

Web Programming

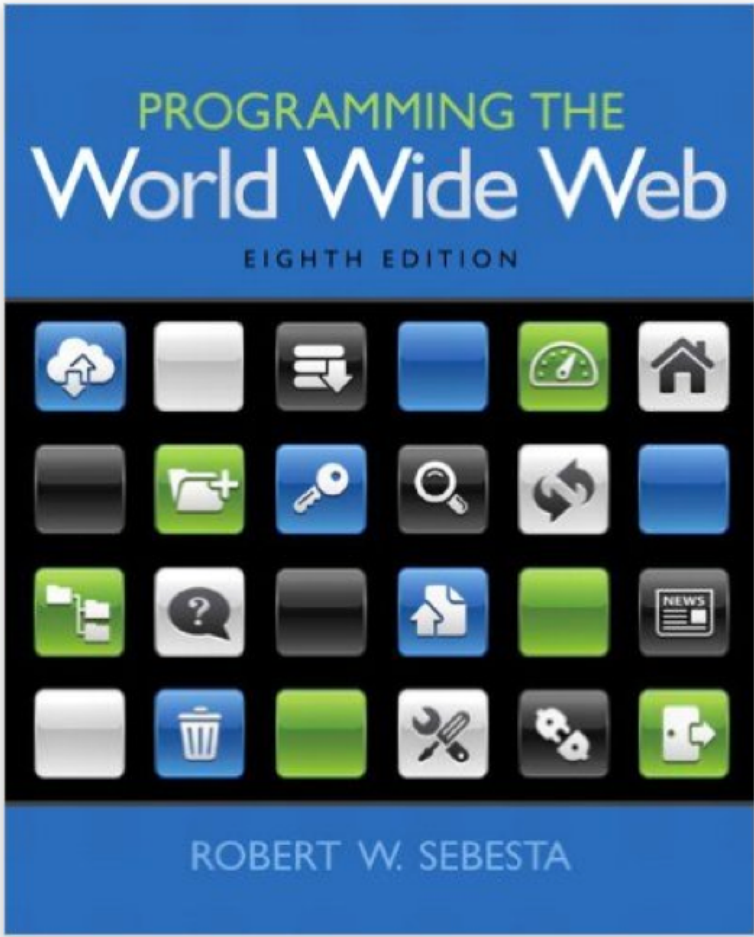
HTML Forms

Forms

- A group of UI controls that accepts information from the user and sends the information to a web server
- Today: how to make forms
- Sending and processing forms comes later

Programming the World Wide Web (8th Edition) 8th Edition

by Robert W. Sebesta (Author)
★★★★☆ 4 customer reviews



Paperback
\$24.25 - \$117.43

Other Sellers
from \$83.32

Rent \$24.25 - \$24.27

Buy new \$117.43

Only 2 left in stock (more on the way).
Ships from and sold by Amazon.com. Gift-wrap available.
List Price: \$147.40 Save: \$29.97 (20%)
24 New from \$99.95

This item ships to Stavanger, Norway. Want it Friday, Jan. 29? Order within 14 hrs 25 mins and choose AmazonGlobal Priority Shipping at checkout. [Learn more](#)

Qty: 1

Add to Cart

Turn on 1-Click ordering



Google Search

I'm Feeling Lucky

Email or Phone

Password

Log In

☐ Keep me logged in

[Forgotten your password?](#)

[Update Status](#) | [Add Photos/Video](#) | [Create Photo Album](#)



What's on your mind?

Friends

Post

Pick a username

Your email

Create a password

Use at least one lowercase letter, one numeral, and seven characters.

Sign up for GitHub

By clicking "Sign up for GitHub", you agree to our [terms of service](#) and [privacy policy](#). We will send you account related emails occasionally.

Name

First

Last

Choose your username

@gmail.com

Create a password

Confirm your password

Birthday

Day

Month

Year

Gender

I am...

