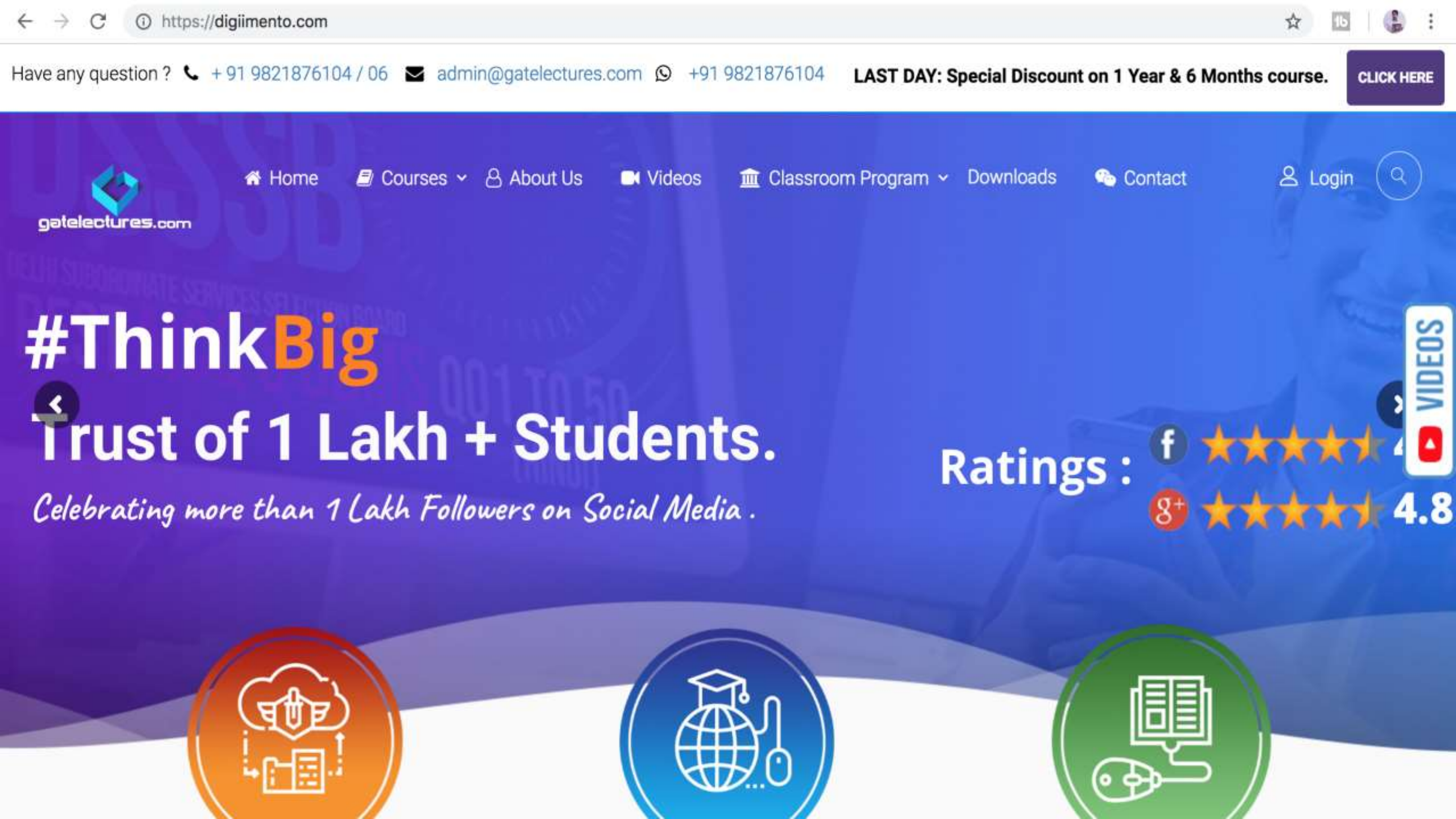
digimento

EDUCATING ON GO...

JAVA SCRIPT



https://digiimento.com

Have any question ? + 91 9821876104 / 06 admin@gatelectures.com +91 9821876104 **LAST DAY: Special Discount on 1 Year & 6 Months course.** CLICK HERE

#ThinkBig

Trust of 1 Lakh + Students.

Celebrating more than 1 Lakh Followers on Social Media .

