

## What is embed code?

Embedding pages into other pages is not something that we invented. The technology to do this, iframes, has been around for two decades. You can use iframes to embed any site – unless the administrator of that site forbids it

For example, if you right-click on our you-tube video for the 2.8 release

[HTTPS://WWW.YOUTUBE.COM/WATCH?V=WWBLSA-II2U](https://www.youtube.com/watch?v=WWBLSA-II2U)

then you also can 'get embed code' When pasted, that code looks like this

```
<iframe width="854" height="480" src="https://www.youtube.com/embed/WWBLSa-II2U"
frameborder="0" allowfullscreen></iframe>
```

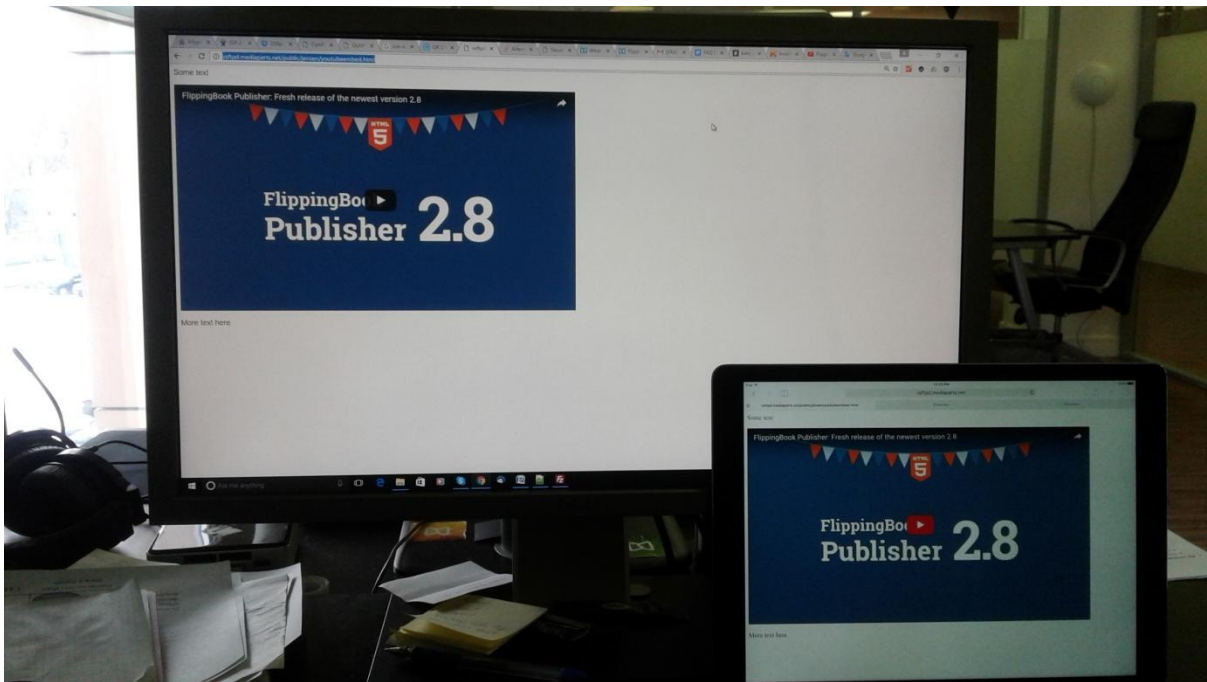
Now we can use this in a very simple HTML page<html>

```
<head>
</head>
<body>
Some text
<p>
<iframe width="854" height="480" src="https://www.youtube.com/embed/WWBLSa-II2U"
frameborder="0" allowfullscreen></iframe>
<p>
More text here
</body>
</html>
```

Which we can see here:

<http://vsftpd.mediaparts.net/public/jeroen/youtubeembed.html>

That works fine on all devices. Sure, a bit small on a phone, but that's expected. Still, everything works quite fine.



So to embed a page in another, we use iframes. In fact, this is the only possible way to technically embed a page from one website in another site!

