

# canons-fullwidth

## Full-Width Content for Classical Layouts

Version 1.2.0 (2025/10/05), LPPL 1.3c

Daniel Quigley

October 6, 2025

Engine	Version	Format	Date
LuaHB $\text{\TeX}$	1.21.0 (TeX Live 2025)	lualatex 2025.8.27	(03 Jan 2025 06:24)
pdf $\text{\TeX}$	3.141592653–2.6–1.40.27 (TeX Live 2025)	pdflatex 2025.8.27	(03 Jan 2025 06:25)
Xe $\text{\TeX}$	3.141592653–2.6–0.999997 (TeX Live 2025)	xelatex 2025.8.27	(03 Jan 2025 06:26)
pdf $\text{\TeX}$	3.141592653–2.6–1.40.27 (TeX Live 2025)	latex 2025.8.27	(03 Jan 2025 06:26)

**Table 1:** Engines and formats with which `canons-fullwidth` was tested.

### Abstract

`canons-fullwidth` provides environments for content that spans the text block plus margin area, following the design pattern popularized by Edward Tufte’s books and the `tufte-latex` document classes. The package integrates with `canons`, automatically adapting to all four margin modes (`symmetric`, `antisymmetric`, `right`, `left`), or defaults to the `textarea` with standard  $\text{\LaTeX}$  layouts. Unlike Tufte-style implementations, this package deliberately places captions above or below fullwidth content rather than in the margins, ensuring captions remain anchored to the text block for consistent reference alignment. The package provides both single-page (`fullwidth`) and multi-page (`fullwidth*`) variants with configurable vertical spacing, justification, and color. A comprehensive package details section provides information condensed for at-a-glance review.

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Design philosophy . . . . .	3
1.2	Key features . . . . .	4
1.3	What this package is not . . . . .	4
1.4	Quick start examples . . . . .	5
<b>2</b>	<b>Conceptual framework</b>	<b>5</b>
2.1	The fullwidth concept . . . . .	5
2.2	Direction and symmetry . . . . .	6
2.3	Caption placement philosophy . . . . .	6
2.4	Single-page versus multi-page . . . . .	7
<b>3</b>	<b>Installation and prerequisites</b>	<b>7</b>
3.1	Manual installation . . . . .	7
3.2	Dependencies . . . . .	8
3.2.1	Required packages . . . . .	8
3.2.2	Optional packages . . . . .	8
3.2.3	Known incompatibilities . . . . .	9
<b>4</b>	<b>Package options</b>	<b>9</b>
<b>5</b>	<b>Environment syntax</b>	<b>9</b>
5.1	The fullwidth environment . . . . .	9
5.2	The fullwidth* environment . . . . .	10
<b>6</b>	<b>Usage patterns</b>	<b>10</b>
6.1	Fullwidth figures . . . . .	10
6.2	Fullwidth tables . . . . .	11
6.3	Fullwidth text blocks . . . . .	12
6.4	Custom spacing and alignment . . . . .	12
<b>7</b>	<b>Integration with canons</b>	<b>13</b>
7.1	Automatic mode detection . . . . .	13
7.2	Dimension coordination . . . . .	13
7.3	Mode-specific behavior . . . . .	14

<b>8</b>	<b>Standalone usage</b>	<b>14</b>
<b>9</b>	<b>Common use</b>	<b>15</b>
9.1	Wide plot across text and margin . . . . .	15
9.2	Table needing extra columns . . . . .	15
9.3	Section divider (with color) . . . . .	15
9.4	Long table spanning pages and margins . . . . .	16
<b>10</b>	<b>Troubleshooting</b>	<b>17</b>
<b>11</b>	<b>Some remarks</b>	<b>19</b>
<b>A</b>	<b>Package details</b>	<b>22</b>

# 1 Introduction

Edward Tufte’s books on information design [5, 3, 6, 2, 4] employ a distinctive layout: extend the main text area into the margin space, which accommodates wide figures, tables, and text blocks. This creates visual hierarchy, accommodates complex graphics without scaling, and maintains the rhythm of the primary text column while providing occasional wide content. This behavior was likewise implemented in the excellent `tufte-latex` [1].

The `canons-fullwidth` package adopts this capability to documents using the `canons` page layout system, and degrades to a plain block behavior that never enters the margins when `canons` is not used. It coordinates with `canons`’ margin semantics to produce correct fullwidth behavior whether margins are on the right, left, outer edges, or inner edges.

