

ACCESSIBLE FORMS

Some text here with **highlight** and `code`

Screen readers and forms

Screen readers generally read and interact with web pages in **two main ways**.

1. **“Read” mode**
2. **“Forms” mode**

Read mode allows users to read and navigate the page but can not enter data into a form.

Forms mode allows the user to interact with form controls (fill in form fields etc).

In forms mode, keyboard access is restricted to page elements that can **accept focus**.

When screen readers are in forms mode, elements that cannot receive focus **are not announced to the user.**

For example, **instructions that are presented as a paragraph** of text will not be announced to the screen reader user, unless this information has been explicitly associated with a form control.

```
<!-- Hint text will not be announced to screen  
reader users when in forms node -->
```

```
<label for="one">Name</label>
```

```
<input id="one" type="text">
```

```
<p>Hint text</p>
```

Switching modes

Most modern screen readers
automatically switch modes when the
encounter a form control and when
leaving a form.

Screen readers provide an **audible sound** when changing from “Read” mode to “Forms” mode and back again.

What are form
controls?

Form controls are the elements within forms that allow user-interaction.

Form controls include **HTML 4.01 elements** such as `<button>`, `<input>`, `<select>` and `<textarea>`.

They can also include **newer HTML5 elements** such as `<datalist>`, `<output>`, `<progress>` and `<meter>`.

Most form controls are “**replaced elements**” - which means that their appearance and dimensions are initially defined outside the HTML document.

The `<form>` element

A `<form>` is a section of a document that is **used to pass data to the server**. A `<form>` can contain normal content and form controls.

The `<form>` **element** requires start and end tags. A wide range of attributes are allowed inside the start tag.

```
<form>
```

...

```
</form>
```

In HTML 4.01 and XHTML 1.0, the `action` attribute is the **only required attribute** for forms.

However, if the `action` attribute is present, it **must not be blank**.

```
<form action="/results">
```

```
...
```

```
</form>
```

In HTML 4.01 and XHTML 1.0, all form controls **had to be placed inside block level elements** such as `<fieldset>`, `<div>` or `<p>` elements.

```
<form action="#">  
  <p>  
    <label for="one">One</label>  
    <input id="one" type="text">  
  </p>  
</form>
```

```
<form action="#">
  <div>
    <label for="one">One</label>
    <input id="one" type="text">
  </div>
</form>
```

```
<form action="#">  
  <fieldset>  
    <legend>Legend content</legend>  
    <label for="one">One</label>  
    <input id="one" type="text">  
  </fieldset>  
</form>
```

In HTML 5, form controls can be placed inside `<form>` elements **without having to be inside block level elements.**

```
<form action="#">  
  <label for="one">One</label>  
  <input id="one" type="text">  
</form>
```


The <fieldset> element

The `<fieldset>` element is used to **group related form controls** and their associated labels within a web form.

```
<form action="#">  
  <fieldset>  
    ...  
  </fieldset>  
</form>
```

The `<fieldset>` **must contain only one** `<legend>` element. This `<legend>` must come directly after the `<fieldset>` start tag.

```
<form action="#">  
  <fieldset>  
    <legend>Contact details</legend>  
    ...  
  </fieldset>  
</form>
```

The `<fieldset>` element can contain **nested** `<fieldset>` elements.

```
<form action="#">
  <fieldset>
    <legend>Contact details</legend>
    ...
    <fieldset>
      <legend>Another legend</legend>
      ...
    </fieldset>
  </fieldset>
</form>
```

Fieldsets for radio buttons

The `<fieldset>` element can be used to
house groups of radio buttons.

Instructions for the radio buttons can be provided by the `<legend>` element.

Choose a payment method

☒ Weekly

☐ Monthly

☐ Yearly

```
<fieldset>
  <legend>Choose a payment method</legend>
  <input id="weekly" type="radio" name="pay">
  <label for="weekly">Weekly</label>

  <input id="monthly" type="radio" name="pay">
  <label for="monthly">Monthly</label>

  <input id="yearly" type="radio" name="pay">
  <label for="yearly">Yearly</label>
</fieldset>
```

Fieldsets and date formats

The `<fieldset>` element can also be used to wrap around the different `<input>` elements associated with **defining a date.**

Choose a date:

Day

Month

Year

```
<fieldset>
  <legend>Choose a date:</legend>
  <label for="day">Day</label>
  <input id="day" type="text">

  <label for="month">Month</label>
  <input id="month" type="text">

  <label for="year">Year</label>
  <input id="year" type="text">
</fieldset>
```


The <legend> element

The `<legend>` element **represents a caption** for the content of its parent `<fieldset>`.

```
<form action="#">  
  <fieldset>  
    <legend>Contact details</legend>  
    ...  
  </fieldset>  
</form>
```

There can be **only one** `<legend>` inside any `<fieldset>` and it must come directly after the `<fieldset>` start tag.