The iframe from Youtube is about as minimal as you can get and contains 'attributes' or 'parameters' (= 'settings') that are always required:

- which parts of the screen should we use (width and height) and
- which site should we show there (src)

```
<iframe width="854" height="480" src="https://www.youtube.com/embed/WWBLSa-II2U"
frameborder="0" allowfullscreen></iframe>
```

These attributes are always required, but there are lots of other optional attributes. These attributes control how the iframe looks, and how it behaves in certain circumstances. In the youtube example we have:

- frameborder="0"
- allowfullscreen

In our 'old' FlippingBook iframe code we have a few others. Our standard code looks like this

```
<iframe frameborder="0" width="640" height="480" title="FlippingBook Publisher 2 - Product
Guide" src="https://flippingbook.cld.bz/product-guide" type="text/html" scrolling="no"
marginwidth="0" allowfullscreen="true" marginheight="0"></iframe>
```

Or the same, but a bit clearer

```
<iframe
  frameborder="0"
  width="640"
  height="480"
  title="FlippingBook Publisher 2 - Product Guide"
  src="https://flippingbook.cld.bz/product-guide"
  type="text/html"
  scrolling="no"
  marginwidth="0"
  allowfullscreen="true"
  marginheight="0">
</iframe>
```

So we give a bit more information to the browser on how to behave than YouTube does. We give the iframe a title, tell what kind of content is in there (html), that we don't want scrolling bars, and that there shouldn't be any additional margins. Note that browsers generally don't add margins anyway, but we tell anyway, just in case.

There are quite a few other attributes that you can add to your iframe too.

As you can see, there are different ways to describe these attributes: We use `allowfullscreen="true"`, whereas YouTube simply says : `allowfullscreen` (technically, both do the same)

To make it more confusing, there is a special attribute called '**style**' which can be used instead of other attributes. For example, the following iframe uses the style attribute to make use of the COMPLETE screen. So it looks exactly like a full screen version of our publication, only the address (URL) is different.

```
<iframe
  style="position:fixed;
  top:0px;
  left:0px;
  bottom:0px;
  right:0px;
  width:100%;
  height:100%;
  border:none;
  margin:0;
  padding:0;
  overflow:hidden;
  z-index:-10;"
  title=" your title"
  src="http://ctu-ar2016.sedgwick-richardson.hk">
</iframe>
```

OK, so to recap:

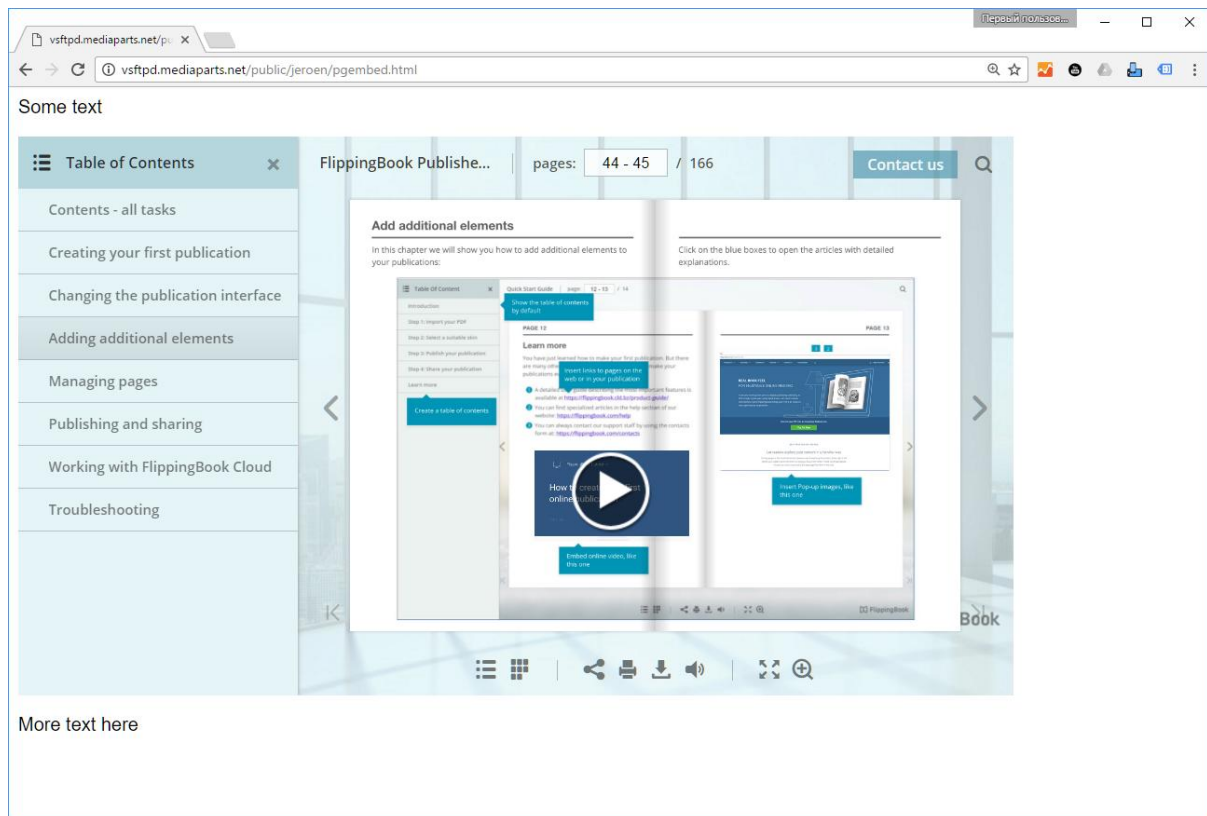
- An iframe can be used to embed ANY site that allows it, and in fact, it is the only way to do so.
- An iframe always has a width, height and src (site to show in the frame)
- There are lots of other attributes that can be used to change the appearance of the iframe.