Two environments (`fullwidth`, `fullwidth*`), three options (`skip`, `justification`, `color`). The defaults handle common cases; options provide precise control when needed.

## 1.1 Design philosophy

We adopt Tufte’s fullwidth concept: content spanning text plus margin. However, we deviate deliberately on caption placement: Tufte places captions in the margin, creating a floating reference system where captions sit beside their content. We

place captions above or below fullwidth material, anchored to the text block edge. This choice reflects our priorities: captions should align with the text column’s rhythm, not float in marginal space. For works using `canons-margins`’ sidenotes for commentary, keeping captions out of the margin preserves that space for annotations while maintaining consistent caption positioning.

Like `canons-margins`, this package detects and adapts to `canons`’ margin mode. A fullwidth figure in a `margins=right` layout extends rightward; the same code in `margins=symmetric` alternates direction by page. Change the canon, and fullwidth content adapts with no manual reconfiguration.

## 1.2 Key features

- Fullwidth content spanning text block plus margin area
- Automatic adaptation to all four `canons` margin modes
- Single-page (`fullwidth`) and multi-page (`fullwidth*`) variants
- Configurable vertical spacing, alignment, and color
- Caption-agnostic (use standard L<sup>A</sup>T<sub>E</sub>X caption commands)
- Lightweight implementation; minimal dependencies
- Works standalone without `canons` (defaults to text width)

## 1.3 What this package is not

`canons-fullwidth` is not a float package, despite its primary argument intending to be a float (though it can work with text); use standard `figure` and `table` environments and wrap fullwidth environments inside them.

The `tufte-latex` classes provide a comprehensive editorial system (sidenotes, margin figures, custom floats, running headers); this package provides only fullwidth environments, intended to be used with the `canons` package. If you want the complete Tufte idiom, use `tufte-book` or `tufte-handout`; if you want fullwidth capability with `canons`’ classical proportions, use this package.

## 1.4 Quick start examples

```
% Basic fullwidth figure
\begin{figure}[H]
  \begin{fullwidth}
    \includegraphics[width=\linewidth]{wide-plot.pdf}
  \end{fullwidth}
  \caption{A wide figure spanning text and margin}
\end{figure}

% Fullwidth with custom spacing and alignment
\begin{fullwidth}[skip=\baselineskip, justification=raggedright]
  Wide content that might break across pages if needed.
\end{fullwidth}

% Multi-page fullwidth table
\begin{table}
  \caption{Large dataset spanning text and margin}
  \begin{fullwidth*}
    \begin{tabular}{l}
      % ... many rows ...
    \end{tabular}
  \end{fullwidth*}
\end{table}

% With color for emphasis
\begin{fullwidth}[color=darkgray, justification=centered]
  \Large Important fullwidth announcement
\end{fullwidth}
```

## 2 Conceptual framework

### 2.1 The fullwidth concept

Standard L<sup>A</sup>T<sub>E</sub>X constrains content to `\textwidth`; wide material must either:

1. scale down to fit (losing detail);

2. rotate (disrupting reading flow);
3. overflow into margins (uncontrolled, inconsistent);
4. use landscape pages (breaking document structure).

Tufte demonstrated a fifth option: *deliberately* extend into the margin as a design feature, not a compromise. The text column remains constant; occasional fullwidth elements provide visual punctuation and accommodate complex graphics.

In `canons` terminology, fullwidth content spans `\textwidth + \fullwidthoverhang`, where `\fullwidthoverhang` equals `\marginparwidth + \marginparsep`. The exact dimension adapts to the chosen canon and margin mode.

## 2.2 Direction and symmetry

The challenge, then, follows: which direction does fullwidth content extend?

Margin mode	Extension direction	Page dependency
<code>right</code>	always rightward	none (all pages identical)
<code>left</code>	always leftward	none (all pages identical)
<code>symmetric</code>	rightward on recto, leftward on verso	yes (alternates)
<code>antisymmetric</code>	leftward on recto, rightward on verso	yes (alternates)

**Table 2:** Fullwidth extension by margin mode

The package handles this automatically via `\checkoddpage` and mode-specific horizontal shifts. User code remains identical across all modes; only the underlying page canon determines direction.

## 2.3 Caption placement philosophy

In Tufte’s books and the `tufte-latex` classes, captions (typically) sit in the margin beside their figures; this creates layouts where explanatory text floats near, but distinct from, visual content. The margin becomes an integrated commentary layer.

