

VIEWERJS

(/). DOCUMENTS. ON YOUR SITE. NOW.

[Home \(/\)](#)

[Examples \(/examples/\)](/examples/)

[Instructions \(/instructions/\)](/instructions/)

[Fonts \(/instructions/fonts.html\)](/instructions/fonts.html)

[Get It \(/getit/\)](/getit/)

[Contact \(/contact/\)](/contact/)

HOW TO USE VIEWERJS

Putting ViewerJS on your website is pretty easy, and shouldn't take more than five minutes. ViewerJS is a pretty versatile document viewer that resides on your web server itself.

Running ViewerJS is as simple as making a link. You can use it to give a full screen presentation, but also for showing documents inline in your website or as a viewer for links to documents. Because of the way your browser protects itself against security risks (which is a very good thing) ViewerJS only works on documents that reside on your own server. ViewerJS is complete and self-contained - it does not rely on other services.

SYNTAX

For using ViewerJS you don't have to change the page you want to add the document to - activating the viewer pretty much works like making a regular link. Normally when you create a link to a document on your site you would use:

```
<a href="/path/to/filename.ext">
```

For initiating ViewerJS we point the browser to the viewer which (if you followed the instructions) resides at `/ViewerJS/index.html`. We only need to write down `/ViewerJS/`, the browser will understand the rest. The next thing we need to do is tell the browser the location of the actual file we want to show.

We tell ViewerJS where it can find the document by giving the regular `path/to/filename.ext` (that is: the absolute path of the document counted from the 'root directory' of your website) like so:

In practical terms that means we indicate to the browser to go back to the top folder with `#..` (if you put the directory with ViewerJS elsewhere,

```
<a href="/ViewerJS/#../path/to/filename.ext">
```

(Since we are already using the regular path in the browser for pointing it to the folder with ViewerJS, we use a relative link back from that location preceded by the `#` character.)

That is it. In principle you don't need to change anything to the 'host' page. Of course you often want to use ViewerJS for embedding a document inside a page. For that we will use a so called `iframe` - really just a page within a web page. Check out the [examples \(/examples/\)](/examples/), and just open the source of the webpage to see how it works.

QUICK START

Here are the basic steps that are necessary to put ViewerJS on your website.

WHAT IS VIEWERJS?

ViewerJS is a combination of a number of excellent open source tools that are built on HTML and JavaScript. ViewerJS was funded by NLnet foundation, consider a [donation \(http://nlnet.nl/donating\)](http://nlnet.nl/donating).



The heavy lifting in ViewerJS is done by these awesome projects:

PDF.JS (BY MOZILLA)

PDF.js is a library created by Andreas Gal and others at Mozilla Labs. It is an HTML5 technology experiment that explores building a faithful and efficient Portable Document Format (PDF) renderer without native code assistance. PDF.js is community-driven and supported by [Mozilla Labs \(http://labs.mozilla.org\)](http://labs.mozilla.org). Its goal is to create a general-purpose, web standards-based platform for parsing and rendering PDFs, and eventually release a PDF reader extension powered by PDF.js.

[Visit project website \(http://mozilla.github.io/pdf.js/\)](http://mozilla.github.io/pdf.js/)



WEBODF (BY KO GMBH)

WebODF is a JavaScript library that shows office documents created by [KO GmbH](http://kogmbh.com)



<http://kogmbh.com>). It was started by Jos van den Oever at KO and is now developed by a growing team including external collaborators. It makes it easy to add Open Document Format (ODF) support to your website and to your mobile or desktop applications. It uses HTML and CSS to display ODF documents.

[Visit project website \(http://webodf.org\)](http://webodf.org)

1. Everything starts with [downloading ViewerJS](#) (</releases/ViewerJS-latest.zip>), from the website.
2. Unpack the file on your computer. This will give you a folder called 'ViewerJS'. Upload that folder to your webserver, preferably to the top folder. Now you are ready to see ViewerJS in action.
3. Upload a document to your webserver (for the demo we'll use a PDF, if you don't have one [use this](#) (</demo/doc.pdf>)). If you already have documents on there, you can skip this step.
4. See if the document is where you think it is, by inserting the address in the address bar of your browser. If your browser offers to download the document, you are set. If not, check if you have the correct location of the file.
5. Create the link to the document as you normally would do (that link can be anywhere, as long as both ViewerJS and the document you want to link to are on the same website). Now comes the tricky part: we have to use the syntax described above to point to the document.

```
<a href="/ViewerJS#../path/to/filename.ext">
```

(If you want, you can always try to see if ViewerJS is installed correctly by going to the address bar of the browser, and inserting **/ViewerJS#..** inbetween the domain name part of your website (below it is example.org) and the rest of the path:

```
<a href="http://example.org/ViewerJS#../path/to/filename.ext">
```

If that works, you can just copy that URI as well.

EMBEDDING IN A PAGE

To embed a document in the same way as you see here:



we need to draw an iframe and then pass the same link as we had before to it. So if we have a document in folder **/demo/ohm2013.odp** it will look something like this:

```
<iframe src = "/ViewerJS#../demo/ohm2013.odp" width='400' height='300' allowfullscreen webkitallowfullscreen></iframe>
```

DOWNLOAD OR FORK NOW



Download now (see [instructions](#) ([/instructions/](#))), or fork the code at GitHub and send us a pull request.



DOWNLOAD

([HTTP://VIEWERJS.ORG/RELEASES/LATEST.ZIP](http://viewerjs.org/releases/latest.zip))



SOURCE CODE

([HTTPS://GITHUB.COM/KOGMI](https://github.com/kogmi))

Note that you need to set the width and the height of the iframe in your page. ViewerJS will try to use all the space you give it, but if you want to make optimal use of your screen estate make sure that you fit the window of ViewerJS to the dimensions of the document you want to show. If you don't know what to do, just try it out with some different formats.

Common formats:

(A4 sized documents - portrait) (1.414 : 1)

- width='724' height='1024'
- width='566' height='800'
- width='389' height='550'
- width='297' height='210'

(A4 sized documents - landscape) (1 : 1.414)

- width='1024' height='724'
- width='800' height='566'
- width='550' height='389'
- width='210' height='297'

(screen sized documents) (4 : 3)

- width='1024' height='768'
- width='800' height='600'
- width='600' height='450'

If you want to position the document to the left or right, use **style="float:left;"** or **style="float:right;"**. If you want to center the document, use **style="text-align:center;"**:

```
<iframe style="float:right;" src = "/ViewerJS/#../demo/ohm2013.odp" width='400'  
height='300' allowfullscreen webkitallowfullscreen></iframe>
```