## Embedding FlippingBooks

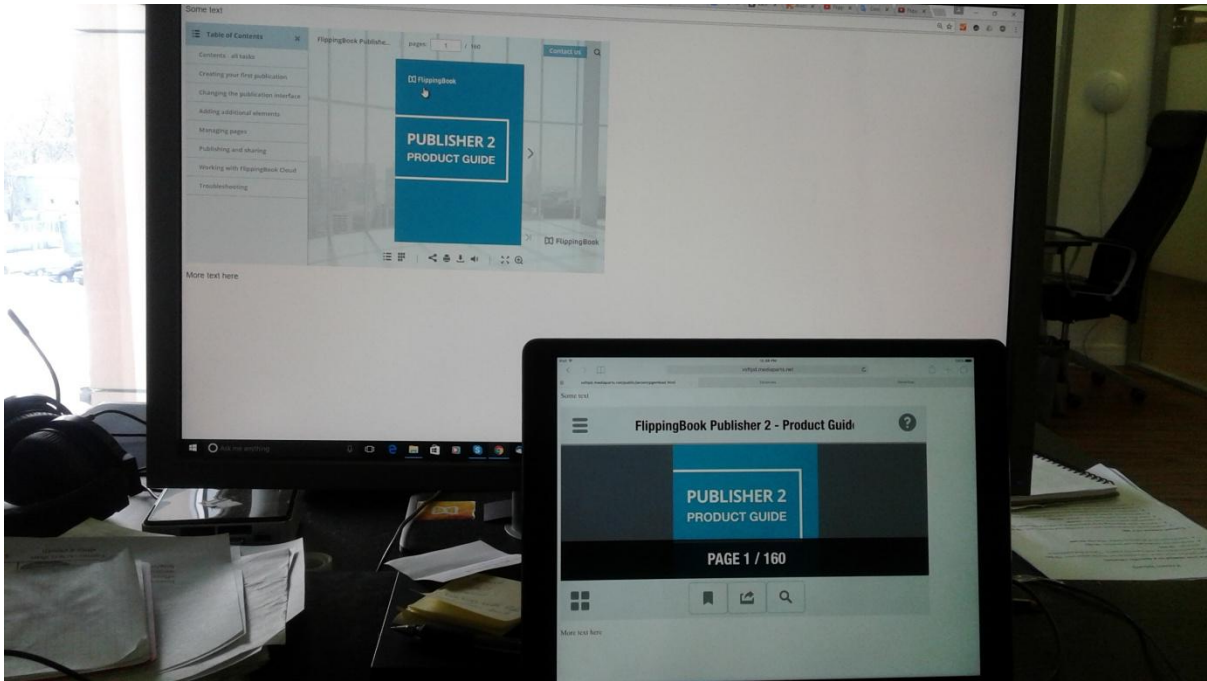
Now what happens when we replace the you-tube src with a flippingbook link?