`canons-fullwidth` places captions in standard positions: above or below fullwidth content, anchored to the *text block* edge, not the fullwidth boundary; rationale for this follows:

1. **Consistency with text rhythm.** Captions align with the text column, maintaining the page’s vertical structure; fullwidth content extends from this anchor, but captions remain tied to the primary reading flow.
2. **Preserving margin space for sidenotes.** When using `canons-margins`, the margin hosts numbered annotations, cross-references, and commentary; caption text in the margin would compete for space and blur the distinction between figure label and margin note.
3. **Simplicity for users of `canons`.** Standard  $\text{\LaTeX}$  caption commands work without modification; no special caption environments, no margin-aware caption positioning, no conflicts with other caption-styling packages.
4. **Compatibility with journal workflows.** Many publishers expect captions in predictable positions, and courses that teach experiment and report design support this for floating objects; margin captions, while elegant, complicate expectations of standard captioning templates.

## 2.4 Single-page versus multi-page

`fullwidth` uses a `minipage` internally, and cannot break across pages; suitable for figures, tables, short text blocks, and provides precise control via calculated horizontal shifts.

`fullwidth*` uses `adjustwidth*` from `changepage`, which can span multiple pages, and so is suitable for long tables, extended quotations, appendices; slightly less precise due to implementation differences, but handles page breaks gracefully.

Choose `fullwidth` for floats and short content; `fullwidth*` for content that might exceed a page. Avoid pagebreaking floats, however, when possible.

## 3 Installation and prerequisites

### 3.1 Manual installation

1. Download `canons-fullwidth.sty` from CTAN

2. Place in your local texmf tree:

- Linux/Mac: ~/texmf/tex/latex/canons-fullwidth/
- Windows: C:\Users\[username]\texmf\tex\latex\canons-fullwidth\

3. Refresh the filename database:

```
texhash ~/texmf      # Linux/Mac  
miktex --update-fndb # Windows
```

## 3.2 Dependencies

### 3.2.1 Required packages

- caption : caption formatting and \captionof command
- adjustbox : internal calculations
- ifoddpage : page parity detection for two-sided modes
- changepage : adjustwidth\* environment for multi-page variant
- xpatch: internal patching
- pgfkeys : option processing
- ifthen : conditional processing
- xcolor : color support

### 3.2.2 Optional packages

- canons : page layout control and provides margin dimensions and mode detection (strongly recommended)
- canons-margins : compatible for margin notes alongside fullwidth content



### 3.2.3 Known incompatibilities

- `flowfram`, `parallel`, `paracol` : these packages manage page layout at a fundamental level and may conflict with margin placement;
- **`multicol`** : `fullwidth` in multi-column layouts is undefined;
- deeply nested minipages may confuse `\@totalleftmargin` calculations; further testing needed
- packages that heavily modify float internals may interact poorly; again, further testing needed

## 4 Package options

Unlike `canons` and `canons-margins`, this package has *no load-time options*. All configuration happens via environment options (see next section).

```
\usepackage{canons}           % Optional but recommended to use  
    marginspaces  
\usepackage{canons-fullwidth} % No options
```

## 5 Environment syntax

### 5.1 The fullwidth environment

```
\begin{fullwidth}[<options>]  
    % Content spanning text + margin  
\end{fullwidth}
```

Options.

**`skip=dimension`** Vertical space before environment. Default: `0.5\baselineskip`.

**`justification=align`** Content alignment. Values: `centering` (default), `raggedright`, `raggedleft`, `justified`.

**color=***color* Text color (requires color defined via `xcolor`). Default: empty (document text color).

**Behavior.**

- Creates a horizontal box of width `\textwidth + \fullwidthoverhang`
- Shifts left by appropriate amount based on margin mode and page parity
- Cannot break across pages
- Suitable for floats and short content

## 5.2 The `fullwidth*` environment

```
\begin{fullwidth*}[<options>]
  % Multi-page content spanning text + margin
\end{fullwidth*}
```

**Options.** Identical to `fullwidth`: `skip`, `justification`, `color`.

**Behavior.**

- Uses `adjustwidth*` for margin adjustment
- Can break across pages
- Suitable for long tables, extended text
- Slightly different horizontal positioning mechanism (inherent to `adjustwidth*`)

## 6 Usage patterns

### 6.1 Fullwidth figures

Standard pattern: wrap `fullwidth` environment inside a float.

```

\begin{figure}[t]
  \begin{fullwidth}
    \includegraphics[width=\linewidth]{example-image-a}
  \end{fullwidth}
  \caption{Example image A across text and margin}
  \label{fig:exampleA}
\end{figure}