Mobile phone

+47

Form

- In between **<form>** and **</form>**
 - **action** is the address where the form is sent
 - **method** — “**GET**” (default) or “**POST**”, how request should be sent

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

The input element

- **<input>** is used for several different types of controls (text, password, radio, checkbox)
- Attributes
 - **type** — determines the type of control (text, checkbox, radio, etc.)
 - **name** — identifies the form element when submitted to a server
- There are additional optional attributes depending on the type

Text input

- `<input type="text" name="..." />`
- Attributes
 - **maxlength** — maximum number of characters the user may enter
 - **value** — sets the default (initial) value for the field

Zip code: `<input type="text" name="zip" maxlength="4" />`

Zip code:

Password input

- `<input type="password" name="..." />`
- Attributes
 - **maxlength** — maximum number of characters the user may enter
 - **value** — sets the default (initial) value for the field
- Password is hidden only on the screen, it is not sent securely to the server (!)

Password: `<input type="password" name="pw" />`

Password:

Pattern validation

- Attribute
 - **pattern** — a regular expression that needs to match the input
 - Use **title** to give information to the user
- Example:

```
<input type="password" name="pw" pattern=".{8,}" title="Eight or more characters" />
```

```
<input type="password" name="pw"  
pattern="(?=.*\d)(?=.*[a-z])(?=.*[A-Z]).{8,}"  
title="Must contain at least one number and one uppercase and lowercase letter,  
and at least 8 or more characters" />
```

```
<input type="text" name="tel" pattern="[0-9- ]*" />  
title="May only contain numbers, minus and space."/>
```


Exercise #1 (#1b)



[github.com/dat310-2024/info/tree/master/
exercises/html/forms](https://github.com/dat310-2024/info/tree/master/exercises/html/forms)



Number input

- `<input type="number" name="..." />`
- Attributes
 - **min** — minimum value
 - **max** — maximum value
 - **step** — increments
 - **value** — sets the default (initial) value for the field

Quantity:

```
<input type="number" name="points" size="3" min="0" max="100" step="10" value="30">
```

Quantity:

Radio button

- Let the user select **only one** of a limited number of choices
- `<input type="radio" name="..." value="..." />`
 - Each option should have the same **name**
 - Value of value is sent to the server for the selected option
 - **checked** indicates which option should be selected initially

Preference:

```
<input type="radio" name="pref" value="male" checked /> Male
```

```
<input type="radio" name="pref" value="female" /> Female
```

Preference: ☒ Male ☐ Female

Checkbox

- Let the user select **zero or more** of a limited number of choices
- `<input type="checkbox" name="..." value="..." />`
 - Each option should have a different **name**
 - Value of **value** is sent to the server for the selected option
 - **checked** indicates if the option should be checked initially

Preference:

```
<input type="checkbox" name="pref1" value="male" /> Male  
<input type="checkbox" name="pref2" value="female" /> Female
```

Preference: ☐ Male ☐ Female

Textarea

- `<textarea name="...">...</textarea>`
- Attributes
 - **cols** — width (measured in characters)
 - **rows** — height (number of rows)
- Closing tag is mandatory!
- Content of the tag is the initial input value

What do you think?

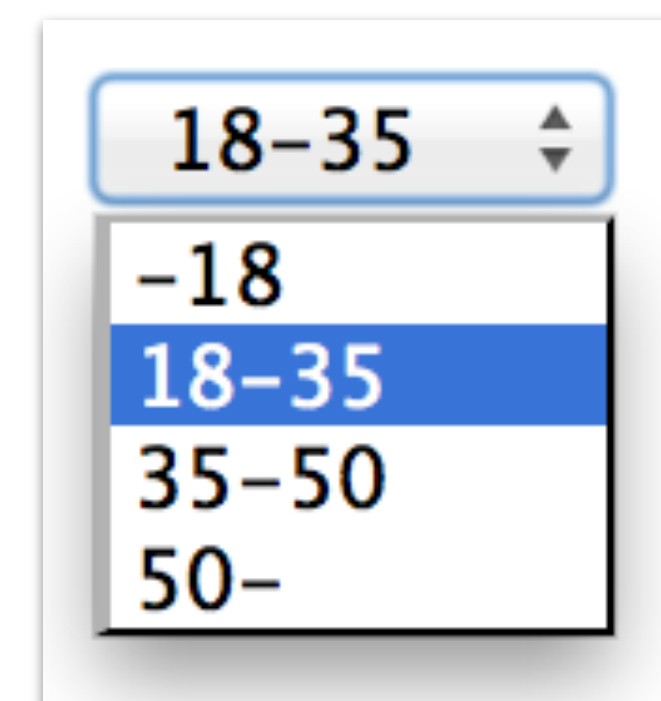
```
<textarea name="comment" cols="40" rows="3"></textarea>
```

What do you think?