<http://vsftpd.mediaparts.net/public/jeroen/pgembed.html>

As you can see it starts of more or less the same:



The difference is when you start to use it on Mobile devices.



The problem is that we determine the size of the interface on some information about the screen. That works fine if you open a flippingbook directly on a mobile device, but the iframe screws this up. Depending on the device and the iframe, it may even scale the buttons so large that they overlap each other and become unusable. Since we cannot control what devices end users use, in which orientation they hold them, and what embed code our users use, we NEVER recommend iframes directly. It was fine a few years ago, when nobody took too much offense if a site looked a bit strange on mobile, but it does matter now.

So what can we do?

- Use cld.bz and use embed code from there
- Use adaptive code (close to cld)
- Suggest to use lightbox

Problem : all these solutions rely on scripts that add programming code. Not all content management systems (The system that clients use to design their HTML-pages in) allow this. So our solutions may not work everywhere: e.g. WordPress (never) , Weebly (sometimes).

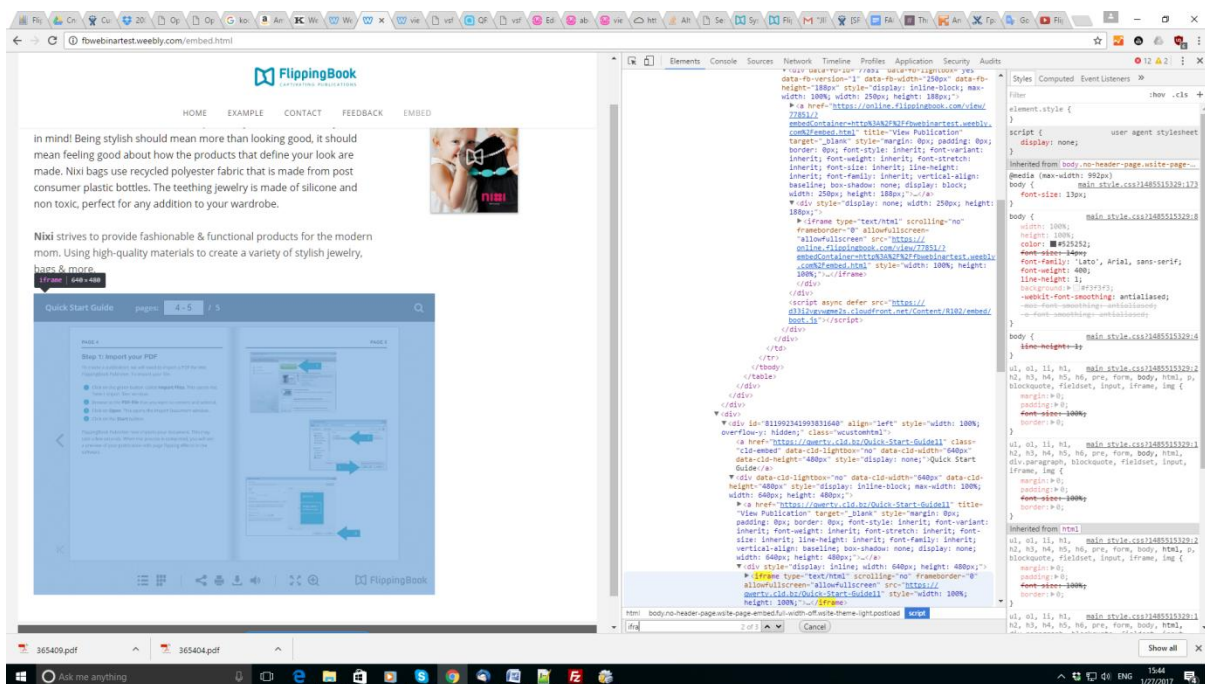
## ABOUT OUR NEW EMBED CODE

To deal with the 'mobile' problem, we have the option to generate new embed code for publications on Cloud. You can get this code by starting the Cloud manager from FlippingBook Publisher, and then Share the publication. Our new embed code looks something like this:

```