Ratings : 4.8

The modern mode, "use strict"

- For a long time JavaScript was evolving without compatibility issues. New features were added to the language, but the old functionality did not change.
- That had the benefit of never breaking existing code. But the downside was that any mistake or an imperfect decision made by JavaScript creators got stuck in the language forever.
- It had been so until 2009 when ECMAScript 5 (ES5) appeared. It added new features to the language and modified some of the existing ones. To keep the old code working, most modifications are off by default. One needs to enable them explicitly with a special directive "use strict"

Reference https://javascript.info/strict-mode

The Modern Java Script Markups

The <script> tag has a few attributes that are rarely used nowadays, but we can find them in old code:

```
<script type="text/javascript"><!--
    ...
//--></script>
```

The old standard HTML4 required a script to have a type. Usually it was type="text/javascript". It’s not required any more.

The Modern Standard

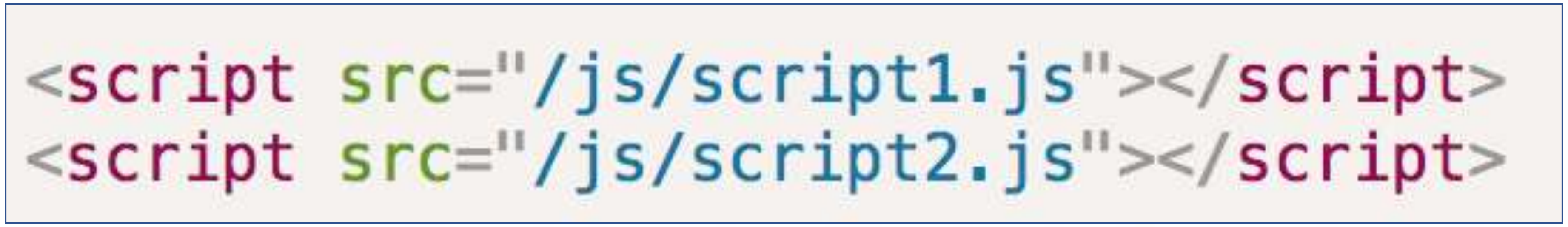
- The modern standard totally changed the meaning of the Script attribute.
- Now it can be used for Javascript modules.
- The language attribute: <script language=…>

```
<script type="text/javascript"><!--
    ...
//--></script>
```

- This attribute was meant to show the language of the script. As of now, this attribute makes no sense, the language is JavaScript by default. No need to use it.

External scripts

- As a rule, only the simplest scripts are put into HTML. More complex ones reside in separate files.
- The benefit of a separate file is that the browser will download it and then store it in its **cache.**
- After this, other pages that want the same script will take it from the cache instead of downloading it. So the file is actually downloaded only once.
- That saves traffic and makes pages faster.

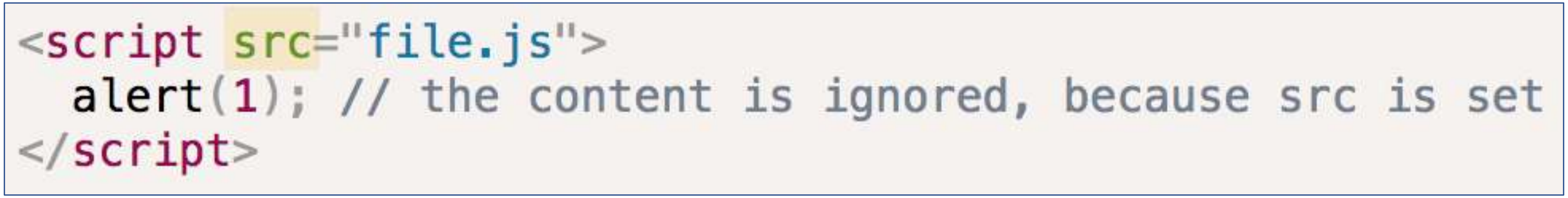


```
<script src="/js/script1.js"></script>
<script src="/js/script2.js"></script>
```

If src is set, the script content is ignored.

A single <script> tag can't have both the src attribute and the code inside.

