# The incgraph package

Manual for version 1.4.2 (2026/06/25)

Thomas F. Sturm[1]

https://www.ctan.org/pkg/incgraph

https://github.com/T-F-S/incgraph

**Abstract**

incgraph provides tools for including graphics on full paper size. The graphics can be centered for a given paper format or the paper may be resized to the graphics dimensions. The main use case for the package incgraph is to transform one or many scans or taken pictures to a PDF document. It can also be applied for full paper size LaTeX created graphics. The package incgraph provides a tool box with basic macros and a convenience user interface which wraps the well-known includegraphics. Also, bookmarking is especially supported.

## Contents



[1]Prof. Dr. Dr. Thomas F. Sturm, Institut für Mathematik und Informatik, University of the Bundeswehr Munich, D-85577 Neubiberg, Germany; email: thomas.sturm@unibw.de

This documentation contains a lot of effects which can only be seen when viewed on a computer screen. If you read this text on paper, you will not notice the paper resizing and the bookmarks. Also, pages in landscape may probably be rotated by your printer.

# 1 Introduction

## 1.1 Motivation

The main purpose of this package is to include one or more graphics on full paper size. This means that a graphic is either centered on a blank page presumable of the given document paper size or the page is resized to the dimensions of the graphic. For the graphics, JPG files or PDF files or other supported formats may be used by inclusion. Alternatively, the graphics (or whatever) can be produced by LaTeX code. An important use case for the package `incgraph` is to transform one or many scans or taken pictures to a PDF document. Optionally, the included graphics can be commented with bookmarks for the resulting PDF document.

The well-known `graphicx` package [1] allows the inclusion of several types of external graphics files. The convenience user interface of `incgraph` described in Section 2 relies on this package and adds assistance for the described purpose. Note that the package is designed for and tested with `pdflatex` to produce PDF directly. Also, the package is intended to harmonize with `lualatex`. Some features like the paper resizing may not be applicable for other work-flows.

Many of the features of the convenience user interface can be used directly with various basic macros. These are collected and described as a "basic tool box" in Section 5.

If this package does not aid your intended purpose, you may take a look at the `pdfpages` package [2] which also supports the insertion of external multi-page PDF documents.

## 1.2 Loading the Package

`incgraph` is loaded in the usual manner in the preamble:

```
\usepackage{incgraph}
```

The package `incgraph` loads the package `pgfkeys` [4]. If no options are given, it also loads the packages `pgf`, `pgffor` [4], the package `graphicx` [1], and the package `bookmark` [3].

The following package options are available:

`pgf` (no value, initially set)
`nopgf` (no value, initially unset)

The option `nopgf` prevents the loading of `pgf` and `pgffor`.
The opposite option `pgf` resets to loading the packages.

`graphicx` (no value, initially set)
`nographicx` (no value, initially unset)

The option `nographicx` prevents the loading of `graphicx`.
The opposite option `graphicx` resets to loading the package.

`bookmark` (no value, initially set)
`nobookmark` (no value, initially unset)

The option `nobookmark` prevents the loading of `bookmark`.
The opposite option `bookmark` resets to loading the package.

So, the minimal package loading is done with the following:

```
\usepackage[nopgf,nographicx,nobookmark]{incgraph}
```

Note that you can always load the mentioned packages yourself. This is intended to avoid possible option clashes the easy way.

# 2 User Interface

## 2.1 Inclusion Macros for External Graphics

The macros of this section rely on the `\includegraphics` command from the package `graphicx` [1].

`\incgraph`[⟨*options*⟩][⟨*graphics options*⟩]{⟨*file name*⟩}

The picture file with the given ⟨*file name*⟩ is included in the center of a separate page. Depending on the ⟨*options*⟩, this page keeps the document size or is resized to the graphics dimensions. The applicable ⟨*options*⟩ are listed in Section 3. If ⟨*graphics options*⟩ are given, these are added to the options for the underlying `\includegraphics` command. See the documentation of `graphicx` [1] for a list of applicable ⟨*graphics options*⟩.

**Example 1: The hand-drawn example (centered); see page 20**

```
\incgraph[paper=current,label={exacenter},overlay page number at bottom,
  bookmark={The hand-drawn example (centered)}]{example.jpg}
```

**Example 2: The hand-drawn example (resized page); see page 21**

```
\incgraph[paper=graphics,label={exaresized},
  bookmark={The hand-drawn example (resized page)}]{example.jpg}
```

**Example 3: The hand-drawn example (rotated and oversized); see page 22**

```
\incgraph[paper=current,label={exarotated},target=oversized,
  bookmark={The hand-drawn example (rotated and oversized)}]%
  [angle=30,scale=3]{example.jpg}
```

`\incmultigraph`[⟨*options*⟩][⟨*graphics options*⟩]{⟨*file name pattern*⟩}{⟨*list*⟩}

All picture files matching the given ⟨*file name pattern*⟩ where some parts are substituted by elements of the ⟨*list*⟩ are included in the center of separate pages. Depending on the ⟨*options*⟩, the pages keep the document size or are resized to the graphics dimensions. The applicable ⟨*options*⟩ are listed in Section 3. If ⟨*graphics options*⟩ are given, these are added to the options for the underlying `\includegraphics` command. See the documentation of `graphicx` [1] for a list of applicable ⟨*graphics options*⟩.
The ⟨*list*⟩ may contain any construction allowed for the `\foreach` statement [4], especially a list of numbers. The elements of the list can be used inside the ⟨*file name pattern*⟩ with the following macros:

- `\n`: The current element of the list (may be a number).
- `\ni`: The position of the current element inside the list, i. e. `\ni` counts from 1 to the size of the list.
- `\nn`: The zero-filled `\n`, if `\n` is a number. The digit number of `\nn` is determined by `/igr/zerofill` → P. 19.

The resolved ⟨*file name pattern*⟩ is stored inside the macro:

- `\nt`: This file name may be used for bookmarking.

In the default behavior, non existing files are ignored.

**Example 4: A series of pictures; see from page 23. The image files exaimage-0001.png to exaimage-0150.png are included but only three of them exist.**

```
\incmultigraph[zerofill=4,bookmark={A series of pictures: \nt},
  paper=current,label={exaseries.\n}]{exaimage-\nn.png}{1,...,150}
```

N 2026-02-25 `\incmultipages[⟨options⟩][⟨graphics options⟩]{⟨file name⟩}{⟨page list⟩}`

This is mainly a specialized version of `\incmultigraph` → P. 5 for inclusion of one or several pages from a singular PDF file with given ⟨*file name*⟩. All pages corresonding to the given ⟨*page list*⟩ are included in the center of separate pages.