<a href="https://qwerty.cld.bz/Quick-Start-Guide11" class="cld-embed" data-cld-lightbox="no" data-cld-width="640px" data-cld-height="480px">Quick Start Guide</a><script async defer src="https://cld.bz/content/embed-boot/boot.js"></script>
```

So with this code, you can embed our Flipbooks into other pages... but didn't we say that the only way to embed was with iframe? How can that be? Well, the new embed code uses a script (yellow) which is located on cld.bz. We generate a few parameters too - our usual width and height (grey) and a link to show in the embed-area (green).

The script that we use reacts to all HTML-links that are marked as "cld-embed" (like in our case, lightblue) The script is a small program that determines if the user uses a mobile device or a desktop. If it is a normal desktop, then the script looks at the other parameters for the link and then quickly generates an iframe. If the script detects a mobile device, then it doesn't generate an iframe, but a clickable cover picture.



Other than that it says that lightbox="no" (if we use lightbox –yes, then it of course will use a lightbox).

## ABOUT 'ADATIVE' CODE

If the client is looking for a solution similar to the 'new embed code' but is self-hosting then it requires a bit more effort from his side. Such is the life of self-hosting. But it is also possible to get something similar for self-hosted publications. The 'bare minimum' example for this is

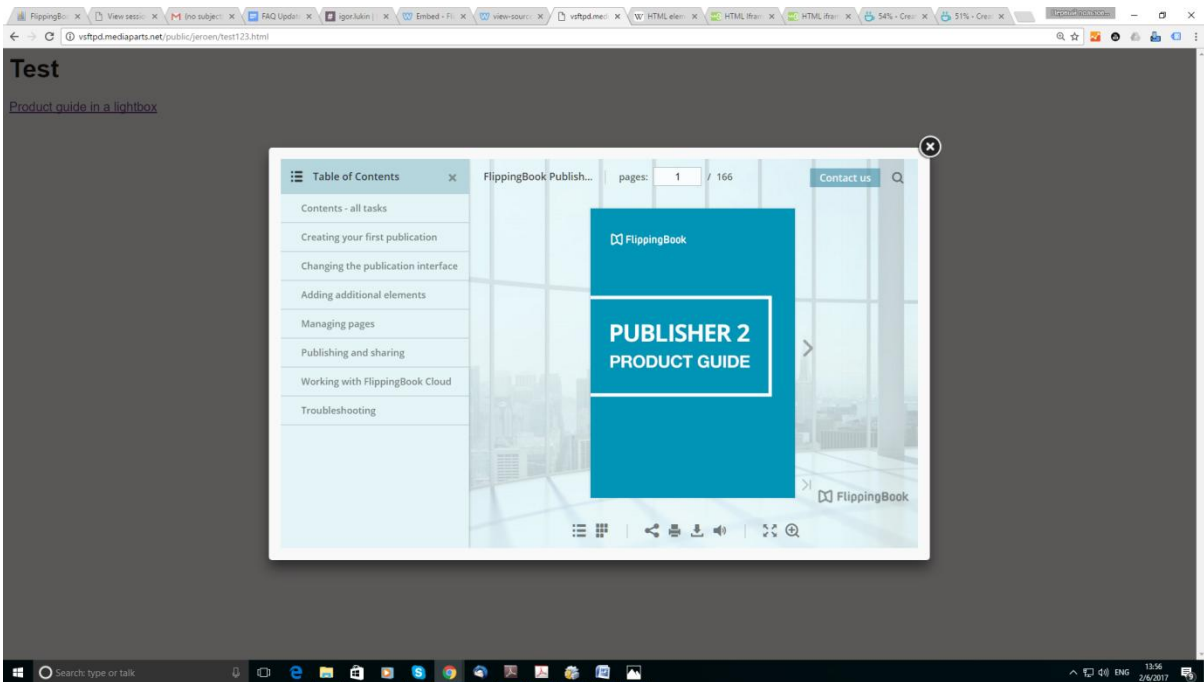
<http://vsftpd.mediaparts.net/public/jeroen/DualEmbed.html>

