Bootstrap Forms

In bootstrap, forms are useful to apply some default styles for form controls such as input elements, labels, textareas, select elements, etc. with a predefined set of classes.

By using bootstrap predefined .form-control class, we can style textual form controls such as <input>, <select> and <textarea> elements. Same way, we need to use other predefined classes such as .form-check-input, .form-control-file, etc. to style checkboxes, file inputs, etc. based on our requirements.

The bootstrap has provided three types of form layouts such as default (vertical), horizontal and inline forms. Now, we will learn how we can create different form layouts with different input controls.

Bootstrap Create Form with Controls

In bootstrap, we can create the form layout by wrapping all form controls inside of <form> element and we need to wrap each form control with a .form-group class to ensure proper margins.

Following is the example of creating the default (vertical) form layout with required form controls in bootstrap.

[Live Preview](https://www.tutlane.com/example/bootstrap/bootstrap-forms-example)

<form>

    <div class="form-group">

        <label for="email">Email</label>

        <input type="email" class="form-control" id="email1" placeholder="Enter email">

    </div>

    <div class="form-group">

        <label for="password">Password</label>

        <input type="password" class="form-control" id="Password1" placeholder="Enter password">

    </div>

    <div class="form-group">

        <label for="fileupload">Profile Pic</label>

        <input type="file" class="form-control-file" id="File1">

    </div>

    <div class="form-group form-check">

        <input type="checkbox" class="form-check-input" id="Checkbox1">

        <label class="form-check-label" for="NewlettersCheck">Subscribe to newsletters</label>

    </div>

    <button type="submit" class="btn btn-primary">Submit</button>

</form>

The above example will return the result like as shown below.

If you observe the above result, by default all the form controls (labels, input controls) aligned vertically. In case, if you want to align the elements horizontally (side by side) you need to create horizontal form layouts.

Bootstrap Horizontal Forms

In bootstrap, we can create the horizontal form layouts by using [grid classes](https://www.tutlane.com/tutorial/bootstrap/bootstrap-grid-system) to align the labels and form controls side by side.

To create horizontal forms, you need to add .row class to form groups and add .col-\*-\* classes to your labels and controls to specify the width. Also, be sure to add .col-form-label to your <label> elements to align vertically centered to their associated form controls.

[Live Preview](https://www.tutlane.com/example/bootstrap/bootstrap-horizontal-forms-example)

<form>

    <div class="form-group row">

        <label for="email" class="col-sm-2 col-form-label">Email</label>

        <div class="col-sm-10">

            <input type="email" class="form-control" id="email1" placeholder="Enter email">

        </div>

    </div>

    <div class="form-group row">

        <label for="password" class="col-sm-2 col-form-label">Password</label>

        <div class="col-sm-10">

            <input type="password" class="form-control" id="Password1" placeholder="Enter password">

        </div>

    </div>

    <div class="form-group row">

        <label for="gender" class="col-sm-2 col-form-label">Gender</label>

        <div class="col-sm-10">

            <div class="form-check">

                <input class="form-check-input" type="radio" name="genderadios" id="Radio1" value="male" checked>

                <label class="form-check-label" for="gridRadios1">Male</label>

            </div>

            <div class="form-check">

                <input type="radio" class="form-check-input" name="genderadios" id="Radio2" value="female">

                <label class="form-check-label" for="gridRadios2">Female</label>

            </div>

        </div>

    </div>

    <div class="form-group row">

        <label for="fileupload" class="col-sm-2 col-form-label">Profile Pic</label>

        <div class="col-sm-10">

            <input type="file" class="form-control-file" id="File1">

        </div>

    </div>

    <div class="form-group row">

        <div class="col-sm-10 offset-sm-2">

            <div class="form-check">

                <input type="checkbox" class="form-check-input" id="Checkbox1">

                <label class="form-check-label" for="NewlettersCheck">Subscribe to newsletters</label>

            </div>

        </div>

    </div>

    <div class="form-group row">

        <div class="col-sm-10 offset-sm-2">

            <button type="submit" class="btn btn-primary">Submit</button>

        </div>

    </div>

</form>

The above example will return the result like as shown below.

If you observe the above result, all the label values are aligned vertically centered with their associated form controls.

Bootstrap Inline Forms

In bootstrap, if you want to show all the form elements such as labels, input controls, etc. on a single horizontal row, then you need to use inline forms.

Generally, the inline forms will display all the labels and form controls on the same line only when viewports that are at least 576px wide otherwise the form controls will appear horizontally.

In bootstrap, we can create the inline forms by adding .form-inline class to <form> element. Following is the example of creating the inline form with required form controls in bootstrap.

[Live Preview](https://www.tutlane.com/example/bootstrap/bootstrap-inline-forms-example)

<form class="form-inline">

    <label for="email">Email: </label>

    <input type="email" class="form-control" id="email1" placeholder="Enter email">

    <label for="password">Password: </label>

    <input type="password" class="form-control" id="Password1" placeholder="Enter password">

    <div class="form-check">

        <input type="checkbox" class="form-check-input" id="Checkbox1">

        <label class="form-check-label" for="inlineCheck">Remember me</label>

    </div>

    <button type="submit" class="btn btn-primary">Submit</button>

</form>

The above example will return the result like as shown below.

Bootstrap inline forms example result

If you observe the above result, the inline form has arranged the controls without having any spaces and it looks like compressed form. In inline form, we need to use bootstrap spacing utility classes to align the controls properly.

Following is the example of using the bootstrap spacing utility classes to align the inline form controls properly.