- Depending on the ⟨*options*⟩, the pages keep the document size or are resized to the graphics dimensions. The applicable ⟨*options*⟩ are listed in Section 3.
- If ⟨*graphics options*⟩ are given, these are added to the options for the underlying `\includegraphics` command. See the documentation of `graphicx` [1] for a list of applicable ⟨*graphics options*⟩. Note that the graphics option `page=\n` is set automatically.
- The ⟨*page list*⟩ may contain any construction allowed for the `\foreach` statement [4], basically a list of numbers. The page count of the PDF file is available as `\last` or alternatively `\igrLastPage`. For example:

| | |
|---|---|
| `{1,...,\last}` | All pages. |
| `{\last,...,1}` | All pages in reverse order. |
| `{2,4,...,\last}` | All even pages. |
| `{2,...,\inteval{\last-1}}` | All pages without the first and the last page. |
| `{\inteval{\last-4},...,\last}` | Last 5 pages. |
| `{1,...,5,17,40,41,...,\last}` | Pages 1,2,3,4,5,17,40,41,42,43 until last page. |

The elements of the list can be used for bookmarking with the following macros:

- `\n`: The current page of the list.
- `\ni`: The position of the current page inside the list, i. e. `\ni` counts from 1 to the size of the list.
- `\nn`: The zero-filled `\n`. The digit number of `\nn` is determined by `/igr/zerofill` → P. 19.

The ⟨*file name*⟩ is stored inside the macro:

- `\nt`: This file name may be used for bookmarking.

In the default behavior, non existing files are ignored. Also, non existing pages are ignored.

```
\incmultipages[
  paper    = a4,
  bookmark = {Imported page \n\ of \nt}
]
{incgraph-example-a.pdf}
{1,...,\last}
```

This example is continued in Section 4.3.

The same output can be generated with `\incmultigraph` → P. 5. Here, the page count has to be computed separately and the behavior in case of missing files or pages is different:

```
\igrGetLastPage{incgraph-example-a.pdf}
\incmultigraph[
  paper    = a4,
  bookmark = {Imported page \n\ of \nt}
]
[page=\n]
{incgraph-example-a.pdf}
{1,...,\igrLastPage}
```

## 2.2 Inclusion Macro for Internal Graphics

```
\begin{inctext}[⟨options⟩]
  ⟨environment content⟩
\end{inctext}
```

The environment content is included in the center of a separate page. Depending on the ⟨*options*⟩, this page keeps the document size or is resized to the content dimensions. The applicable ⟨*options*⟩ are listed in Section 3.

**Example 5: Some text on a shrunk paper; see page 26**

```
\begin{inctext}[paper=graphics,label={inctext1},bookmark={A huge ABC}]
  \fontsize{20mm}{20mm}\selectfont\bfseries ABC
\end{inctext}
```

**Example 6: A tikzpicture as text content; see page 27**

```
\begin{inctext}[paper=a6,landscape,label={inctext2},bookmark={Graph},
    overlay page number at bottom=8mm]
  \begin{tikzpicture}[zustand/.style={circle,fill=Gold,draw},scale=2]
  \draw node[zustand] (s0) {$s_0$}
    +(30:3cm) node[zustand] (s1) {$s_1$}
    ++(-30:3cm) node[zustand] (s2) {$s_2$}
    +(30:3cm) node[zustand] (s3) {$s_3$};
  \path[very thick,-latex]
    (s0) edge node[above left] {a} (s1)
    edge node[below left] {b} (s2)
    (s1) edge[out=-120,in=120] node[left] {b} (s2)
    edge node[above right] {a} (s3)
    (s2) edge[out=60,in=-60] node[right] {a} (s1)
    edge node[below right] {b} (s3);
  \end{tikzpicture}
\end{inctext}
```

## 2.3 (Global) Option Setting

```
\igrset{⟨options⟩}
```

Sets options for `\incgraph` →P. 5, `\incmultigraph` →P. 5, and `inctext` inside the current TEX group. For example, the options `/igr/paper` →P. 8 and `/igr/zerofill` →P. 19 may be defined for the whole document by this:

```
\igrset{paper=current,zerofill=3}
```

# 3 Option Keys

The package `incgraph` uses the key-value interface of the `pgfkeys` [4].

## 3.1 Paper (Media) Size

`/igr/currentpaper` (no value)

The paper size keeps unchanged at the current size except if `/igr/landscape` is used. The current paper size has not to be the document paper size. See page 20 for the output of Example 1 on page 5.

`/igr/documentpaper` (no value)

The paper size is set to the initial document paper size. If this option is applied inside the preamble, stored paper dimension may be incorrect (so, do not use inside the preamble).

`/igr/graphicspaper` (no value, initially set)

The paper is resized to the dimensions of the included image. The `/igr/landscape` option is ignored for this paper. See page 21 for the output of Example 2 on page 5.

`/igr/paper size=`⟨*width*⟩`:`⟨*height*⟩ (no default)

The paper is resized to the given ⟨*width*⟩ and ⟨*height*⟩.

`/igr/paper=`⟨*name*⟩ (no default)

The paper size is chosen according to the given ⟨*name*⟩. Feasible values for the ⟨*name*⟩ are `current`, `document`, `graphics`, `a0`, `a1`, `a2`, `a3`, `a4`, `a5`, `a6`, `a7`, `a8`, `a9`, `a10`, `b0`, `b1`, `b2`, `b3`, `b4`, `b5`, `b6`, `b7`, `b8`, `b9`, `b10`, `c0`, `c1`, `c2`, `c3`, `c4`, `c5`, `c6`, `c7`, `c8`, `c9`, `c10`, `d0`, `d1`, `d2`, `d3`, `d4`, `d5`, `d6`, `d7`, `executive`, `letter`, `legal`, `ledger`.

`/igr/landscape` (no value)

If set the width and height of the chosen paper are switched. Note that this turns the paper by 90 degrees but the contents of the paper is not turned.

`/igr/portrait` (no value, initially set)

Disables the `/igr/landscape` mode.

`/igr/a0paper` (no value)

The paper size is set to `841mm` × `1189mm`.

`/igr/a1paper` (no value)

The paper size is set to `594mm` × `841mm`.

`/igr/a2paper` (no value)

The paper size is set to `420mm` × `594mm`.

`/igr/a3paper` (no value)

The paper size is set to `297mm` × `420mm`.

`/igr/a4paper` (no value)

The paper size is set to `210mm` × `297mm`.

`/igr/a5paper` (no value)

The paper size is set to `148mm` × `210mm`.

`/igr/a6paper` (no value)

The paper size is set to `105mm` × `148mm`.

`/igr/a7paper` (no value)

The paper size is set to `74mm` × `105mm`.

/igr/a8paper (no value)

The paper size is set to 52mm × 74mm.

/igr/a9paper (no value)

The paper size is set to 37mm × 52mm.

/igr/a10paper (no value)

The paper size is set to 26mm × 37mm.

/igr/b0paper (no value)

The paper size is set to 1000mm × 1414mm.

/igr/b1paper (no value)

The paper size is set to 707mm × 1000mm.

/igr/b2paper (no value)

The paper size is set to 500mm × 707mm.

/igr/b3paper (no value)

The paper size is set to 353mm × 500mm.

/igr/b4paper (no value)

The paper size is set to 250mm × 353mm.

/igr/b5paper (no value)

The paper size is set to 176mm × 250mm.

/igr/b6paper (no value)

The paper size is set to 125mm × 176mm.

/igr/b7paper (no value)

The paper size is set to 88mm × 125mm.

/igr/b8paper (no value)

The paper size is set to 62mm × 88mm.

/igr/b9paper (no value)

The paper size is set to 44mm × 62mm.

/igr/b10paper (no value)

The paper size is set to 31mm × 44mm.