We can open this on a normal desktop and it shows a Flippingbook which is embedded in an otherwise empty page. If you open the same page on mobile, you will instead see a clickable cover page. When clicked, it opens the FlippingBook in a new tab.

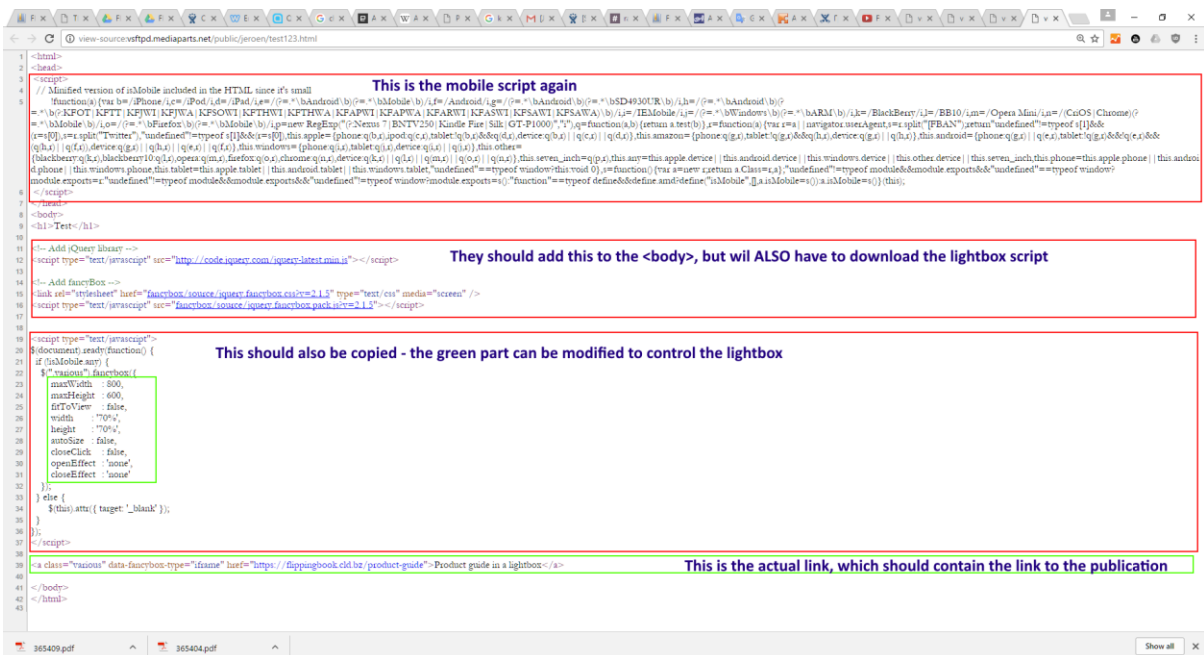
To make this work, the client will have to be able to edit his page to a greater extent than just pasting embed code. But it is also not impossibly difficult, and if a client chooses to host his own site,







Technically, they create an extra layer over the main page, that is mostly transparent grey. Then a white box, and in this white box, there is an iframe with the page.



If a client wants to do this, then T1 is not expected to be able to assist with that (you can try if you want, I think I will make a macro). It is even more technical than the Adaptive Code. But you SHOULD know about the possibility and explain to a client what it achieves.

Note that to use this function, the client will have to download a separate set of files that create lightboxes and upload them to his server. So that is a bit more complex still than to use the adaptive code.