In HTML5, **block level elements** are now allowed to be placed inside the `<legend>` element.

```
<form action="#">
  <fieldset>
    <legend>
      <h2>Contact details</h2>
    </legend>
    ...
  </fieldset>
</form>
```

The <label> element

The `<label>` element is used to **define labels for form controls.**


```
<label for="phone">Phone</label>  
<input id="phone" type="text">
```

```
<label for="number">Choose a number</label>  
<select id="number">  
  <option>One</option>  
  <option>Two</option>  
  <option>Three</option>  
</select>
```

```
<label for="comment">Add a comment</label>  
<textarea id="comment"></textarea>
```

```
<input for="subscribe" type="radio">  
<label id="subscribe">Subscribe</label>
```

```
<input for="choose" type="checkbox">  
<label id="choose">Yes</label>
```

The “for” attribute

The **for** attribute is used to **explicitly**
associate a **<label>** with the relevant **ID**
of a form control.

```
<label for="phone">Phone</label>  
<input type="text" id="phone" name="phone">
```


When is the `<label>` not required?

Some form controls **do not require** a `<label>` element to describe their purpose.

The `<button>` element uses the **content of the button** as a label.

```
<button>Button text</button>
```

The `<input type="hidden">` requires no label as **it is not displayed to the user.**

```
<input type="hidden">
```

The `<input type="submit">`, `<input type="reset">` and `<input type="button">` are buttons and **use the value as a label.**

```
<input type="button" value="Cancel">  
<input type="submit" value="Submit">  
<input type="reset" value="Reset">
```


However, all other form controls **require some sort of description**, and the `<label>` element is the preferred solution.

Before or after?

In all cases except radio buttons and checkboxes, the contents of the `<label>` must come **before the form control**.

```
<label for="phone">Phone</label>  
<input id="phone" type="text">
```

```
<label for="number">Choose a number</label>  
<select id="number">  
  <option>One</option>  
  <option>Two</option>  
  <option>Three</option>  
</select>
```

```
<label for="comment">Add a comment</label>  
<textarea id="comment"></textarea>
```

For radio buttons and checkboxes, the contents of the `<label>` must come **after the form control.**

```
<input id="subscribe" type="radio">  
<label for="subscribe">Subscribe</label>
```



```
<input id="choose" type="checkbox">  
<label for="choose">Yes</label>
```

Wrapping the `<label>`

It is acceptable to wrap the `<label>`
**around the label content and the form
control.**

However, the `for` and `id` attributes still need to be present in order for the `<label>` and form control **to be explicitly associated.**

And, the same rules about label content being placed before or after **still apply**.

In all cases except radio buttons and checkboxes, the contents of the `<label>` must come **before the form control**.

```
<label for="phone">
```

```
  Phone
```

```
  <input id="phone" type="text">
```

```
</label>
```

```
<label for="number">  
  Choose a number  
  <select id="number">  
    <option>One</option>  
    <option>Two</option>  
    <option>Three</option>  
  </select>  
</label>
```



```
<label for="comment">
```

Add a comment

```
<textarea id="comment"></textarea>
```

```
</label>
```

For radio buttons and checkboxes, the contents of the `<label>` must come **after the form control.**

```
<label id="subscribe">  
  <input for="subscribe" type="radio">  
  Subscribe  
</label>
```

```
<label id="choose">  
  <input for="choose" type="checkbox">  
    Yes  
</label>
```

Using the placeholder
attribute

Some developers use the `placeholder` attribute **to provide a label** for form controls.

```
<input type="text" placeholder="Add your name">
```

However, this method is **not considered an acceptable alternative** as it is not supported by older assistive technologies.

Using the aria-label

In some situations, elements can be given the `aria-label` attribute **to provide an accessible name** for situations when there is no visible label due to a chosen design approach.

However, this method can only be used if the context and visual appearance of the control **make its purpose clear**.

```
<input type="search" aria-label="Search the site">
```

Exercise 02:
Using the label, fieldset
and legend elements.

Open **exercise02-label-fieldset-legend/
exercise2-start.html**

Review your work against **exercise02-
label-fieldset-legend/exercise2-
finished.html**

1. Explicitly associate the “name” and “email” labels information with their form controls
2. Adjust the radio button set instructions using `<fieldset>` and `<legend>` elements.