form

```html
<form class="row g-3 needs-validation" novalidate>
  <div class="col-md-4 position-relative">
    <label for="validationTooltip01" class="form-label">First name</label>
    <input type="text" class="form-control" id="validationTooltip01" value="Mark" re
    <div class="valid-tooltip">
      Looks good!
    </div>
  </div>
  <div class="col-md-4 position-relative">
    <label for="validationTooltip02" class="form-label">Last name</label>
    <input type="text" class="form-control" id="validationTooltip02" value="Otto" re
    <div class="valid-tooltip">
      Looks good!
    </div>
  </div>
  <div class="col-md-4 position-relative">
    <label for="validationTooltipUsername" class="form-label">Username</label>
    <div class="input-group has-validation">
      <span class="input-group-text" id="validationTooltipUsernamePrepend">@</span>
      <input type="text" class="form-control" id="validationTooltipUsername" aria-de
      <div class="invalid-tooltip">
        Please choose a unique and valid username.
      </div>
    </div>
  </div>
  <div class="col-md-6 position-relative">
    <label for="validationTooltip03" class="form-label">City</label>
    <input type="text" class="form-control" id="validationTooltip03" required>
    <div class="invalid-tooltip">
      Please provide a valid city.
    </div>
  </div>
  <div class="col-md-3 position-relative">
    <label for="validationTooltip04" class="form-label">State</label>
    <select class="form-select" id="validationTooltip04" required>
      <option selected disabled value="">Choose...</option>
      <option>...</option>
    </select>
    <div class="invalid-tooltip">
      Please select a valid state.
    </div>
  </div>
  <div class="col-md-3 position-relative">
    <label for="validationTooltip05" class="form-label">Zip</label>
    <input type="text" class="form-control" id="validationTooltip05" required>
    <div class="invalid-tooltip">
      Please provide a valid zip.
```

```html
      </div>
    </div>
    <div class="col-12">
      <button class="btn btn-primary" type="submit">Submit form</button>
    </div>
```

Sass

Variables

```scss
$form-feedback-margin-top:            $form-text-margin-top;
$form-feedback-font-size:             $form-text-font-size;
$form-feedback-font-style:            $form-text-font-style;
$form-feedback-valid-color:           $success;
$form-feedback-invalid-color:         $danger;

$form-feedback-icon-valid-color:      $form-feedback-valid-color;
$form-feedback-icon-valid:            url("data:image/svg+xml,<svg xmlns='http://www.w
$form-feedback-icon-invalid-color:    $form-feedback-invalid-color;
$form-feedback-icon-invalid:          url("data:image/svg+xml,<svg xmlns='http://www.w
```