Dropdown list

- Let the user select a **single option** from a dropdown list
- `<select name="...">...</select>`
- Each option is `<option value="...">...</option>`
 - Value of value is sent to the server for the selected option
 - **selected** can be used to set the default option

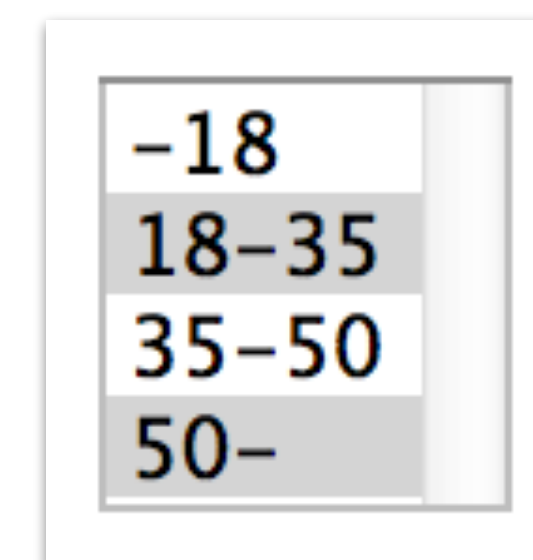
```
<select name="age">  
  <option value="1">-18</option>  
  <option value="2" selected>18-35</option>  
  <option value="3">35-50</option>  
  <option value="4">50-</option>  
</select>
```



Multichoice select list

- Let the user select **multiple options** from a list
- **<select name="..." multiple>...</select>**
 - **size** can set the number of options shown
- Each option is **<option value="...">...</option>**
 - Value of value is sent to the server for the selected option
 - **selected** can be used to set the default selection(s)

```
<select name="age" multiple>
  <option value="1">-18</option>
  <option value="2" selected>18-35</option>
  <option value="3">35-50</option>
  <option value="4" selected>50-</option>
</select>
```




Hidden form variables

- To pass on information that is not entered by the user
- Not shown on the page, but sent along to the server the same way as any other variable

```
<input type="hidden" name="secret" value="nosecret" />
```

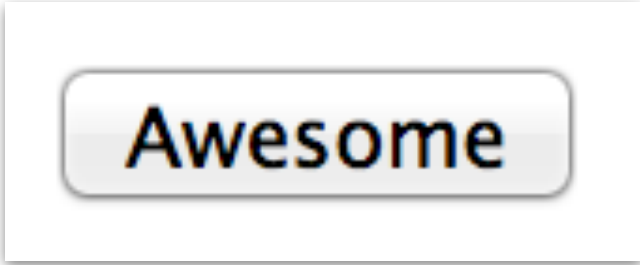

Global attributes

- **disabled** — the field is not usable, content cannot be copied from it
- **readonly** — the content cannot be changed, but a user can tab to the field and copy content from it
-  - **required** — the input field must be filled out before submitting the form

Submit button

- `<input type="submit" />`
- Attributes
 - **value** — the text that appears on the button
 - **name** — name of the button

```
<input type="submit" value="Awesome" />
```



A rectangular button with a light gray background and a thin border. The word "Awesome" is centered on the button in a dark gray, sans-serif font.

Exercises #2



[github.com/dat310-2024/info/tree/master/](https://github.com/dat310-2024/info/tree/master/exercises/html/forms)
exercises/html/forms

Labeling form controls

- `<label>` can be used in two ways
 - Wrapped around both the text description and the form input

```
<label>Name: <input type="text" name="name" /></label>
```

- Kept separate from the form control and using the **for** attribute
 - The **for** attribute of the `<label>` tag should be equal to the **id** attribute of the related element to bind them together

```
<label for="nameinput">Name</label>  
<input type="text" name="username" id="nameinput" />
```

Name



Placeholder text

- placeholder="..." — specifies a short hint that describes the expected value of an input field
 - The hint is displayed before the user enters a value into the field
 - Works for the following
 - input types: text, search, url, tel, email, password
 - textarea

```
<input type="text" name="name" size="20" placeholder="Firstname, lastname" />
```

Name

Grouping fields

- **<fieldset>...</fieldset>**
 - Used for grouping related elements in a form
 - **<legend>** can be used inside to provide a caption (optional)

```
<fieldset>
  <legend>Delivery address</legend>
  <label>City: <input type="text" name="city" size="10" /></label>
  <label>Zip: <input type="text" name="zip" size="4" /></label>
  <label>Street: <input type="text" name="street" size="20" /></label>
  <label>House no: <input type="text" name="housetno" size="4" /></label>
</fieldset>
```





More HTML5 input types

- Date
- Email
- URL
- Search
- Color
- ...