/igr/c0paper (no value)

The paper size is set to 917mm × 1297mm.

/igr/c1paper (no value)

The paper size is set to 648mm × 917mm.

/igr/c2paper (no value)

The paper size is set to 648mm × 917mm.

/igr/c3paper (no value)

The paper size is set to 324mm × 458mm.

/igr/c4paper (no value)

The paper size is set to 229mm × 324mm.

/igr/c5paper (no value)

The paper size is set to 162mm × 229mm.

/igr/c6paper (no value)

The paper size is set to 114mm × 162mm.

/igr/c7paper (no value)

The paper size is set to 81mm × 114mm.

/igr/c8paper (no value)

The paper size is set to 57mm × 81mm.

/igr/c9paper (no value)

The paper size is set to 40mm × 57mm.

/igr/c10paper (no value)

The paper size is set to 28mm × 40mm.

/igr/d0paper (no value)

The paper size is set to 771mm × 1091mm.

/igr/d1paper (no value)

The paper size is set to 545mm × 771mm.

/igr/d2paper (no value)

The paper size is set to 385mm × 545mm.

/igr/d3paper (no value)

The paper size is set to 272mm × 385mm.

/igr/d4paper (no value)

The paper size is set to 192mm × 272mm.

/igr/d5paper (no value)

The paper size is set to 136mm × 192mm.

/igr/d6paper (no value)

The paper size is set to 96mm × 136mm.

/igr/d7paper (no value)

The paper size is set to 68mm × 96mm.

/igr/executivepaper (no value)

The paper size is set to 7.25in × 10.5in.

/igr/letterpaper (no value)

The paper size is set to 8.5in × 11in.

/igr/legalpaper (no value)

The paper size is set to 8.5in × 14in.

/igr/ledgerpaper (no value)

The paper size is set to 17in × 11in.

## 3.2 Graphics Inclusion

`/igr/options={`⟨*graphics options*⟩`}` (no default, initially empty)

The ⟨*graphics options*⟩ are applied to the underlying `\includegraphics` command. See the documentation of `graphicx` [1] for a list of applicable ⟨*graphics options*⟩.

**Example 7: A resized image; see page 28**

```
\igrset{options={width=10cm,height=10cm}, paper=graphics,
  overlay page number at top=5mm
  }

\incgraph[bookmark={A resized image}, label={exagraphresize}]%
         {exaimage-0037.png}
```

`/igr/options add={`⟨*graphics options*⟩`}` (no default, initially empty)

The ⟨*graphics options*⟩ are added to the current list of options for the underlying `\includegraphics` command.

`/igr/include command=`⟨*macro*⟩ (default and initially `\includegraphics`)

Replaces the internally used `\includegraphics` command by the given ⟨*macro*⟩. Note that ⟨*macro*⟩ has to have the same signature as `\includegraphics`, i. e. it has to take two arguments where the first argument is optional.

`/igr/existence check=`⟨*macro*⟩ (no default)

Replaces the internally used `\IfFileExists` (in truth `\file_if_exist:nTF`) command by the given ⟨*macro*⟩. Note that ⟨*macro*⟩ has to have the same signature as `\IfFileExists`, i. e. it has to take three arguments.

N 2021-10-16

`/igr/no existence check` (no value)
`/igr/fail on not found` (no value)

Omits any existence checks by the package, i.e. the underlying `\includegraphics` is called directly and may stop the compilation with an error if the included file does not exist. Both options are equal, but the newer `/igr/no existence check` is considered to be more comprehensible.

`/igr/ignore on not found` (no value, initially set)

Not existing included files are ignored without warning. This is the default setting and is intended to be used with fragmentary file lists.

N 2021-10-16 **/igr/ignore on not found with extensions={⟨*extension list*⟩}** (initially unset)

N 2021-10-16 **/igr/ignore on not found with extensions*={⟨*extension list*⟩}** (initially unset)

Tests, if the included file with an added *extension* exists and performs the inclusion on existence. Otherwise, included files are ignored without warning. Tested *extensions* are taken from the optional comma separated ⟨*extension list*⟩ or (exclusively) from the configured list from /igr/extensions and friends (only, if ⟨*extension list*⟩ is not given).

/igr/ignore on not found with extensions* also tests, if the included file without added *extension* exists and takes it on existence.

Note that if a file is included with a given extension, this extension is not replaced by the listed extension, but prolonged with it.

These options are intended for mixed file lists, e.g. .jpg and .png, or unkown type file lists. Note that if you *know* the (singular) extension of your files, you are encouraged to use /igr/ignore on not found →P. 11 instead for performance reasons.

```
\incmultigraph[ignore on not found with extensions={.jpg,.png}]{myfile\n}{1,...,100}
  % includes e.g. `myfile1.jpg`, `myfile2.png`, `myfile42.jpg`, ...
\incmultigraph[ignore on not found with extensions={.jpg,.png}]{myfile\n.xy}{1,...,100}
  % includes e.g. `myfile1.xy.jpg`, `myfile2.xy.png`, `myfile42.xy.jpg`, ...
\incgraph[ignore on not found with extensions={.jpg,.png}]{myfile}
  % tests `myfile.jpg` and `myfile.png` for existence (and inclusion)
\incgraph[ignore on not found with extensions*={.jpg,.png}]{myfile}
  % tests `myfile`, `myfile.jpg`, and `myfile.png` for existence (and inclusion)
\incmultigraph[ignore on not found with extensions]{myfile\n}{1,...,100}
  % includes files with extensions given by /igr/extensions
```

N 2021-10-16 **/igr/extensions={⟨*extension list*⟩}** (initially .jpg,.png)

Specifies a comma separated ⟨*extension list*⟩ to be used in combination with /igr/ignore on not found with extensions.

N 2021-10-16 **/igr/extensions add={⟨*extension list*⟩}** (initially unset)

Adds a comma separated ⟨*extension list*⟩ to an already defined list from /igr/extensions.

N 2021-10-16 **/igr/extensions from graphics** (initially unset)

Takes the list of extensions from \DeclareGraphicsExtensions of the graphics [1] package as setup for /igr/extensions. While this looks like the most comfortable options, this can significantely increase compilation time when used for fragmentary file lists, because every non-existing file from the file list will induce a lot of existence checks. Also note that this options uses internals from graphics and is not guaranteed to work with future versions of graphics.

### 3.3 Hypertargets, Labels, and Bookmarks

`/igr/hyper` (no value, initially set)

An automated hyper target is set to the current image. The hyper target is placed at the top left corner of the page. It is used internally, when a bookmark is added.

`/igr/no hyper` (no value)

No automated hyper target is set to the current image. Use this option, if the package `bookmark` is not included.

`/igr/target=`⟨*anchor*⟩ (no default)

The next hypertarget destination value is set to ⟨*anchor*⟩ instead of an automatically created value. This may be used for hyperlinks.

```
\hyperlink{oversized}{This is linked to the oversized example (click me)}.
The target value '|oversized|' was defined in Example~\ref{exarotated.listing},
see page~\pageref{exarotated.listing}.
```

This is linked to the oversized example (click me). The target value '`oversized`' was defined in Example 3, see page 5.

`/igr/label`=⟨*text*⟩ (no default, initially empty)

Adds a LaTeX label to the included image.

