



Introduction to HTML Forms

HTML forms are a fundamental part of web development, enabling users to interact with web applications. They allow users to input and submit data, which is then processed and displayed.



by amarjeet saini

Last edited less than a minute ago

Form Structure

Every HTML form starts with the `<form>`

tag. The `action` attribute specifies the URL where the form data will be submitted. The `method` attribute defines the HTTP method used for submission, typically "GET" or "POST".

Form Tag

```
<form action="submit.php" method="POST">
  ... form elements ...
</form>
```

Input Fields

Form elements are used to collect user input. These elements are typically enclosed within the `<form>` tag.

Submission

A submit button is used to send the form data to the server for processing. It is typically placed at the end of the form.

Form Elements

HTML forms utilize a variety of elements to gather user input. Each element serves a specific purpose and provides a distinct way for users to interact with the form.

1

Text Input

The `<input type="text">` element is used for single-line text input. It is commonly used

for collecting names, email addresses, and other short text strings.

2 Password Input

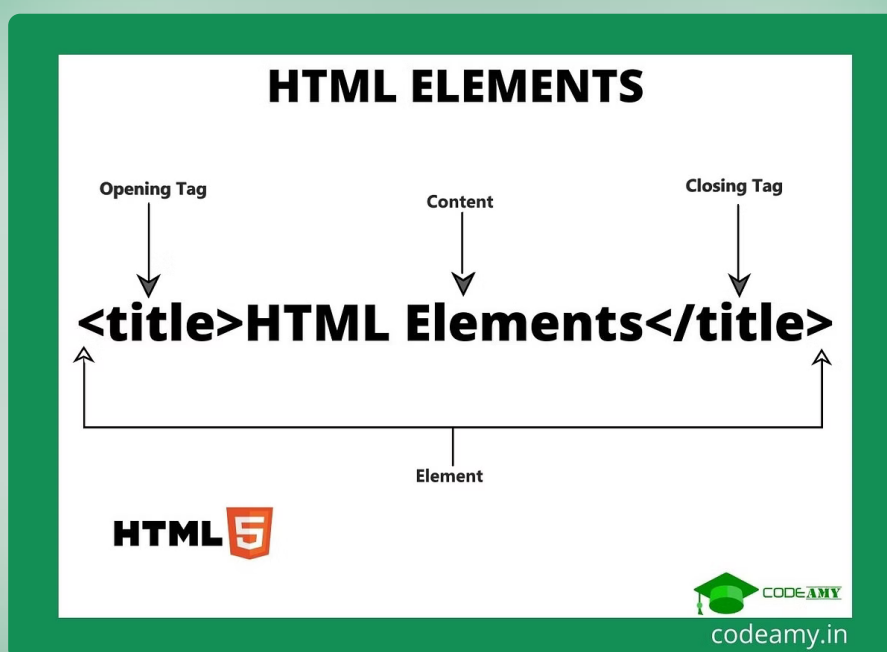
The `<password>` element is used to collect passwords. The text is masked for security purposes, preventing unauthorized viewing.