[Live Preview](https://www.tutlane.com/example/bootstrap/bootstrap-inline-forms-with-utilities-example)

<form class="form-inline">

    <label for="email" class="mr-sm-2">Email:</label>

    <input type="email" class="form-control mb-2 mr-sm-2" id="email1" placeholder="Enter email">

    <label for="password" class="mr-sm-2">Password:</label>

    <input type="password" class="form-control mb-2 mr-sm-2" id="Password1" placeholder="Enter password">

    <div class="form-check mb-2 mr-sm-2">

        <input type="checkbox" class="form-check-input" id="Checkbox1">

        <label class="form-check-label" for="inlineCheck">Remember me</label>

    </div>

    <button type="submit" class="btn btn-primary mb-2">Submit</button>

</form>

In above example, we used right margin (.mr-sm-2) and bottom margin (.mb-2) classes to align inline form controls properly on all devices and it will return the result like as shown below.

Bootstrap inline forms with utilities example result

If you observe the above result, the inline form controls aligned properly after adding spacing utility classes to input controls.

Bootstrap Form Validation

The bootstrap is having a built-in form validation classes to provide the actionable feedback to users. Generally, the bootstrap will use browser form validation APIs to validate the form.

To implement form validations, you need to add either .was-validated or .needs-validation class to <form> element to provide the validation feedback before or after submitting the form and you need to add the required attribute to the form input controls to enable validation.

The .was-validated class is useful to provide the validation feedback before submitting the form and the .needs-validation class will provide the validation feedback after submitting the form. So, based on your requirement you need to add either .was-validated or .needs-validation class to your <form> element.

The bootstrap will apply green (valid) or red (invalid) color borders to the input controls in the form to differentiate between the valid and invalid fields. You can also add .valid-feedback or .invalid-feedback messages to the form to inform the user what is missing or what needs to be done before submitting the form.

Following is the example of implementing the form controls validation using a .was-validated class to provide the validation feedback before submitting the form in bootstrap.

[Live Preview](https://www.tutlane.com/example/bootstrap/bootstrap-form-validations-example)

<form class="was-validated">

    <div class="form-group">

        <label for="fname">First Name</label>

        <input type="text" class="form-control" id="Text1" name="fname" placeholder="Enter first name" required>

        <div class="valid-feedback">Valid.</div>

        <div class="invalid-feedback">Please enter first name.</div>

    </div>

    <div class="form-group">

        <label for="lname">Last Name</label>

        <input type="text" class="form-control" id="Text2" placeholder="Enter last name" required>

        <div class="valid-feedback">Valid.</div>

        <div class="invalid-feedback">Please enter last name.</div>

    </div>

    <div class="form-group">

        <label for="email">Email</label>

        <input type="email" class="form-control" id="email1" placeholder="Enter email" required>

        <div class="valid-feedback">Valid.</div>

        <div class="invalid-feedback">Please enter email.</div>

    </div>

    <div class="form-group form-check">

        <input type="checkbox" class="form-check-input" id="Checkbox1" required> I agree all terms & conditions

        <div class="valid-feedback">Valid.</div>

        <div class="invalid-feedback">Please agree terms & conditions.</div>

    </div>

    <button type="submit" class="btn btn-primary">Submit</button>

</form>

If you observe the above example, we added .was-validated class to <form> element to show the validation feedback before submitting the form and added required attribute to the input controls to enable validation.

The above example will return the result like as shown below.

Now, we will learn how to provide validation feedback after submitting the form. Following is the example of implementing the form controls validation using .needs-validation class.

[Live Preview](https://www.tutlane.com/example/bootstrap/bootstrap-form-validations-after-submit-example)

<form class="needs-validation" novalidate>

    <div class="form-group">

        <label for="fname">First Name</label>

        <input type="text" class="form-control" id="Text1" name="fname" placeholder="Enter first name" required>

        <div class="valid-feedback">Valid.</div>

        <div class="invalid-feedback">Please enter first name.</div>

    </div>

    <div class="form-group">

        <label for="lname">Last Name</label>

        <input type="text" class="form-control" id="Text2" placeholder="Enter last name" required>

        <div class="valid-feedback">Valid.</div>

        <div class="invalid-feedback">Please enter last name.</div>

    </div>

    <div class="form-group">

        <label for="email">Email</label>

        <input type="email" class="form-control" id="email1" placeholder="Enter email" required>

        <div class="valid-feedback">Valid.</div>

        <div class="invalid-feedback">Please enter email.</div>

    </div>

    <div class="form-group form-check">

        <input type="checkbox" class="form-check-input" id="Checkbox1" required> I agree all terms & conditions

        <div class="valid-feedback">Valid.</div>

        <div class="invalid-feedback">Please agree terms & conditions.</div>

    </div>

    <button type="submit" class="btn btn-primary">Submit</button>

</form>

<script>

    // JavaScript to disable form submissions if there are invalid fields

    (function() {

        'use strict';

        window.addEventListener('load', function() {

            // Get the forms we want to apply validation styles

            var forms = document.getElementsByClassName('needs-validation');

            // Loop over them and prevent submission

            var validation = Array.prototype.filter.call(forms, function(form) {

                form.addEventListener('submit', function(event) {

                    if (form.checkValidity() === false) {

                        event.preventDefault();

                        event.stopPropagation();

                    }

                    form.classList.add('was-validated');

                }, false);

            });

        }, false);

    })();

</script>

If you observe the above example, we added novalidate attribute along with .needs-validation class to the <form> element to disable the browser default feedback tooltips and we used [JavaScript](https://www.tutlane.com/tutorial/javascript) code to disable the form submission in case if there are invalid fields.

The above example will return the result like as shown below.

If you observe the above result, the validation feedback messages are appearing only when we submit the form.

This is how we can use the forms and form controls to apply bootstrap styles based on our requirements.