`/igr/bookmark`=⟨*text*⟩ (no default, initially empty)

Adds a PDF bookmark with the given ⟨*text*⟩ to the current image.

`/igr/bookmark options`={⟨*bookmark options*⟩} (no default, initially empty)

Sets the options for a bookmark. See the documentation of `bookmark` [3] for a list of applicable ⟨*bookmark options*⟩.

**Example 8: Bookmark options; see page 29**

```
% not every PDF reader will show the effect!
\igrset{bookmark options={bold,color={red}},currentpaper}
\incgraph[bookmark={This ugly image again!},label={exabookmark}]%
         {example.jpg}
```

`/igr/bookmark heading`=⟨*text*⟩ (no default, initially empty)

For `\incmultigraph` → P. 5 and `\incmultipages` → P. 6, an additional bookmark with the given ⟨*text*⟩ is set as a heading before the images are included.

**Example 9: A series of pictures; see from page 30**

```
\incmultigraph[zerofill=4,currentpaper,
  bookmark heading={A series of pictures},
  bookmark heading options={level=subsection},
  bookmark={\nt},bookmark options={level=subsubsection},
  overlay page number at bottom,
  label={exaheading.\n}]{exaimage-\nn.png}{1,...,150}
```

`/igr/bookmark heading options`={⟨*bookmark options*⟩} (no default, initially empty)

Sets the options for a `/igr/bookmark heading`. See the documentation of `bookmark` [3] for a list of applicable ⟨*bookmark options*⟩.

### 3.4 Autosize

N 2024-03-09 **/igr/autosize=true|false** (default `true`, initially `false`)

If set to `true`, the picture box is resized to fit into the paper while keeping the aspect ratio. This setting has no effect for /igr/graphicspaper → P. 8. Note that this autosize is applied after possible size and scaling options for `\includegraphics` are conducted, see `\incgraph` → P. 5 and `\incmultigraph` → P. 5.

**Example 10: Autosized pictures; see from page 33**

```
\incmultigraph[zerofill=4, currentpaper, autosize,
  bookmark heading={Autosized pictures},
  bookmark heading options={level=subsection},
  bookmark={\nt},bookmark options={level=subsubsection},
  overlay page number at bottom,
  label={autosize.\n}]{exaimage-\nn.png}{1,...,150}
```

The following tuning options influence how the resizing is done.

N 2024-03-09 **/igr/autosize width gap=**⟨*length*⟩ (no default, initially `0mm`)

N 2024-03-09 **/igr/autosize height gap=**⟨*length*⟩ (no default, initially `0mm`)

N 2024-03-09 **/igr/autosize gap=**⟨*length*⟩ (no default, initially `0mm`)

⟨*length*⟩ is subtracted from the width or height of the current paper size to compute the target size for the automatic picture box scaling. /igr/autosize gap sets both values at the same time. For example, `autosize gap=20mm` gives (at least) `10mm` space around the rescaled picture box.

Negative ⟨*length*⟩ values allow the picture box to extrude over the borders of the current paper.

N 2024-03-09 **/igr/autosize min scale=**⟨*floating-point expression*⟩ (no default, initially `0.001`)

N 2024-03-09 **/igr/autosize max scale=**⟨*floating-point expression*⟩ (no default, initially `1000`)

The scale number of the automatic scaling will always be between /igr/autosize min scale and /igr/autosize max scale even if other settings are violated.

`autosize min scale=1` means that the picture box is never to be shrunk.

`autosize max scale=1` means that the picture box is never to be enlarged.

### 3.5 Borders and Shifts

The following settings enlarge or shrink the picture box, if /igr/graphicspaper → P. 8 is used. For other paper settings, the result will be a certain shift of the picture box since the enlarged box will be centered on the paper.

N 2015-03-12 U 2024-03-08 **/igr/left border**=⟨*length*⟩ (no default, initially 0pt)

Adds a space of ⟨*length*⟩ at the left hand side.

N 2015-03-12 U 2024-03-08 **/igr/bottom border**=⟨*length*⟩ (no default, initially 0pt)

Adds a space of ⟨*length*⟩ at the bottom.

N 2015-03-12 U 2024-03-08 **/igr/right border**=⟨*length*⟩ (no default, initially 0pt)

Adds a space of ⟨*length*⟩ at the right hand side.

N 2015-03-12 U 2024-03-08 **/igr/top border**=⟨*length*⟩ (no default, initially 0pt)

Adds a space of ⟨*length*⟩ at the top.

N 2015-03-12 U 2024-03-08 **/igr/horizontal border**=⟨*length*⟩ (no default, initially 0pt)

Adds a space of ⟨*length*⟩ at the left hand side and the right hand side.

N 2015-03-12 U 2024-03-08 **/igr/vertical border**=⟨*length*⟩ (no default, initially 0pt)

Adds a space of ⟨*length*⟩ at the top and bottom.

N 2015-03-12 U 2024-03-08 **/igr/border**=⟨*length*⟩ (no default, initially 0pt)

Adds a space of ⟨*length*⟩ at all four sides.

N 2024-03-08 **/igr/xshift**=⟨*length*⟩ (no default, initially 0pt)

Shifts the picture box in horizontal direction without changing the paper size. This adapts the current values of /igr/left border and /igr/right border. Therefore, use this option *after* the border values are set.

N 2024-03-08 **/igr/yshift**=⟨*length*⟩ (no default, initially 0pt)

Shifts the picture box in vertical direction without changing the paper size. This adapts the current values of /igr/top border and /igr/bottom border. Therefore, use this option *after* the border values are set.

### 3.6 Map and Match

`/igr/set matches={`⟨*list*⟩`}` (no default, initially empty)

The ⟨*list*⟩ is a comma separated list of ⟨*key*⟩=⟨*value*⟩ pairs. For every pair, the given ⟨*key*⟩ is mapped to the given ⟨*value*⟩. Later, this ⟨*value*⟩ can be retrieved by `/igr/if match code`, `/igr/if match set`, and `/igr/if match set bookmark`.

```
\igrset{set matches={
  foo = bar,
    1 = A very red image,
   37 = A not so centered number,
  123 = A greenish example}}
```

`/igr/if match code={`⟨*key*⟩`}{`⟨*then*⟩`}{`⟨*else*⟩`}` (no default)

If the ⟨*key*⟩ was defined by `/igr/set matches`, `\igrsetmatchvalue` → P. 54, or `\igrsetmatches` → P. 54, the corresponding value is put in the macro `\igrmatchvalue` and the ⟨*then*⟩ code is executed. If the ⟨*key*⟩ is unknown, the ⟨*else*⟩ code is executed.

`/igr/if match set={`⟨*key*⟩`}{`⟨*then*⟩`}{`⟨*else*⟩`}` (no default)

If the ⟨*key*⟩ was defined by `/igr/set matches`, `\igrsetmatchvalue` → P. 54, or `\igrsetmatches` → P. 54, the corresponding value is put in the macro `\igrmatchvalue` and `\igrset{`⟨*then*⟩`}` is executed. If the ⟨*key*⟩ is unknown, `\igrset{`⟨*else*⟩`}` is executed.

