HTML form cheat sheet

The tags, attributes and checks behind a working, accessible HTML form. No JavaScript needed.

The skeleton

```
<form action="https://example.com/your-form-handler" method="post">
  <label for="email">Email address</label>
  <input id="email" name="email" type="email" autocomplete="email" required>

  <fieldset>
    <legend>How should we reply?</legend>
    <input id="reply-email" name="reply_by" type="radio" value="email" checked>
    <label for="reply-email">By email</label>
    <input id="reply-phone" name="reply_by" type="radio" value="phone">
    <label for="reply-phone">By phone call</label>
  </fieldset>

  <button type="submit">Send</button>
</form>
```

Input types and when to use them

type or element	Use it for	Tip
`text`	Names, short answers	Add autocomplete="name" (or given-name) for names.
`email`	Email addresses	The browser checks the format before it sends.
`tel`	Phone numbers	No format check. Add autocomplete="tel".
`url`	Web addresses	Needs a full address, like https://example.com.
`number`	Counts and amounts	Set min, max and step.
`date`	Dates	Most browsers show a date picker.
`checkbox`	Yes or no, or pick any	Give every box in a group the same name.
`radio`	Pick one	Same name for the group. Wrap it in fieldset and legend.
`select`	Pick one from a long list	For a required list, make the first option value="".
`textarea`	Long answers	Set maxlength so nobody pastes a novel.
`file`	Uploads	Needs method="post" and enctype="multipart/form-data".
`hidden`	Values people never see	Where the visitor came from, or the time check.

Attributes that matter

- action: the address that receives the answers. method: post for anything you store.
- name: the key each answer arrives under. A field with no name sends nothing.
- id plus label for: the same text in both, so the label belongs to the field.
- required: the browser stops the send until the field has a value.
- autocomplete: lets the browser fill in names, emails and phone numbers.
- maxlength, min, max: limits the browser checks before it sends.

Where the data goes

- The HTML only draws the form. A program on a server receives each response.
- That program checks every answer again (browser checks are easy to skip), then emails or stores it.
- No server? Build the form in a form builder and paste its embed code into the page instead.

Accessible by default (W3C WAI Forms Tutorial)

- Label every field with a label element. A placeholder is not a label.
- Group radio buttons and checkboxes in a fieldset, with a legend as the question.
- Put instructions before the form, and say "required" or "optional" in the label text.
- Keep a visible focus outline so keyboard users can see where they are.
- Ask only what you need. Short forms are easier for everyone.

Stop spam without a puzzle

- Honeypot: a field placed off screen. People leave it empty; bots fill it in. Drop those responses.
- Time check: the server writes the time it sent the page into a hidden field and drops responses that come back in under about three seconds.
- Rate limit: the server refuses a flood of responses from one address.

Three starter forms (contact, sign-up, feedback) in the snippet pack at hornetforms.com/blog/html-form-generator