This won't work:



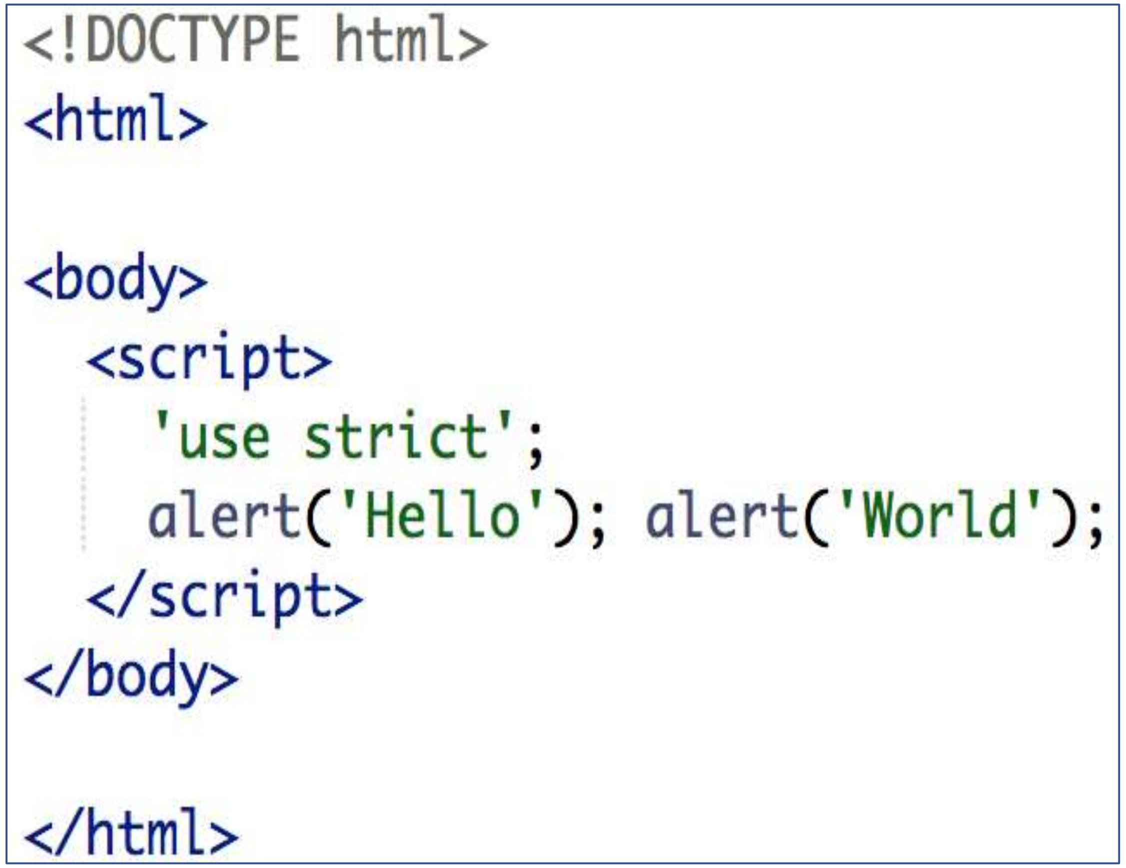
```
<script src="file.js">
  alert(1); // the content is ignored, because src is set
</script>
```

The example can be split into two scripts to work:

```
<script src="file.js"></script>
<script>
  alert(1);
</script>
```

“use strict”

- The directive looks like a string: **"use strict"** or **'use strict'.** When it is located on the top of the script, then the whole script works the “modern” way.
- Looking ahead let’s just note that **"use strict"** can be put at the start of a function (most kinds of functions) instead of the whole script. Then strict mode is enabled in that function only. But usually people use it for the whole script.



```
<!DOCTYPE html>
<html>

<body>
  <script>
    'use strict';
    alert('Hello'); alert('World');
  </script>
</body>

</html>
```

Ensure that “use strict” is at the top

Looking ahead let’s just note that "use strict" can be put at the start of a function (most kinds of functions) instead of the whole script. Then strict mode is enabled in that function only. But usually people use it for the whole script.

```
"use strict";

// this code works the modern way
...
```

Ensure that “use strict” is at the top

- Please make sure that "use strict" is on the top of the script, otherwise the strict mode may not be enabled.
- There is no strict mode here

```
alert("some code");
// "use strict" below is ignored, must be on the top

"use strict";

// strict mode is not activated
```

Only comments may appear above "use strict".

There’s no way to cancel use strict

There is no directive "no use strict" or alike, that would return the old behavior.

Once we enter the strict mode, there’s no return.

The differences of "use strict" versus the “default” mode

- At this point in time it’s enough to know about it in general:
- The "use strict" directive switches the engine to the “modern” mode, changing the behaviour of some built-in features. We’ll see the details as we study.
- The strict mode is enabled by "use strict" at the top. Also there are several language features like “classes” and “modules” that enable strict mode automatically.
- The strict mode is supported by all modern browsers.
- It’s always recommended to start scripts with "use strict". All examples in this tutorial assume so, unless (very rarely) specified otherwise.