```

Caption appears below the fullwidth content, aligned with text block. Use `\caption` before `fullwidth` for top-aligned captions:

```

\begin{figure}[t]
  \caption{Example image B across text and margin}
  \begin{fullwidth}
    \includegraphics[width=\linewidth]{example-image-b}
  \end{fullwidth}
  \label{fig:exampleB}
\end{figure}

```

## 6.2 Fullwidth tables

For tables fitting on one page:

```

\begin{table}[H]
  \caption{Comprehensive results requiring extra width}
  \begin{fullwidth}
    \begin{tabular}{llllllll}
      \toprule
      Col1 & Col2 & Col3 & Col4 & Col5 & Col6 & Col7 & \\
      \midrule
      % ... data ...
      \bottomrule
    \end{tabular}
  \end{fullwidth}
  \label{tab:results}
\end{table}

```

For tables spanning multiple pages (use `fullwidth*`):

```

\begin{table}
  \caption{Extended dataset spanning pages and margins}
  \begin{fullwidth*}
    \begin{longtable}{llllll}
      \toprule
      Col1 & Col2 & Col3 & Col4 & Col5 & Col6 \\
      \midrule
    \endfirsthead
    % ... many rows ...
    \bottomrule
    \end{longtable}
  \end{fullwidth*}
  \label{tab:long-data}
\end{table}

```

`longtable` inside `fullwidth*` requires careful handling; this has not been fully tested; proceed with caution.

### 6.3 Fullwidth text blocks

Extended quotations or special text:

```

\begin{fullwidth}[skip=\baselineskip, justification=justified]
  Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur
  felis ipsum, imperdiet non enim vitae, dapibus tristique ex.
  Phasellus hendrerit, mauris at molestie interdum, magna tortor
  mollis eros, sit amet finibus velit diam et lorem. Morbi ut
  dignissim velit. Nulla nec nisl imperdiet, pellentesque sem et,
  rhoncus lectus. Quisque nunc magna, viverra quis risus vitae,
  faucibus vestibulum arcu. Mauris eu arcu sit amet mauris vulputate
  ultricies. Etiam vel pretium neque.
\end{fullwidth}

```

### 6.4 Custom spacing and alignment

Announce section breaks or special content:

---

```

\begin{fullwidth}[skip=2\baselineskip, justification=centered, color=
  darkgray]
  \Large\scshape
  Interlude: A Brief Digression
\end{fullwidth}

```

## 7 Integration with canons

### 7.1 Automatic mode detection

When `canons` is loaded, `canons-fullwidth` queries `\pagecanonmargins` to determine the margin mode and configures `fullwidth` behavior accordingly. No user intervention required.

```

% Document uses right-margin layout
\usepackage[canon=tufte, margins=right]{canons}
\usepackage{canons-fullwidth}
% All fullwidth content extends rightward automatically

% Change to symmetric
\usepackage[canon=vdg, margins=symmetric]{canons}
\usepackage{canons-fullwidth}
% Fullwidth content now alternates direction by page

```

### 7.2 Dimension coordination

The package uses `\fullwidthoverhang` from `canons` to calculate the extension width. This dimension equals `\marginparwidth + \marginparsep`, ensuring `fullwidth` content precisely spans text plus margin notes area.

If `canons` is not loaded, `\fullwidthoverhang` is undefined and the package falls back to using just `\textwidth` (no extension).

## 7.3 Mode-specific behavior

Mode	Implementation	Page check	Extension
right	minipage shift	no	+ overhang rightward
left	minipage shift	no	+ overhang leftward
symmetric	minipage/adjustwidth*	yes	alternating
antisymmetric	minipage/adjustwidth*	yes	alternating (reversed)

**Table 3:** Implementation details by margin mode

## 8 Standalone usage

Without `canons`, the package provides basic fullwidth environments constrained to `\textwidth`; that is, objects are contained to the `\textwidth` only, regardless of presence of margins:

```
\documentclass{article}
\usepackage{canons-fullwidth} % No canons loaded

\begin{document}
% This creates a \textwidth minipage with centering, but no margin
  extension
\begin{fullwidth}
  Content constrained to text width
\end{fullwidth}
\end{document}
```

Essentially, standalone mode provides:

- environment syntax compatibility;
- `skip`, `justification`, `color` options;
- `minipage` behavior for `fullwidth`;
- simple paragraph for `fullwidth*`.

In this case, there is no margin extension, mode-aware behavior, integration with canonical proportions. Use standalone mode for compatibility during development, or when fullwidth is primarily a semantic marker rather than a layout feature.