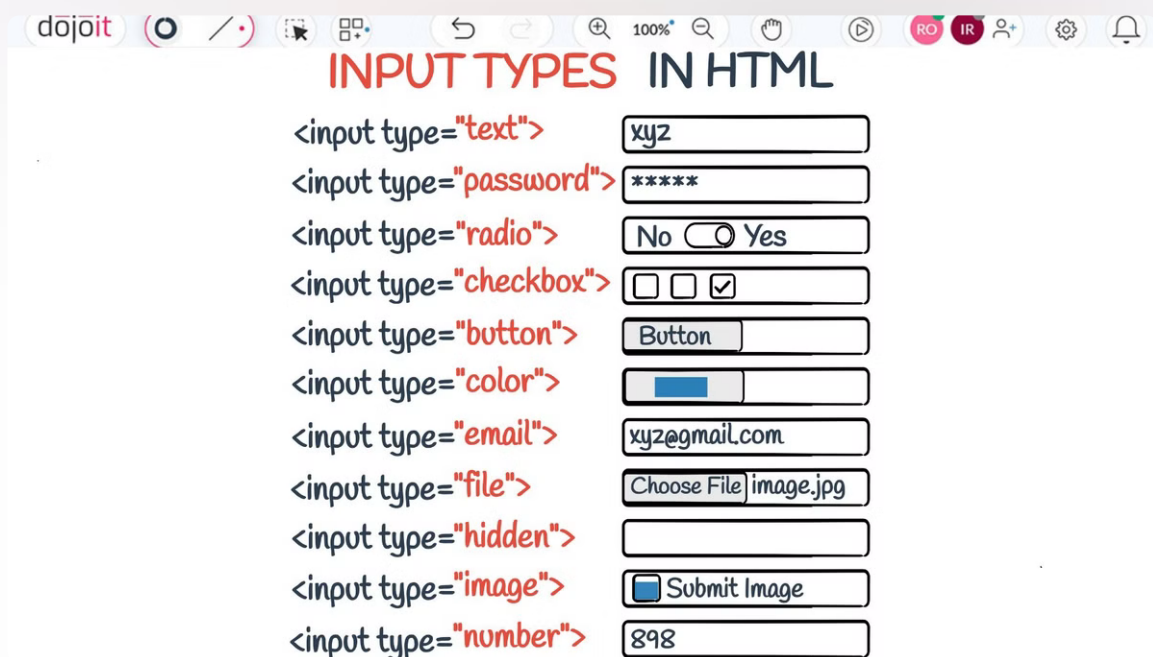
3 Radio Buttons

The `<radio>` element allows users to select only one option from a set of choices. It is suitable for selecting a single value from a predefined list.

4 Checkboxes

The `<checkbox>` element allows users to select multiple options from a set of choices. It is commonly used for selecting multiple items or toggling options.





Input Types

HTML forms offer various input types that cater to diverse data collection needs. Each type dictates how users interact with the form element, providing a tailored experience.

Text	Password	Email	Number	Tel	Url
Single-line text	Masked text for password input	Email address validation	Numeric input	Phone number	URL validation

Validation

Form validation ensures the accuracy and integrity of user input before submission. It helps prevent errors and ensures that the submitted data meets specific criteria.

1

Client-Side Validation

This validation occurs in the user's browser. JavaScript is commonly used to validate form data before submitting it to the server.

2

Server-Side Validation

This validation happens on the server, after the form data has been submitted. It provides an additional layer of security and ensures data integrity.

Your project was not created

CREATE A PROJECT

Name *

The name field is required.

Description*

The description field is required.

Create project

This demo reuses back end validation on the front end.
See [Github](#) for details.

Submission and Processing

SUBMISSION AND PROCESSING

When a user submits a form, the data is sent to the server for processing. The server-side script receives the data, performs necessary actions, and responds accordingly.

1

Form Submission

The user clicks the submit button, initiating the form submission process.

2

Server-Side Processing

The server receives the form data and processes it according to the defined logic. This may involve database updates, email notifications, or other actions.

3

Response

The server sends a response back to the user, typically displaying a confirmation message or redirecting to a new page.

Creating Accessible Forms

Error: First Name (required) is not blank. — Error Message at the top of the page

APPLICATION — What the form is for

Tell us about you and your funding needs with the following 6 fields. — How many fields it contains

The form will take approximately 2 minutes to complete. — Approximately how long it will take to fill out

Personal information will not be shared and will remain on file for up to 90 days. — How any personal information collected will be handled

First Name (required) — Use "First Name" instead of merely "Name"

Last Name (required) — Use the word "required" instead of just red-colored text or an asterisk

Date (required)
Month / Day / Year — Clarify in the label description how an error should be formatted

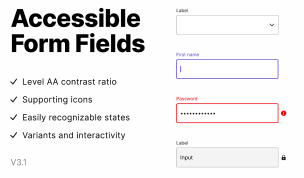
Email (required)
name@url.com

Which topic interests you?
 Science
 Programs for Children
 New Events — Place radio button label or checkbox to the right of the input

Choose from the following (required):
Select — "Select" should be provided, and the "choose" option for readability for verification

Submit Your Entry Now — Encourage submission through a button labeled with a clear CTA

Submittable



Accessibility

Accessible forms ensure that users with disabilities can easily interact with and submit forms. Accessibility is crucial for inclusivity and a positive user experience.



Keyboard Navigation

Users should be able to navigate through all form elements using only the keyboard.



Screen Reader Compatibility

Forms should be compatible with screen readers, ensuring that users with visual impairments can access and understand the form elements.



Color Contrast

Ensure adequate color contrast between text and background to enhance readability for users with visual impairments.



Clear Labels

All form elements should have clear and concise labels to identify their purpose and provide context to users.



Best Practices

Following best practices ensures that forms are user-friendly, efficient, and secure. These practices contribute to a positive user experience.

Clear Instructions

Provide clear and concise instructions for each form element, guiding users through the input process.

Error Handling

Implement robust error handling to inform users of invalid or missing data, allowing them to correct errors easily.

Data Validation

Validate user input to prevent errors and ensure data integrity. This can be done both client-side and server-side.

Security

Protect form data from unauthorized access and ensure that sensitive information is handled securely.

