

Foreword

HTML and CSS are beautiful languages and one of the landmark achievements of collaboration and the open web. Together, they allow you to create beautifully typeset digital content that can be viewed on any device, without any special software or paid licenses.

The web however was developed for scrolling content, and not paged media (by which we mean discrete pages, like in print or PDF). Unfortunately, whilst there are [W3C standards](#) on paged media, adoption by browsers has not been adequate to date. High quality paid alternatives exist in [PrinceXML](#), but a native, free solution has been missing.

A [happenstance discussion](#) on Hacker News brought my attention to [paged.js](#), a genius idea to polyfill the required capabilities to allow browsers to natively handle paged media, without having to build the whole rendering engine from scratch.

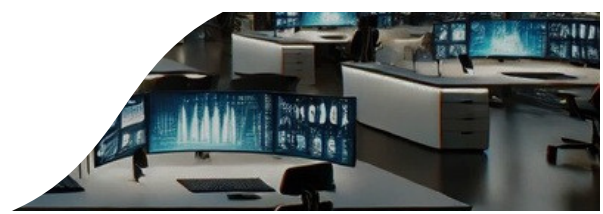
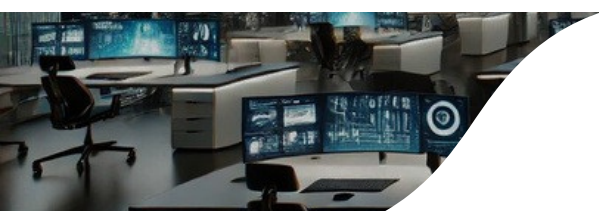
This guide provides a basic tutorial to using paged.js, for my own future reference and yours, and I highly recommend you visiting their site and supporting their project. In particular, their [documentation](#) covers many aspects of advanced paged media and is a worthwhile next step after reading this guide.

HTML and CSS are beautiful languages and one of the landmark achievements of collaboration and the open web. Together, they allow you to create beautifully typeset digital content that can be viewed on any device, without any special software or paid licenses.

The web however was developed for scrolling content, and not paged media (by which we mean discrete pages, like in print or PDF). Unfortunately, whilst there are [W3C standards](#) on paged media, adoption by browsers has not been adequate to date. High quality paid alternatives exist in [PrinceXML](#), but a native, free solution has been missing.

A [happenstance discussion](#) on Hacker News brought my attention to [paged.js](#), a genius idea to polyfill the required capabilities to allow browsers to natively handle paged media, without having to build the whole rendering engine from scratch.

This guide provides a basic tutorial to using paged.js, for my own future reference and yours, and I highly recommend you visiting their site and supporting their project. In particular, their [documentation](#) covers many aspects of advanced paged media and is a worthwhile next step after reading this guide.



HTML and CSS are beautiful languages and one of the landmark achievements of collaboration and the open web. Together, they allow you to create beautifully typeset digital content that can be viewed on any device, without any special software or paid licenses.

The web however was developed for scrolling content, and not paged media (by which we mean discrete pages, like in print or PDF). Unfortunately, whilst there are [W3C standards](#) on paged media, adoption by browsers has not been adequate to date. High quality paid alternatives exist in [PrinceXML](#), but a native, free solution has been missing.

A [happenstance discussion](#) on Hacker News brought my attention to [paged.js](#), a genius idea to polyfill the required capabilities to allow browsers to natively handle paged media, without having to build the whole rendering engine from scratch.

This guide provides a basic tutorial to using paged.js, for my own future reference and yours, and I highly recommend you visiting their site and supporting their project. In particular, their [documentation](#) covers many aspects of advanced paged media and is a worthwhile next step after reading this guide.

HTML and CSS are beautiful languages and one of the landmark achievements of collaboration and the open web. Together, they allow you to create beautifully typeset digital content that can be viewed on any device, without any special software or paid licenses.

The web however was developed for scrolling content, and not paged media (by which we mean discrete pages, like in print or PDF). Unfortunately, whilst there are [W3C standards](#) on paged media, adoption by browsers has not been adequate to date. High quality paid alternatives exist in [PrinceXML](#), but a native, free solution has been missing.

A [happenstance discussion](#) on Hacker News brought my attention to [paged.js](#), a genius idea to polyfill the required capabilities to allow browsers to natively handle paged media, without having to build the whole rendering engine from scratch.

This guide provides a basic tutorial to using paged.js, for my own future reference and yours, and I highly recommend you visiting their site and supporting their project. In particular, their [documentation](#) covers many aspects of advanced paged media and is a worthwhile next step after reading this guide.