## 9 Common use

### 9.1 Wide plot across text and margin

```
\begin{figure}[t]
  \begin{fullwidth}
    \includegraphics[width=\linewidth]{example-image-c}
  \end{fullwidth}
  \caption{Example image C}
\end{figure}
```

### 9.2 Table needing extra columns

```
\begin{table}[H]
  \caption{Table with wide space requirements}
  \begin{fullwidth}
    \small
    \begin{tabular}{@{}lcccccc@{}}
      \toprule
      Feature & Sys1 & Sys2 & Sys3 & Sys4 & Sys5 & Sys6 & \\
      \midrule
      % ... some data ...
      \bottomrule
    \end{tabular}
  \end{fullwidth}
\end{table}
```

### 9.3 Section divider (with color)

```
\begin{fullwidth}[skip=2\baselineskip, justification=centered, color=
mediumgray]
\Large\scshape Part II: Experimental Methods
\end{fullwidth}
```

## 9.4 Long table spanning pages and margins

Note that this is brittle. Refrain from spanning pages if possible.

```
\begin{fullwidth*}[justification=raggedright]
\begin{longtable}{@{}l11111l@{}}
\caption{Long and wide dataset} \\
\toprule
ID & Var1 & Var2 & Var3 & Var4 & Var5 \\
\midrule
\endfirsthead

\multicolumn{6}{c}{\tablename\ \thetable{} -- continued} \\
\toprule
ID & Var1 & Var2 & Var3 & Var4 & Var5 \\
\midrule
\endhead

\midrule
\multicolumn{6}{r}{Continued on next page} \\
\endfoot

\bottomrule
\endlastfoot

% ... many rows of data ...
\end{longtable}
\end{fullwidth*}
```



## 10 Troubleshooting

**Fullwidth content not extending into margin.** Fullwidth environment creates content at text width only; no margin extension. `canons` not loaded, so `\fullwidthoverhang` is undefined. Load `canons` before `canons-fullwidth`:

```
\usepackage{canons}           % Defines margin dimensions
\usepackage{canons-fullwidth} % Uses canons dimensions
```

**Content extending wrong direction.** In two-sided layout, fullwidth content extends the same direction on every page. Using `fullwidth` in outer scope where `\checkoddpage` cannot determine page parity, or page counter is wrong. The solution is a little involved, but first: **COMPILE TWICE**; this seems to resolve this issue. Additionally:

- use `fullwidth*` for content that must respect page parity across breaks;
- unsure content is typeset in normal document flow, not in pre-shipout hooks;
- check that page numbering is active and correct.

**Vertical spacing inconsistent.** Vertical space before fullwidth content varies unpredictably; default `skip=0.5\baselineskip` interacts poorly with surrounding spacing. Set explicit skip:

```
% No extra space
\begin{fullwidth}[skip=0pt]

% Full baselineskip
\begin{fullwidth}[skip=\baselineskip]
```

**Content overflow or underflow.** Content does not fit properly in fullwidth box; horizontal overflow or awkward gaps. Content sized for `\textwidth` when it should use `\linewidth` (which equals `\textwidth + \fullwidthoverhang` inside fullwidth). Check size content relative to `\linewidth`:

```
% Wrong:
\begin{fullwidth}
  \includegraphics[width=\textwidth]{example-image-a} % Too narrow
\end{fullwidth}

% Right:
\begin{fullwidth}
  \includegraphics[width=\linewidth]{example-image-a} % Uses full
    width
\end{fullwidth}
```

**Captions misaligned** This is an annoying but persistent issue, when captions do not align with text block as expected. Frequently, caption placed inside fullwidth environment rather than outside. Place caption outside fullwidth:

```
% Wrong:
\begin{figure}
  \begin{fullwidth}
    \includegraphics[width=\linewidth]{example-image-b}
    \caption{Example image B} % Inside fullwidth
  \end{fullwidth}
\end{figure}

% Right:
\begin{figure}
  \begin{fullwidth}
    \includegraphics[width=\linewidth]{example-image-b}
  \end{fullwidth}
  \caption{Example image B} % Outside fullwidth
\end{figure}
```

**Fullwidth\* breaks poorly.** Multi-page fullwidth\* content has awkward page breaks. `adjustwidth*` doesn't prevent breaks; L<sup>A</sup>T<sub>E</sub>X's page builder makes standard decisions. **AVOID MULTIPAGE FLOATS LIKE TABLES.**

- Use explicit `\pagebreak` before `fullwidth*` to control breaks

- Consider `longtable` inside `fullwidth*` for better table breaking
- Adjust content to fit page boundaries
- Use penalties: `\nopagebreak` before sensitive content

## 11 Some remarks

**Why not modify floats directly?** We could have created `figure*` and `table*` environments (like `multicol`'s starred floats). We chose not to because:

1. starred float syntax implies span columns in `multicol` context; using it for span to margin creates semantic confusion;
2. users may want `fullwidth` non-float content (quotations, section breaks);
3. keeping `fullwidth` as an inner environment preserves compatibility with arbitrary float packages;
4. clearer separation of concerns: floats handle positioning, `fullwidth` handles just that: width.