## What about Wordpress?

Wordpress doesn't allow any scripting in code that users embed into it. So to overcome this, we have developed a plugin. That wasn't a bad idea, but it's a bit confusing as we presented it as that you can now 'Publish to Wordpress'. But in fact, with the plugin, you don't upload anything to wordpress. You upload to cloud, and after that you embed to WP.

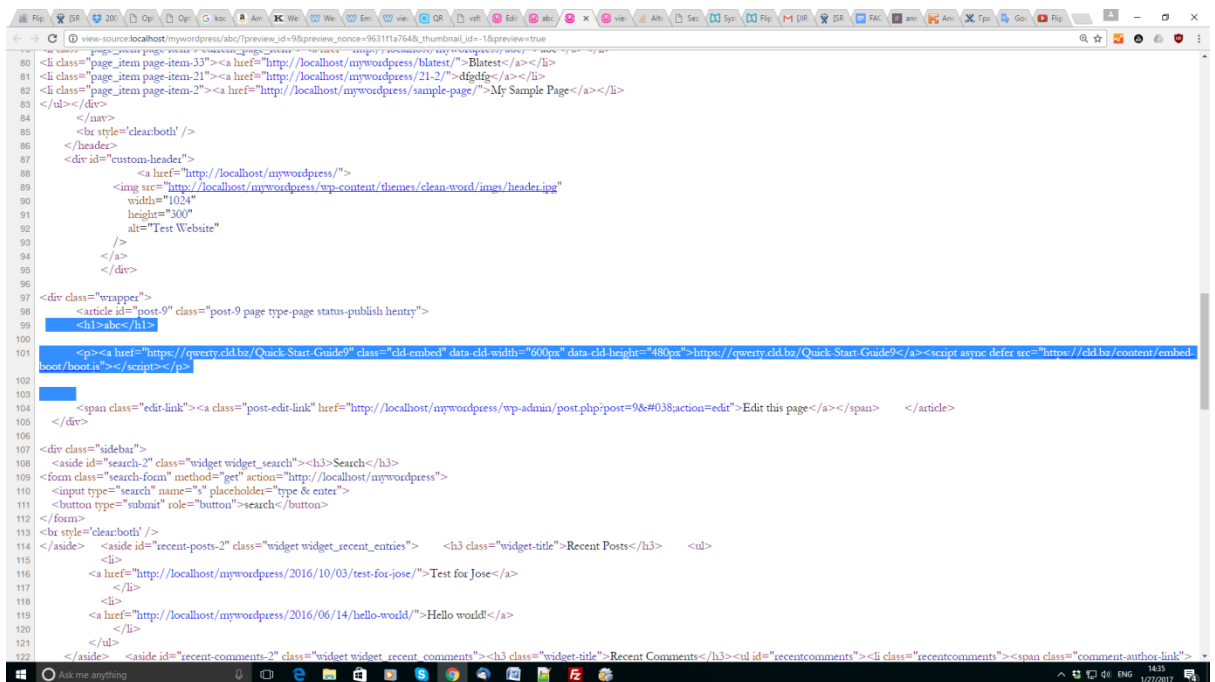
The embed code looks like this:

```
[EMBED WIDTH="600" HEIGHT="480"]HTTPS://QWERTY.CLD.BZ/QUICK-START-GUIDE9[/EMBED]
```

But if you look at the actual source code from the page when it is loaded in a browser, then you will see that the actual code looks like this:

```
<h1>abc</h1>
```

```
<p><a href="https://qwerty.cld.bz/Quick-Start-Guide9" class="cld-embed" data-cld-width="600px" data-cld-height="480px">https://qwerty.cld.bz/Quick-Start-Guide9</a><script async defer src="https://cld.bz/content/embed-boot/boot.js"></script></p>
```