Mixins

Two mixins are combined together, through our [loop](#), to generate our form validation feedback styles.

```scss
@mixin form-validation-state-selector($state) {
  @if ($state == "valid" or $state == "invalid") {
    .was-validated #{if(&, "&", "")}:#{$state},
    #{if(&, "&", "")}.is-#{$state} {
      @content;
    }
  } @else {
    #{if(&, "&", "")}.is-#{$state} {
      @content;
    }
  }
}

@mixin form-validation-state(
```

```scss
  $state,
  $color,
  $icon,
  $tooltip-color: color-contrast($color),
  $tooltip-bg-color: rgba($color, $form-feedback-tooltip-opacity),
  $focus-box-shadow: 0 0 $input-btn-focus-blur $input-focus-width rgba($color, $inpu
) {
  .#{$state}-feedback {
    display: none;
    width: 100%;
    margin-top: $form-feedback-margin-top;
    @include font-size($form-feedback-font-size);
    font-style: $form-feedback-font-style;
    color: $color;
  }

  .#{$state}-tooltip {
    position: absolute;
    top: 100%;
    z-index: 5;
    display: none;
    max-width: 100%; // Contain to parent when possible
    padding: $form-feedback-tooltip-padding-y $form-feedback-tooltip-padding-x;
    margin-top: .1rem;
    @include font-size($form-feedback-tooltip-font-size);
    line-height: $form-feedback-tooltip-line-height;
    color: $tooltip-color;
    background-color: $tooltip-bg-color;
    @include border-radius($form-feedback-tooltip-border-radius);
  }

  @include form-validation-state-selector($state) {
    ~ .#{$state}-feedback,
    ~ .#{$state}-tooltip {
      display: block;
    }
  }

  .form-control {
    @include form-validation-state-selector($state) {
      border-color: $color;

      @if $enable-validation-icons {
        padding-right: $input-height-inner;
        background-image: escape-svg($icon);
        background-repeat: no-repeat;
        background-position: right $input-height-inner-quarter center;
        background-size: $input-height-inner-half $input-height-inner-half;
      }
```

```scss
    &:focus {
      border-color: $color;
      box-shadow: $focus-box-shadow;
    }
  }
}


// stylelint-disable-next-line selector-no-qualifying-type
textarea.form-control {
  @include form-validation-state-selector($state) {
    @if $enable-validation-icons {
      padding-right: $input-height-inner;
      background-position: top $input-height-inner-quarter right $input-height-inn
    }
  }
}


.form-select {
  @include form-validation-state-selector($state) {
    border-color: $color;

    @if $enable-validation-icons {
      &:not([multiple]):not([size]),
      &:not([multiple])[size="1"] {
        padding-right: $form-select-feedback-icon-padding-end;
        background-image: escape-svg($form-select-indicator), escape-svg($icon);
        background-position: $form-select-bg-position, $form-select-feedback-icon-
        background-size: $form-select-bg-size, $form-select-feedback-icon-size;
      }
    }

    &:focus {
      border-color: $color;
      box-shadow: $focus-box-shadow;
    }
  }
}


.form-check-input {
  @include form-validation-state-selector($state) {
    border-color: $color;

    &:checked {
      background-color: $color;
    }

    &:focus {
      box-shadow: $focus-box-shadow;
```

```scss
      }

      ~ .form-check-label {
        color: $color;
      }
    }
  }
  .form-check-inline .form-check-input {
    ~ .#{$state}-feedback {
      margin-left: .5em;
    }
  }

  .input-group .form-control,
  .input-group .form-select {
    @include form-validation-state-selector($state) {
      @if $state == "valid" {
        z-index: 1;
      } @else if $state == "invalid" {
        z-index: 2;
      }
      &:focus {
        z-index: 3;
      }
    }
  }
}
```

Map

This is the validation Sass map from `_variables.scss`. Override or extend this to generate different or additional states.

```scss
$form-validation-states: (
  "valid": (
    "color": $form-feedback-valid-color,
    "icon": $form-feedback-icon-valid
  ),
  "invalid": (
    "color": $form-feedback-invalid-color,
    "icon": $form-feedback-icon-invalid
  )
);
```

Maps of `$form-validation-states` can contain three optional parameters to override tooltips and focus styles.
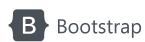
Loop

Used to iterate over `$form-validation-states` map values to generate our validation styles. Any modifications to the above Sass map will be reflected in your compiled CSS via this loop.

```
@each $state, $data in $form-validation-states {
  @include form-validation-state($state, $data...);
}
```

Customizing

Validation states can be customized via Sass with the `$form-validation-states` map. Located in our `_variables.scss` file, this Sass map is how we generate the default `valid`/`invalid` validation states. Included is a nested map for customizing each state's color, icon, tooltip color, and focus shadow. While no other states are supported by browsers, those using custom styles can easily add more complex form feedback.

Please note that **we do not recommend customizing `$form-validation-states` values without also modifying the `form-validation-state` mixin**.

B Bootstrap

Designed and built with all the love in the world by the Bootstrap team with the help of our contributors.

Code licensed MIT, docs CC BY 3.0.

Currently v5.1.3.

Analytics by Fathom.

Links

Home

Docs

Examples

Guides

Getting started

Starter template

Webpack

Themes   Parcel

Blog

Swag Store

Projects   ## Community

Bootstrap 5   Issues

Bootstrap 4   Discussions

Icons   Corporate sponsors

RFS   Open Collective

npm starter   Stack Overflow