**Why two environments?** `fullwidth` and `fullwidth*` serve different use cases:

- `fullwidth`: precise control via calculated shifts; works well in floats; cannot break;
- `fullwidth*`: page-break capability; uses `adjustwidth*`; slight horizontal positioning differences.

We could have merged them with a `breakable` option, but keeping them separate makes the breaking behavior explicit at the call site. Additionally, page breaking floats and working with page breaking floats is fragile. Please refrain where possible.

**Caption placement controversy** This is essentially a style, design, and aesthetic decision. Our choice to keep captions outside fullwidth (anchored to text block) follows from:

1. maintaining typographic hierarchy;
2. compatibility and simplicity;
3. respecting margin space allocation.

Let us elaborate.

The text block is the primary reference frame. Captions identify and explain floating content; they are part of the document’s linear structure. Fullwidth content is an *extension* of normal figures/tables, not a different category. Captions should remain in the stable reference frame (text column) even when content expands beyond it.

Standard caption commands work without modification: no special handling for caption width; no mode-aware caption positioning; no conflicts with caption-styling packages. Publishers, collaborators, and journal templates expect captions in predictable locations.

When using `canons-margins`, margins host sidenotes, marginnotes, cross-references; caption text in margins competes for space and creates visual ambiguity: is this text a caption, a note, or a citation?

## References

- [1] Bil Kleb, Bill Wood, and Kevin Godby. *tufte-latex. Document classes inspired by the work of Edward Tufte*. Version 3.5.2. Repository: <https://tufte-latex.github.io/tufte-latex/>. CTAN. URL: <https://ctan.org/pkg/tufte-latex> (visited on 10/04/2025).
- [2] E.R. Tufte. *Beautiful Evidence*. Graphics Press, 2006. ISBN: 9780961392178.
- [3] E.R. Tufte. *Envisioning Information*. Graphics Press, 1990. ISBN: 9781930824140.
- [4] E.R. Tufte. *Seeing with Fresh Eyes: Meaning, Space, Data, Truth*. Graphics Press, 2020. ISBN: 9780961392192.
- [5] E.R. Tufte. *The Visual Display of Quantitative Information*. Graphics Press, 2001. ISBN: 9781930824133.

- [6] E.R. Tufte. *Visual Explanations: Images and Quantities, Evidence and Narrative*. Graphics Press, 1997. ISBN: 9781930824157.

# A Package details

## Environment summary

Environment	Breakable	Use case	Implementation
fullwidth	no	floats, short content	minipage with shift
fullwidth*	yes	long tables, multi-page text	adjustwidth*

## Option summary

Option	Default	Values
skip	0.5\baselineskip	any dimension
justification	centering	centering raggedright raggedleft justified
color	(empty)	any defined color name

## Margin mode behavior

Mode	Extension	Notes
right	always rightward	single-sided
left	always leftward	single-sided
symmetric	alternates (recto→R, verso→L)	two-sided
antisymmetric	alternates (recto→L, verso→R)	two-sided
(no canons)	none (text width only)	fallback

## Minimal recipes

```
% Basic figure
\begin{figure}
  \begin{fullwidth}
    \includegraphics[width=\linewidth]{example-image-a}
  \end{fullwidth}
  \caption{Example image A}
\end{figure}

% With custom options
\begin{fullwidth}[skip=\baselineskip, justification=raggedright, color=
  gray]
  Content here
```

```

\end{fullwidth}

% Multi-page table
\begin{fullwidth*}
  \begin{longtable}{llll}
    % ... table content ...
  \end{longtable}
\end{fullwidth*}

```

## Dimensions

Inside `fullwidth` environment:

- `\linewidth = \textwidth + \fullwidthoverhang` (use this for content width)
- `\textwidth` = main text column width (unchanged)
- `\fullwidthoverhang = \marginparwidth + \marginparsep`