`/igr/if match set bookmark={`⟨*key*⟩`}{`⟨*then*⟩`}{`⟨*else*⟩`}` (no default)

If the ⟨*key*⟩ was defined by `/igr/set matches`, `\igrsetmatchvalue` → P. 54, or `\igrsetmatches` → P. 54, the corresponding value is put in the macro `\igrmatchvalue` and the current PDF bookmark is set to ⟨*then*⟩. If the ⟨*key*⟩ is unknown, the current PDF bookmark is set to ⟨*else*⟩.

**Example 11: Map and match example; see from page 36**

```
\incmultigraph[zerofill=4,paper=graphics,
  bookmark heading={Map and match example},
  bookmark heading options={level=subsection},
  bookmark options={level=subsubsection},
  if match set bookmark={\n}{\igrmatchvalue\ (\n)}{\nt},
  overlay page number at bottom,
  label={examatch.\n}]{exaimage-\nn.png}{1,...,150}
```

`/igr/disable match` (no value, initially set)

Disables the statements by `/igr/if match code`, `/igr/if match set`, and `/igr/if match set bookmark`.

### 3.7 Overlays

`/igr/overlay={`⟨*tikz code*⟩`}` (no default, initially unset)

Introduces arbitrary ⟨*tikz code*⟩ to be drawn over the included image. Note that the Ti*k*Z package [4] has to be loaded separately. To support positioning inside the picture, two `tikz` nodes named `box` and `page` are defined. `box` takes the dimensions of the included image and `page` takes the dimensions of the image or of the page depending on the usage of `/igr/paper` → P. 8.

**Example 12: Overlay; see page 39**

```
\igrset{bookmark options={level=subsection}, paper=current}
\incgraph[bookmark={Picture with overlay},label={overlay},
  overlay={
    \node[draw=red,line width=3pt,fill=red,fill opacity=0.1,
          minimum width=14cm,circle] (circ) at (page.center) {};
    \node[fill=blue!5!white,below right,text width=4cm] (A)
          at ([xshift=1cm,yshift=-1cm]page.north west)
          {This included image is overlayed with |tikz| code.};
    \node[fill=green!10!white,above,text width=7cm] (B)
          at ([yshift=2cm]page.south)
          {Image Name: \nt\\Page number: \thepage\\
           Example~\ref{overlay.listing} on page~\pageref{overlay.listing}};
    \draw[line width=2pt,->] (A)--(circ);
    \draw[line width=2pt,green!50!black,dashed]
         (box.south west)--(box.south east);
    \draw[line width=2pt,->,green!50!black] (B)--(box.south);
  }]{example.jpg}
```

`/igr/overlay page number at=`⟨*position*⟩ (no default, initially unset)

Overlays the page number at the given `tikz` ⟨*position*⟩.

`/igr/overlay page number at bottom=`⟨*length*⟩ (default `1.5cm`)

Overlays the page number at ⟨*length*⟩ above the bottom edge of the paper. See Example 1 on page 5 and the result on page 20.

`/igr/overlay page number at top=`⟨*length*⟩ (default `1.5cm`)

Overlays the page number at ⟨*length*⟩ below the top edge of the paper. See Example 7 on page 11 and the result on page 28.

N 2021-10-11 `/igr/overlay page number options={`⟨*tikz options*⟩`}` (no default, initially `fill opacity=0.35,draw opacity=0.5,text opacity=1,inner xsep=1em, rounded corners,fill=white,draw=black`)

Defines the Ti*k*Z node style for drawing the page number with one of the options above, e.g. `/igr/overlay page number at`

N 2021-10-21 `/igr/overlay page number options app={`⟨*tikz options*⟩`}` (no default, initially unset)

Appends the given ⟨*tikz options*⟩ to `/igr/overlay page number options`.

`/igr/no overlay` (no value, initially set)

Removes the overlay setting.

