

Running Javascript in an HTML page

HTML, JS

Language Skills : 2

When you need to execute code on an HTML page, you immediately think of JavaScript. But what are the methods to do it properly. Here is a summary of the different ways to do it.

Published Monday October 4th 2021, 09:04
Modified Monday August 26th 2024, 10:04



By Olivier Paudex

Introduction

Executing Javascript on a web page can be done in a variety of ways. Even so, you still need to understand the advantages and disadvantages of each. Here are some tips for executing Javascript code in different ways.

The basic files

This publication is based on two files:

- **index.html**, which contains the HTML code of the page to be displayed
- **script.js**, which contains the Javascript code that will run on the objects on the HTML page

For the examples to work, both files must be created in the same folder.

The Javascript file

Let's start by creating a **"script.js"** file, which will serve as the basis for the examples.

Below is a sample code that will display in the browser console the DOM element whose ID is**"info".** It will then modify it, and then display its new content.

```
const myElement = document.getElementById('info');
console.log (myElement.innerHTML);
myElement.innerHTML = "Ceci est le texte modifié";console.log (myElement.innerHTML);
```

The HTML file

The second step is to create the **"index.html"** file that we will modify as we go through the examples.

Below is an example of code that will display a title, formatted as**"H2"**, then a paragraph, whose ID is **"info"**.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>My Web page</title>
  </head>
  <body>
    <div>
      <h2>Titre</h2>
      <p id="info">Ceci est un texte</p>
    </div>
  </body></html>
```

The embedded or classic method

The simplest method is to execute the JavaScript code directly from the HTML content, by adding the **"<script>"** and **"</script>" tags.**

It is also possible to call an external file this way.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>My Web page</title>
  </head>
  <body>
    <div>
      <h2>Titre</h2>
      <p id="info">Ceci est un texte</p>
    </div>
  </body>
  <script src="./script.js" type="text/javascript"></script></html>
```

However, the position of the **"<script>"** tag is very important. To execute the changes on the DOM object **"info"**, it is mandatory to write this line after the **"body"**.

The "defer"

If we want to execute the JavaScript code after the page is fully loaded, the easiest way is still to use the **"defer"** keyword. For this, you need to add the reference to the Javascript file in the **"head"** section of the HTML file. Unlike the classic method, this line is written in the header of the HTML page, which makes the code immediately much clearer.

```
<script src="./script.js" type="text/javascript" defer></script>
```

The "DOMContentLoaded"

To apply this method, add the same reference as above to the HTML code, but without the**"defer"**.

```
<script src="./script.js" type="text/javascript"></script>
```

Then, as a second step, we need to modify the JavaScript code. This method will create an event that fires after all DOM objects on the web page are loaded. In other words, you should only use this method if you do not want to modify the images and CSS code.

```
document.addEventListener("DOMContentLoaded", function(){
  const myElement = document.getElementById('info');
  console.log (myElement.innerHTML);
  myElement.innerHTML = "Ceci est le texte modifié";
  console.log (myElement.innerHTML);});
```

The "load" method

Like the DOM method, this method will create an event that fires after the web page is fully loaded, images and stylesheet included. With the page fully loaded, it is therefore safe to use this method.

```
window.addEventListener("load", function(){
  const myElement = document.getElementById('info');
  console.log (myElement.innerHTML);
  myElement.innerHTML = "Ceci est le texte modifié";
  console.log (myElement.innerHTML);});
```

The “onload”

The **“onload”** method is to be applied to HTML code, on the **“body”** DOM, for example. It allows to load a well-defined function.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>Ma page web</title>
    <script src="./script.js" type="text/javascript"></script>
  </head>
  <body onload="info()">
    <div>
      <h2>Titre</h2>
      <p id="info">Ceci est un texte</p>
    </div>
  </body></html>
```

The JavaScript code then becomes a simple function whose name is **“info”**.

```
function info(){
 const myElement = document.getElementById('info');
 console.log (myElement.innerHTML);
 myElement.innerHTML = "Ceci est le texte modifié";
 console.log (myElement.innerHTML);};
```

The jQuery library

The jQuery library makes it easy to write code. If using a CMS like**"WordPress"**, it is already included and ready to use. But if using no CMS, as in the examples in this publication, then, it is necessary to download the function library and reference it to the HTML file.

- **Download the latest version of jQuery.**
- Copy the file to the same folder as the HTML and Javascript file.
- Reference the library as below **(for compressed and minified version 3.6.0)**

```
<script src="jquery-3.6.0.min.js"></script>
<script src="./script.js" type="text/javascript"></script>
```

Once the reference is added to the HTML code, we can start writing jQuery code.

> **Beware of syntax**: Don't forget to close the parenthesis on the last line.

```
$(document).ready(function(){
  const myElement = document.getElementById('info');
  console.log (myElement.innerHTML);
  myElement.innerHTML = "Ceci est le texte modifié";
  console.log (myElement.innerHTML);});
```

The final word

These are some of the ways to run Javascript code on a web page, with or without a CMS. The difficulty of choice comes when thinking about site performance. Choosing to load the whole page before executing the Javascript code is often the best solution, but also the least efficient. As for jQuery, its methods have become so popular that the library is a standard applied by many developers.