Theory: Connecting JS to HTML

8 minutes 0 / 5 problems solved

If you want to display static content that the visitor of your page has little or no interaction with, HTML would suffice. However, web pages like this are slowly but surely going out of fashion. Most content on the World Wide Web is now *interactive* with things like slide shows, animation, drop-down menus, or pop-up windows. Various moving elements simplify perception, better convey information to the users, and make the site more dynamic.

The key to successfully managing the behavior of HTML elements lies in using the possibilities of programming languages. For working with web pages, the most common and convenient one is **JavaScript**. It allows developers to easily create websites that can interact with visitors.

You can include Javascript programs already placed via the Internet in your HTML file, or you can write your own.

Programs written in JavaScript are saved in files with the `.js` extension. The number of programs that can be connected is not limited, but remember that a large number of heavy files can badly decrease the speed of downloading a page.

There are several ways to connect a Javascript file to an HTML document. In this topic, we will consider the simplest one.

Current topic:

Connecting JS to HTML

Topic depends on:

HTML page structure Stage 1

§1. External JS file

You can add JavaScript code located in an external file to an HTML document. The script located inside the external file is called an external one. To connect JavaScript files to an HTML document, use the paired tag `<script>` with the attribute `src`. In the attribute, a path to the necessary file is specified.

Take a look at an example:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Connecting JavaScript to HTML</title>
  </head>
  <body>
    <script src="assets/js/main.js"></script>
  </body>

</html>
```

Here, JavaScript is connected in the body before the closing tag `</body>`.

§2. head VS body

However, some developers include JavaScript files not before the closing `</body>` tag, but inside the `<head>`. Here's an example of that:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Connecting JavaScript to HTML</title>
    <script src="assets/js/main.js"></script>
  </head>
  <body>
  </body>

</html>
```

This type of connection also works; so, does the position make a difference, and if so, where is it better to implement the `<script>` tag?

The difference does in fact exist. It is recommended to enable JavaScript at the end of the `<body>` section since this type of connection allows you to speed up the loading of the page. Web pages are loaded in the order specified in HTML markup: first, the browser analyzes the elements inside the `<head>` tag and then goes to `<body>`. If you use the connection in the `<head>` tag, the page content will not be visible in the browser until the JavaScript file is loaded. The situation may become critical with larger files. The priority here is to show the basic information posted on the web page so that the user does not sit looking at a blank screen monitor for a long time.

§3. Internal JS

JS code can also be written directly in HTML, not in a separate file. To do this, just enclose the JS code in a paired `<script>` tag:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Connecting JavaScript to HTML</title>
    <script>
      This is where your JS code should be.
    </script>
  </head>

  <body>

    <script>

      This is where your JS code should be.

    </script>

  </body>

</html>
```

A script located directly inside an element is called **an internal script**.

Now you know how to make your web pages even more functional; let's proceed to practice!

Report a typo

122 users liked this theory. **1** didn't like it. **What about you?**

Start practicing

Comments (0) Hints (0) Useful links (1) Show discussion