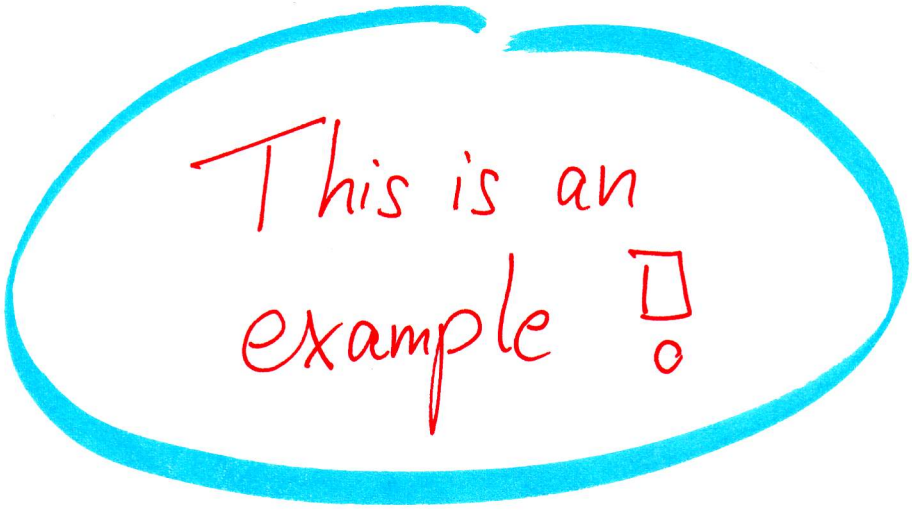
### 3.8 Miscellaneous

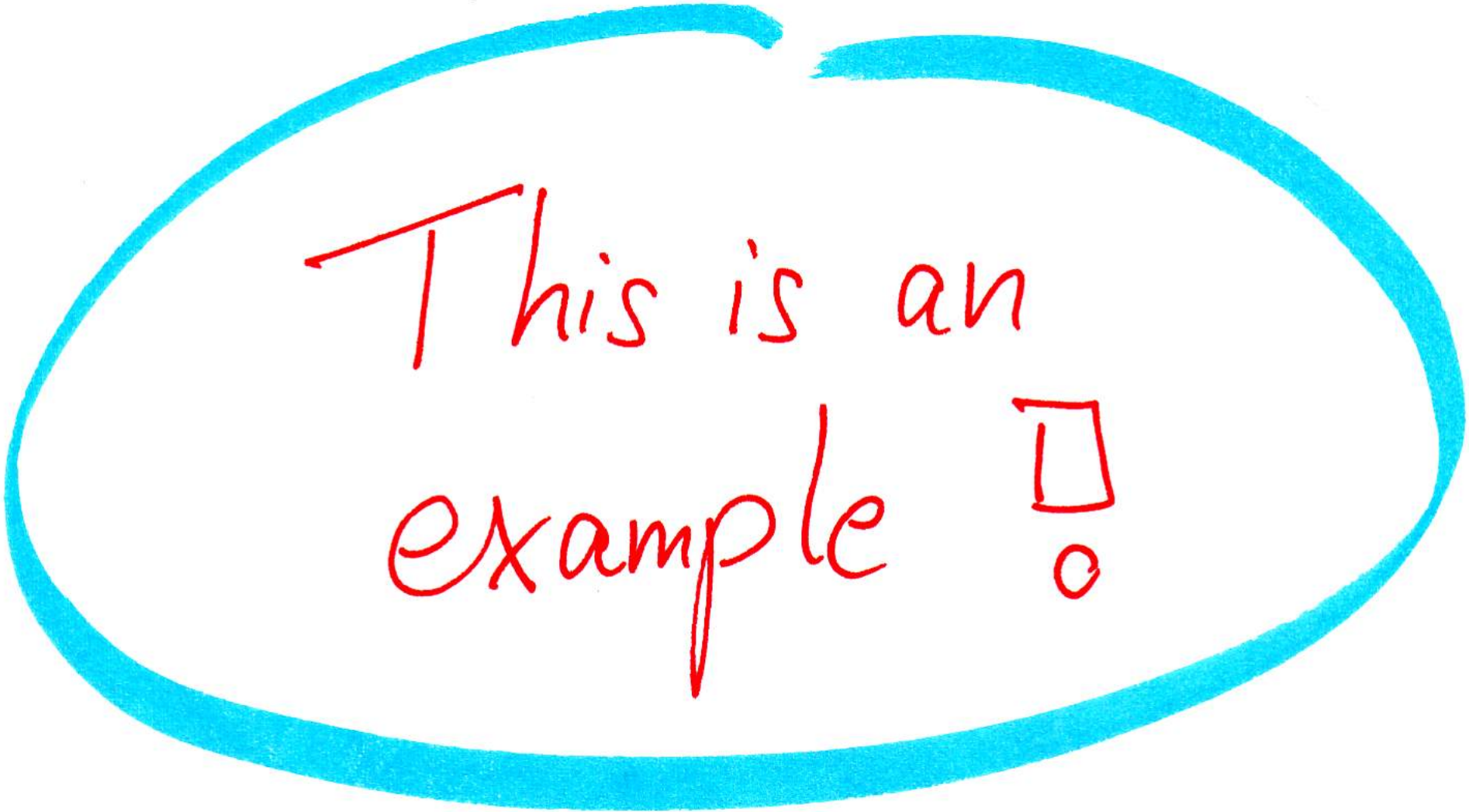
`/igr/pagestyle`=⟨*page style*⟩ (no default, initially `empty`)

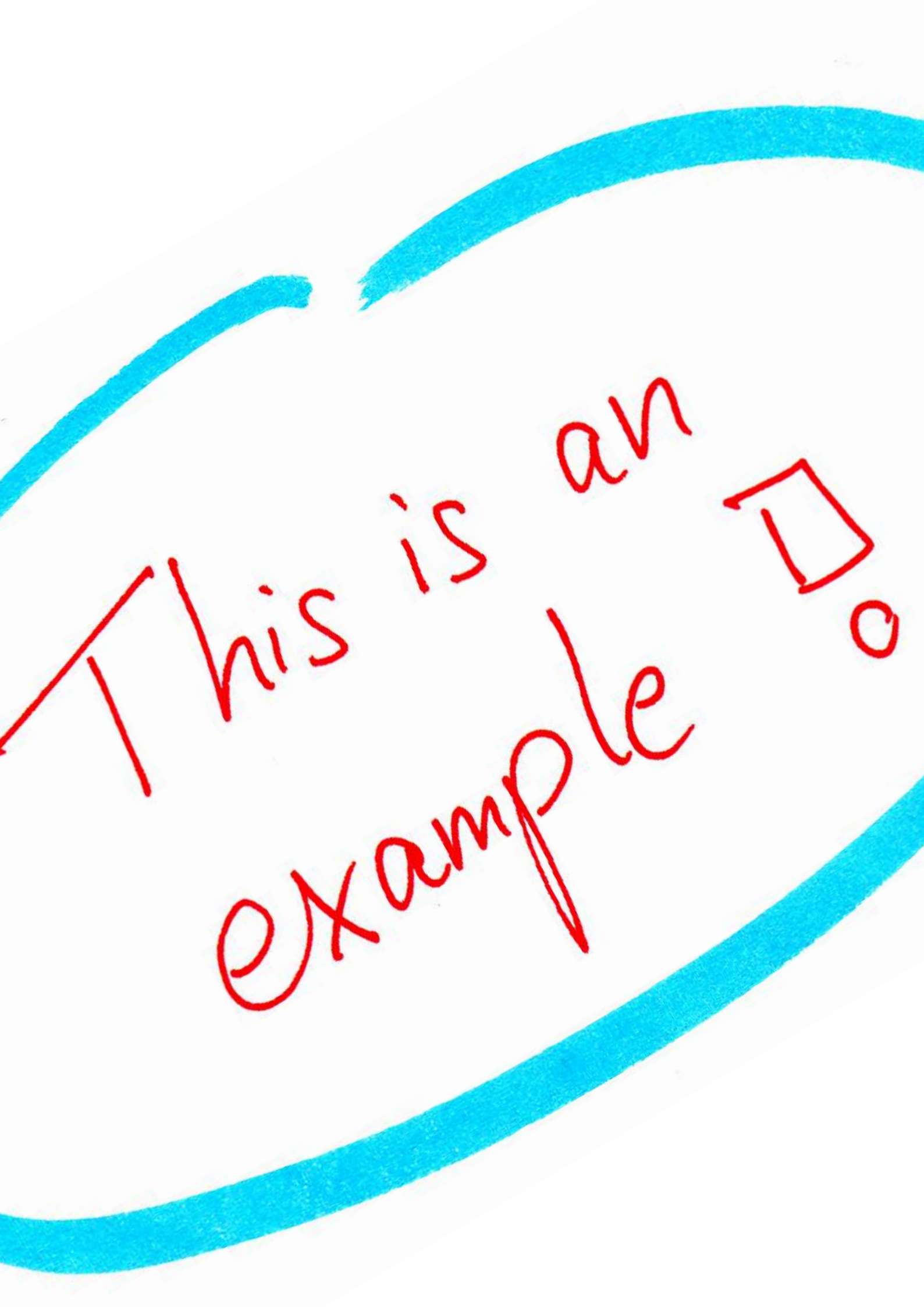
Sets the ⟨*page style*⟩ for the included graphics.

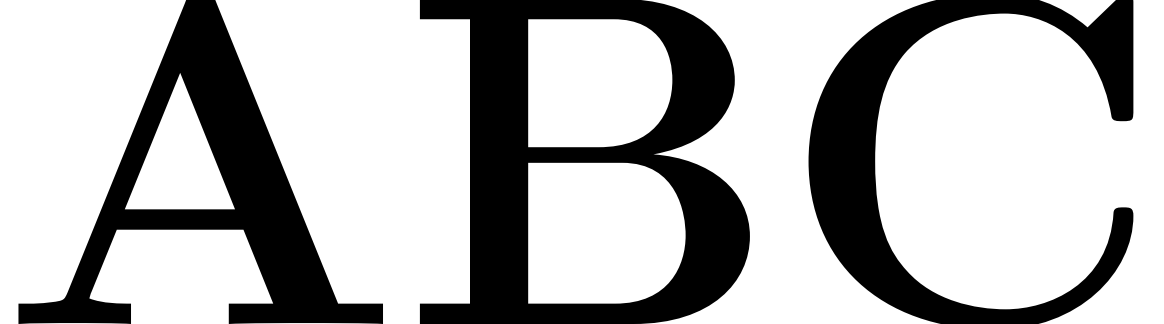
`/igr/zerofill`=⟨*digits*⟩ (no default, initially `0`)

