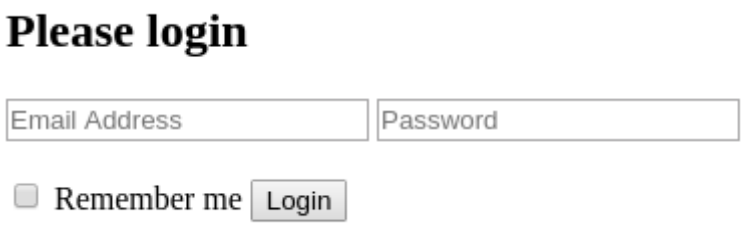
Theory: HTML forms

13 minutes 0 / 5 problems solved

§1. HTML Forms

Social networks, dating sites, fanpages, online shops: there are so many exciting opportunities to share your precious data with the world and get a sense of belonging. What could be more familiar than filling out yet another form on the Internet? You probably filled a few dozens of them in your lifetime; but have you ever created one? This topic will lead you through the process of creating an HTML form.

Forms are designed to send data from a web page to a server that can process it. By means of such forms, one can make purchases, send messages, register on various websites and so on:



Forms are always created to receive information and then transmit it to the server.

§2. Creating forms

To create your form, you should use the paired tag with a rather suggestive name `<form>` . The elements for information input are placed inside of that tag. It has two important attributes:

- `action` specifies the address of the program or document that processes the data of the form;
- `method` informs the server of the request method.

Attributes are used to extend the capabilities of individual tags.

Here is an example:

```
<!DOCTYPE html>
<html>
  <head>
    <title>HTML Forms</title>
  </head>
  <body>
    <form action="[value]" method="post">
       Input elements
    </form>

  </body>

</html>
```

The `get` and `post` methods are used to submit data from the form to a server. If the method attribute is not specified, `get` will be used by default.

The `get` method is designed to obtain the required information and pass the data in the address bar, so it is usually used in search forms.

The `post` method is used when you need to send a lot of data to the server: for example, when databases need to be populated, in forums and mail services, or for sending files.

§3. Input elements

Current topic:

HTML forms

Topic depends on:

× HTML attributes id and class

Topic is required for:

Pseudo-classes

Using models with templates

Submitting data

Table of contents:

Input elements are for the users to enter their information. They can be created using the `<input>` tag (no closing tag needed). All information that the browser needs is contained directly in the `<input>` tag and set with the help of various attributes. The input items come in different types: text fields, password, radio buttons, submit buttons, etc.

Let's consider the examples of the most popular input types.

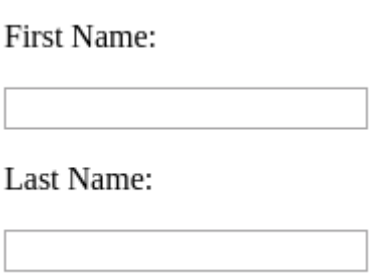
§4. Text Fields

Hello, user! What is your name? With `<input type="text">`, you can create a single-line text input field for the user to enter this information:

```
<form action="[value]" method="[value]">
  <p>First Name:</p>
  <input type="text" name="firstName">
  <p>Last Name:</p>
  <input type="text" name="lastName">
</form>
```

The `name` attribute specifies a unique name for a form element. This name is usually used when sending data to the server.

Here is the result that you get:



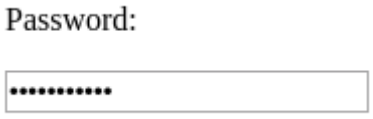
The default width of the text field is 20 characters.

§5. Password Field

Now that our user revealed their name, let's win their trust and protect their data.
`<input type="password">` provides the option to enter the password securely:

```
<form action="[value]" method="[value]">
  <p>Password:</p>
  <input type="password" name="password">
</form>
```

And this is what we get:



The characters entered in the password field are masked. They are displayed as black dots. This is very useful: no peeping Tom will spy on the password entered by the user, and the data will be safe and sound.

§6. Radio Buttons

Every new user is so valuable, and it makes sense we want to know more about them.
With `<input type="radio">`, you can create a radio button:

```
<form action="[value]" method="[value]">
  <input type="radio" name="language" value="english"> English
  <input type="radio" name="language" value="spanish"> Spanish
</form>
```

The result looks like this:

Radio buttons allow the user to select only one of the given options.

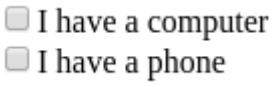
§7. Checkboxes

Sometimes one option is not enough. In this case, you can create checkbox forms using `<input type="checkbox">`:

```
<form action="[value]" method="[value]">

<input type="checkbox" name="technique" value="computer">I have a computer
  <br>
  <input type="checkbox" name="technique" value="phone">I have a phone
</form>
```

And this is the result:



Checkboxes allow the user to select any number of options from the list. In HTML, some elements may look stuck together, so the `<br>` tag is used in this code to move the element to the line below. Without it, the form would look like this:

We will talk about this strange behavior of the elements and other ways to solve similar problems in further lessons. Now let's get to know another tag that's often used in forms.

§8. Label

The `<label>` element links the text to the form element. The tag does not show itself visually, so to understand whether it is involved or not, you can only use the text. If clicking on the text activates a nearby form element, it means `<label>` works. Let's try to rewrite the previous example using the `<label>` tag:

```
<form action="[value]" method="[value]">
  <label>
<input type="radio" name="language" value="english">English</label>
  <label>
<input type="radio" name="language" value="spanish">Spanish</label>
</form>
```

To better understand the behavior of elements with and without the `<label>` tag, you can experiment with code on codepen.

The `<label>` tag has a `for` attribute that helps link the item with the text, regardless of its location. For instance:

```
<form action="[value]" method="[value]">
  <input id="english" type="radio" name="language" value="english">
  <label for="english">English</label>
  <label for="spanish">Spanish</label>
  <input id="spanish" type="radio" name="language" value="spanish">
</form>
```

In this example we added the `id` attribute to the `<input>` elements and then set the same values for the `id` attribute and `for`.

§9. Buttons

We collected enough information: now all we need is a magic button that will help us transfer the data. The paired tag `<button type="submit">` defines the button:

```
<form action="[value]" method="[value]">
  <button type="submit">Submit</button>
</form>
```

The result will look like this in browser:

Buttons are one of the most intuitive interface elements. From their appearance, it instantly becomes apparent that the only action you can take with them is to click on them. The button in forms is most often used to send data to the server. The data is sent to the page specified in the action form attribute.

Report a typo

181 users liked this theory. **3** didn't like it. What about you?

Start practicing

Comments (10) Hints (0) Useful links (3) Show discussion