Pick a date

Email address

! Please include an '@' in the email address. 'john.smith' is missing an '@'.

URL:

! Please enter a URL.

Search

- See http://www.w3schools.com/html/html_form_input_types.asp

Exercise #3-#4



[github.com/dat310-2024/info/tree/master/](https://github.com/dat310-2024/info/tree/master/exercises/html/forms)
exercises/html/forms

Styling tip

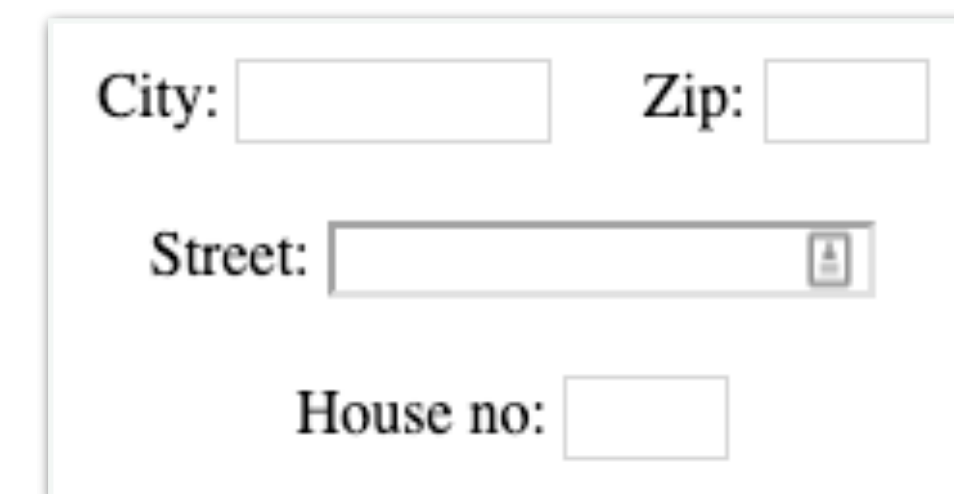
 [examples/html/forms/form_styling.html](https://github.com/alexanderdavydov/examples/html/forms/form_styling.html)

- To be able to nicely style a form, e.g. using a flexbox, it can be usefull to wrap each label and input into a separate **<div>**

```
<div>
  <label for="city">City:</label>
  <input type="text" name="city" size="10" id="city" />
</div>
<div>
  <label for="zip">Zip:</label>
  <input type="text" name="zip" size="4" id="zip" />
</div>
<div>
  <label for="street">Street:</label>
  <input type="text" name="street" size="20" id="street" />
</div>
<div>
  <label for="housetno">House no:</label>
  <input type="text" name="housetno" size="4" id="housetno" />
</div>
```



A screenshot of a web form rendered from the first code block. It contains four input fields arranged in a 2x2 grid. The top row has 'City:' followed by a text input of size 10, and 'Zip:' followed by a text input of size 4. The bottom row has 'Street:' followed by a text input of size 20, and 'House no:' followed by a text input of size 4. The labels are positioned to the left of their respective inputs.



A screenshot of a web form rendered from the second code block. It contains four input fields arranged in a 2x2 grid. The top row has 'City:' followed by a text input of size 10, and 'Zip:' followed by a text input of size 4. The bottom row has 'Street:' followed by a text input of size 20, and 'House no:' followed by a text input of size 4. The labels are positioned to the left of their respective inputs.

References

- HTML5 test
 - <https://css-tricks.com/centering-css-complete-guide/>
- Styling forms using CSS
 - <http://tutorials.jenkov.com/css/forms.html>
 - <https://jonathan-harrell.com/advanced-css-form-styling/>
 - <https://www.sanwebe.com/2014/08/css-html-forms-designs>
- A/B testing quiz
 - <http://bit.ly/2n7P6un>