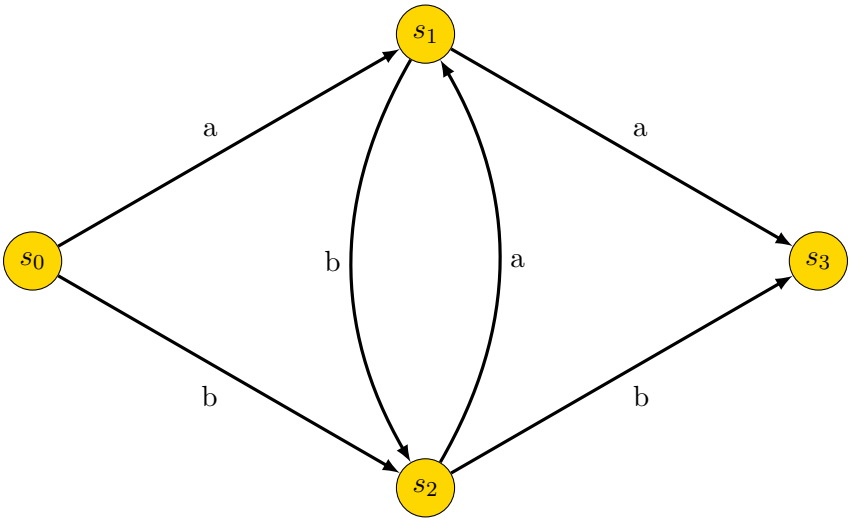
For `\incmultigraph` → P. 5, the current number element is filled up with leading zeros until the ⟨*digits*⟩ count is reached. If ⟨*digits*⟩ is 0 or 1, nothing is added. A ⟨*digits*⟩ value greater than 10 is treated as 10 which is the maximum number of possible digits. The result is accessible as `\nn`, see `\incmultigraph` → P. 5. Note that `zerofill` should be set to `0` if the list elements in `\incmultigraph` → P. 5 are not numbers.

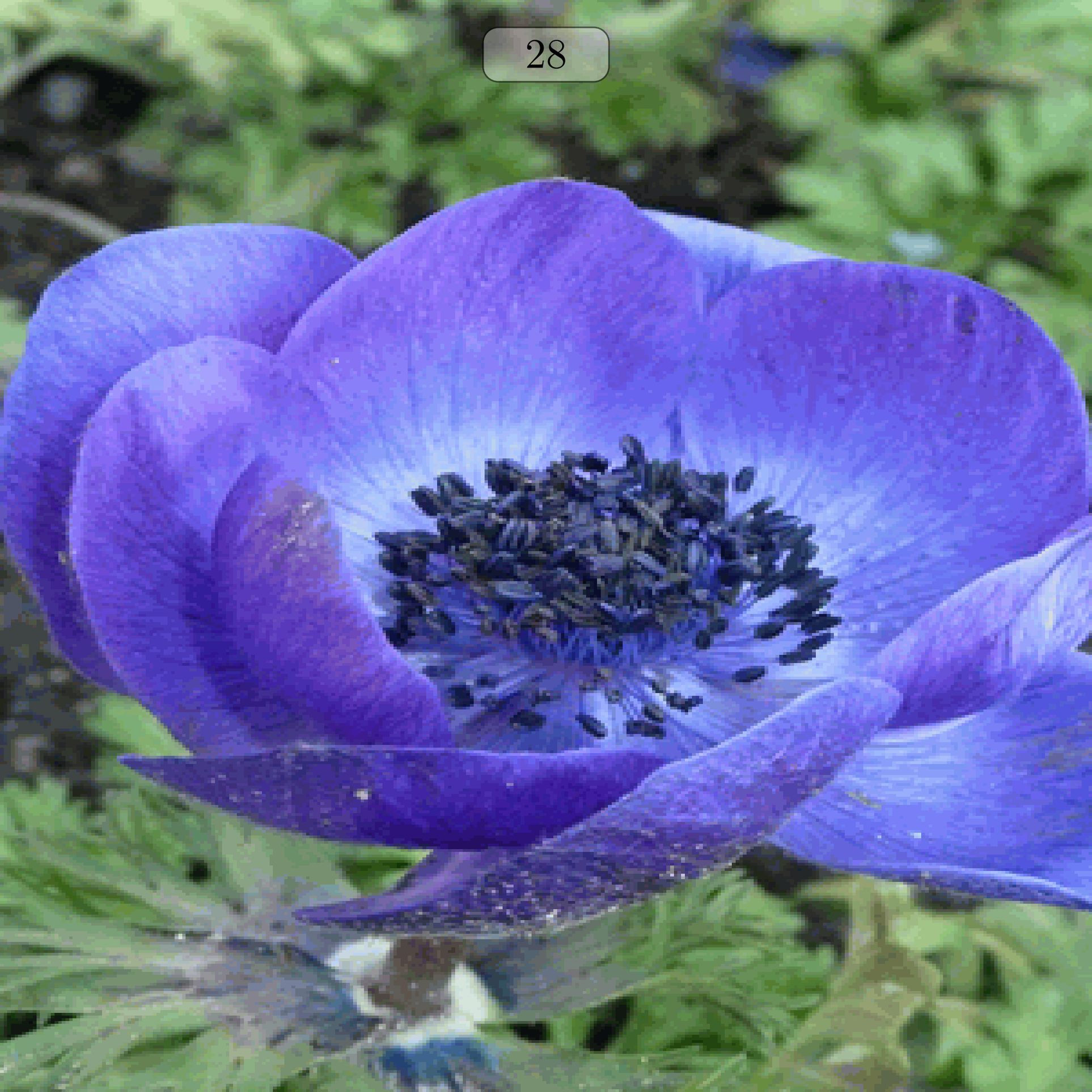

This is an
example !

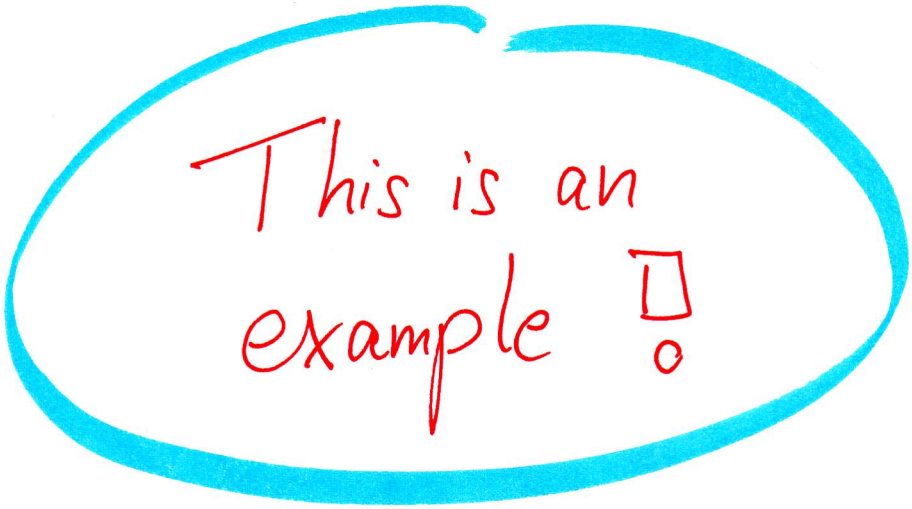

This is an
example !


