o install using npm:

```
npm install print-js --save
```

To install using yarn:

```
yarn add print-js
```

When installing via npm or yarn, import the library into your project:

```
import print from 'print-js'
```

CDN is also available, thanks to KeyCDN:

```
https://printjs-4de6.kxcdn.com/print.min.js
https://printjs-4de6.kxcdn.com/print.min.css
```

Getting Started

First we need to include the Print.js library on the page.

```
<script src="print.js"></script>
```

If you will use the modal feature, also include Print.css on the page.

```
<link rel="stylesheet" type="text/css" href="print.css">
```

That's it. You can now use Print.js in your pages.

When writing your javascript code, remember that the library occupies a global variable of `printJS`.

Using Print.js

There are four print document types available: `'pdf'`, `'html'`, `'image'` and `'json'`.

The default type is `'pdf'`.

It’s basic usage is to call `printJS()` and just pass in a PDF document url: `printJS('docs/PrintJS.pdf')`.

For image files, the idea is the same, but you need to pass a second argument: `printJS('images/PrintJS.jpg', 'image')`.

To print HTML elements, in a similar way, pass in the element id and type: `printJS('myElementId', 'html')`.

When printing JSON data, pass in the data, type and the data properties that you want to print:

`printJS({printable: myData, type: 'json', properties: ['prop1', 'prop2', 'prop3']})`;

Configuration

Print.js will accept an object as argument, where you can configure some options:

Argument
Default Value
Description
printable
null
Document source: pdf or image url, html element id or json data object.
type
’pdf’
Printable type. Availble print options are: pdf, html, image, json and raw-html.
header
null

Optional header to be used with HTML, Image or JSON printing. It will be placed on the top of the page. This property will accept text or raw HTML.

headerStyle

’font-weight: 300;’

Optional header style to be applied to the header text.

maxWidth

800

Max document width in pixels. Change this as you need. Used when printing HTML, Images or JSON.

css

null

This allow us to pass one or more css files URLs that should be applied to the html being printed. Value can be a string with a single URL or an array with multiple URLs.

style

null

This allow us to pass a string with custom style that should be applied to the html being printed.

scanStyles

true

When set to false, the library will not process styles applied to the html being printed. Useful when using the `css` parameter.

targetStyle

null

By default, the library process some styles only, when printing HTML elements. This option allows you to pass an array of styles that you want to be processed. Ex.: [’padding-top’, ’border-bottom’]

targetStyles

null

Same as \`targetStyle\`, however, this will process any a range of styles. Ex.: [’border’, ’padding’], will include ’border-bottom’, ’border-top’, ’border-left’, ’border-right’, ’padding-top’, etc. You can also pass [’*’] to process all styles.

ignoreElements

[ ]

Accepts an array of html ids that should be ignored when printing a parent html element.

properties

null

Used when printing JSON. These are the object property names.

gridHeaderStyle

’font-weight: bold;’

Optional style for the grid header when printing JSON data.
gridStyle
'border: 1px solid lightgray; margin-bottom: -1px;'
Optional style for the grid rows when printing JSON data.
repeatTableHeader
true
Used when printing JSON data. When set to `false`, the data table header will show in first page only.
showModal
null
Enable this option to show user feedback when retrieving or processing large PDF files.
modalMessage
'Retrieving Document...'
Message displayed to users when `showModal` is set to `true`.
onLoadingStart
null
Function to be executed when PDF is being loaded
onLoadingEnd
null
Function to be executed after PDF has loaded
documentTitle
'Document'
When printing html, image or json, this will be shown as the document title.
fallbackPrintable
null
When printing pdf, if the browser is not compatible (check browser compatibility table), the library will open the pdf in a new tab. This allow you to pass a different pdf document to be opened instead of the original passed in `printable`. This may be useful if you inject javascript in your alternate pdf file.
onPdfOpen
null
When printing pdf, if the browser is not compatible (check browser compatibility table), the library will open the pdf in a new tab. A callback function can be passed here, which will be executed when this happens. It may be useful in some situations where you want to handle your print flow, update user interface, etc.
onPrintDialogClose
null
Callback function executed once the browser print dialog is closed.

onError
error => throw error
Callback function to be executed when an error occurs.
base64
false
Used when printing PDF documents passed as base64 data.
honorMarginPadding (deprecated)
true
This is used to keep or remove padding and margin from elements that are being printed. Sometimes these styling settings are great on screen but it looks pretty bad when printing. You can remove it by setting this to false.
honorColor (deprecated)
false
To print text in color, set this property to true. By default all text will be printed in black.
font (deprecated)
'TimesNewRoman'
Typeface used when printing HTML or JSON.
font_size (deprecated)
'12pt'
Font size used when printing HTML or JSON.
imageStyle (deprecated)
'width:100%;'
Used when printing images. Accepts a string with custom styles to be applied to each image.

Browser Compatibility

Currently, not all library features are working between browsers. Below are the results of tests done with these major browsers, using their latest versions.
Google Chrome
Safari
Firefox
Edge
Opera
Internet Explorer
PDF
HTML
Images
JSON
Thank you BrowserStack for the support. Amazing cross-browser testing tool.