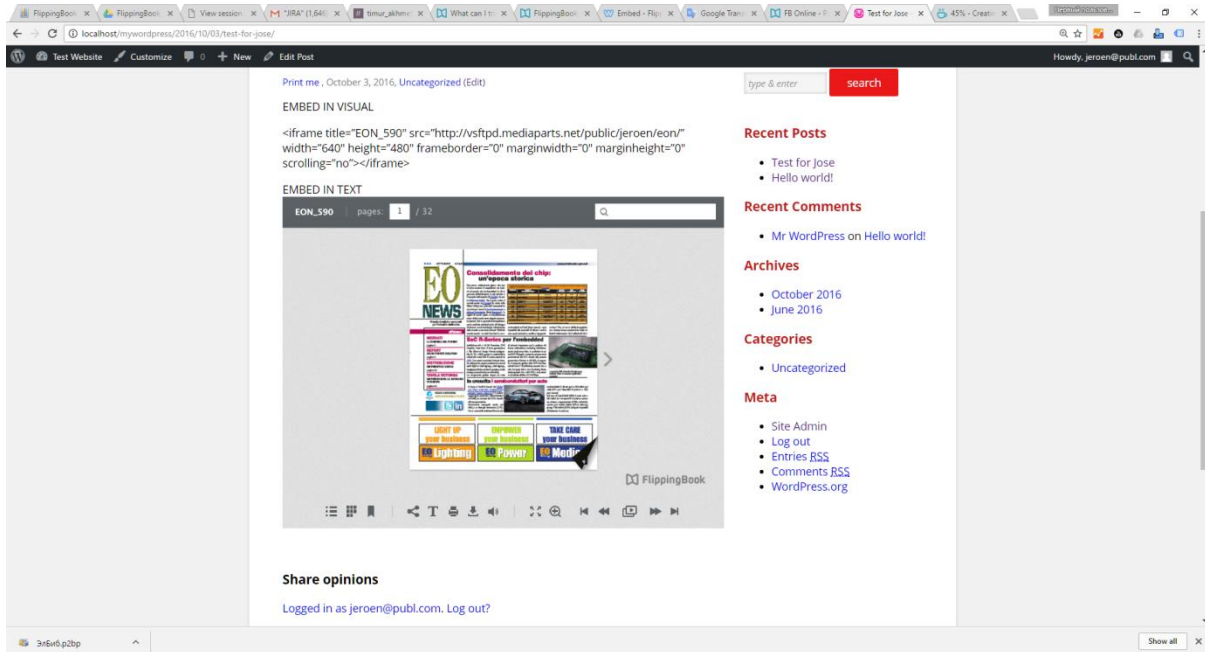


So that is almost the same as the 'new embed code' that we can generate for Publisher:

```
<a href="https://qwerty.cld.bz/Quick-Start-Guide11" class="cld-embed" data-cld-lightbox="no" data-cld-width="640px" data-cld-height="480px">Quick Start Guide</a><script async defer src="https://cld.bz/content/embed-boot/boot.js"></script>
```

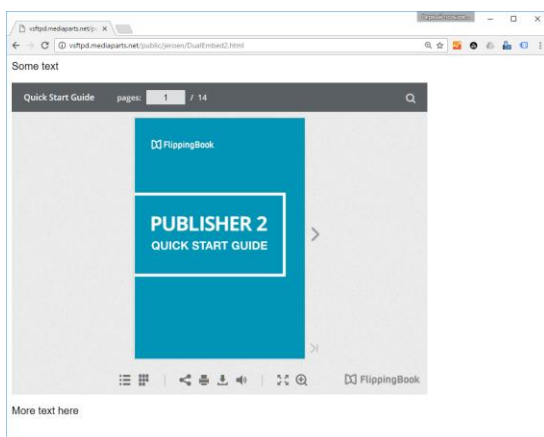
In short: the Wordpress plug-in creates the same code that Publisher generates. However, that only works for publications on the FlippingBook Cloud. And because you can't embed scripts in

WordPress, the only way to embed self-hosted publications is to use i-frames and accept that on mobile it will look strange. To do this, clients can use our old iframe code and in Wordpress go to the TEXT-view and paste it there. (it won't work when you paste it in the VISUAL view).



## Troubleshooting

If the client has problems with embed code, then we often have to look at the code itself to see what is going on. To do this, you will have to open the page on which he embedded the code. For example: <http://vsftpd.mediaparts.net/public/jeroen/DualEmbed2.html>



Now you can right-click anywhere on the page and choose 'View page source'