This is an example □


ABC


$s_1$
a
a
$s_0$
b
a
$s_3$
b
b
$s_2$




This is an
example !

This included image is overlayed with `tikz` code.

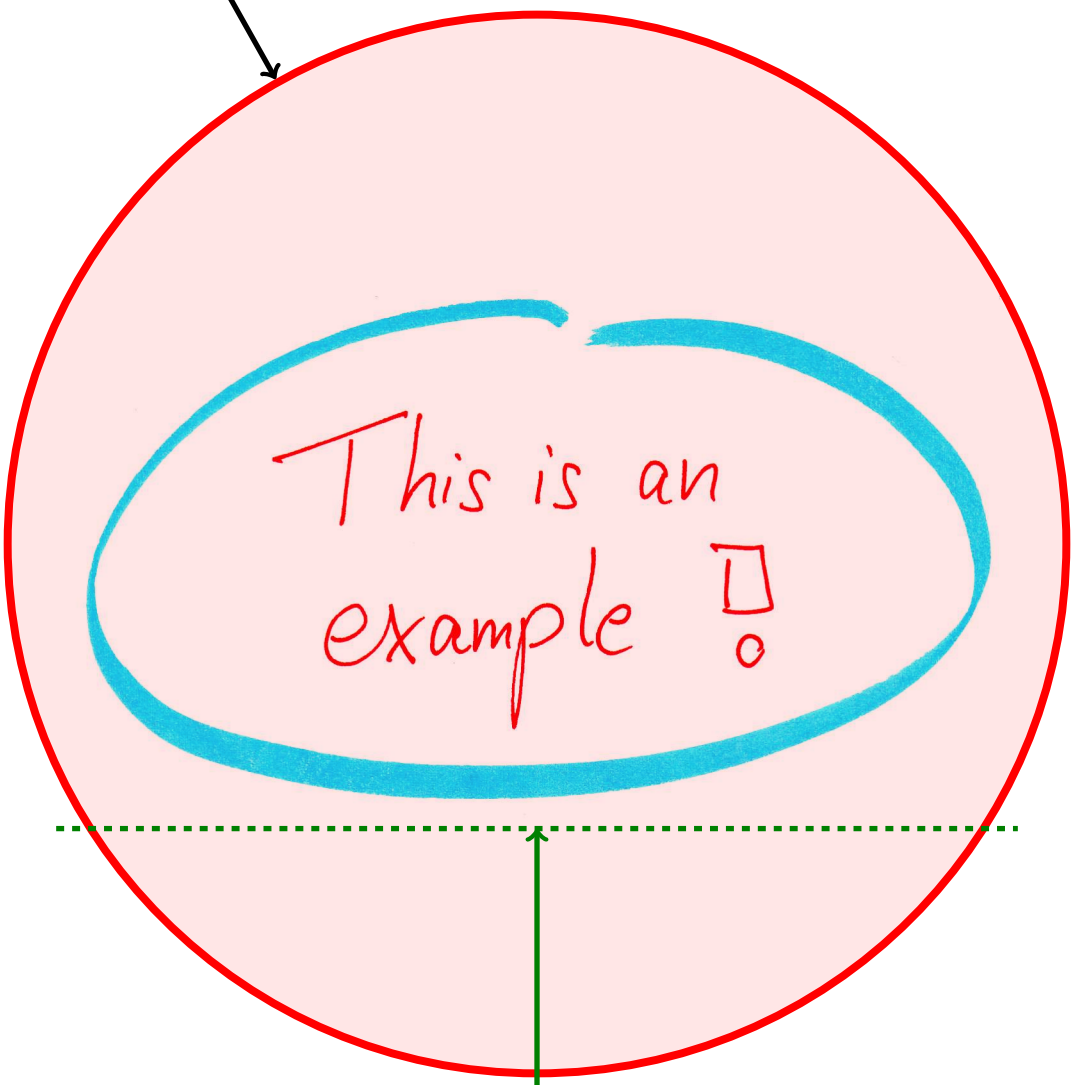



Image Name: example.jpg
Page number: 39
Example 12 on page 18

# 4 Examples

## 4.1 Including some Scans to Standard Paper

In this scenario, we have some scans (or images from whatever source) which should be combined to a PDF document for our paperless office. The paper size of the PDF document is set to a standard paper (here: letter size) to allow the document to be printed.

The following Example 13 is a complete template for such a document. Here, the images `example.jpg`, `exaimage-0001.png`, `exaimage-0037.png`, and `exaimage-0123.png` are used for the resulting document. All included images are automatically bookmarked with the page number and the file name of the source image.

**Example 13: `incgraph-example-a.tex`**

```
\documentclass{article}
\usepackage{incgraph}

\igrset{
  paper    = letter,
  zerofill = 4,
  bookmark = {Page \theigrpage\ (\nt)}
}

\begin{document}%
  \incgraph{example.jpg}
  \incmultigraph{exaimage-\nn.png}{1,...,150}
\end{document}
```

The compiled result of this stand-alone source code is not found in this document but as a separate file in the documentation directory of the package.

## 4.2 Creating a Picture Book

For this example, we assume again that a bunch of image files is to be combined to a PDF document. This time, the target document should be read or displayed mainly on computer screens and may never be printed. Therefore, the paper size is set flexible for the current image.

The following Example 14 is a complete template for such a document. All included images are resized to a common width, but this is not necessary. The resulting document is considered as an e-book where the bookmarks are the most important navigation accessory. Single page inclusions with `\incgraph` → P. 5 are bookmarked directly, but multi-page inclusions with `\incmultigraph` → P. 5 can be bookmarked using the map-and-match feature of the package. The example shows a mixed usage of the macros. Note that the bookmarks of the multi-page part are matched with the numbers contained in the file names of `exaimage-0001.png` to `exaimage-0150.png`.

**Example 14: `incgraph-example-b.tex`**

```
\documentclass{article}
\usepackage{incgraph,tikz}

\igrset{
  paper    = graphics,
  zerofill = 4,
  options  = {width=18cm},
  overlay page number at top = 7mm
}

\igrsetmatches{   1 = A very red image,
                123 = A greenish example }

\begin{document}

  \incgraph[bookmark={Page \theigrpage: My first example}]{example.jpg}

  \incmultigraph[
    if match set bookmark = {\n}
                            {Page \theigrpage: \igrmatchvalue}
                            {Page \theigrpage}
    ]
    {exaimage-\nn.png}{1,...,150}

\end{document}
```

The compiled result of this stand-alone source code is not found in this document but as a separate file in the documentation directory of the package.

### 4.3 Reformatting from Letter to DIN A4 (and vice versa)

In this scenario, we assume to have a PDF document with the 'wrong' paper size. Here, `incgraph-example-a.pdf` has the letter format, but A4 paper is needed. `incgraph` is used to reformat to the desired paper size. Of course, it also works the other way around.

The following Example 15 is a complete template for such a document. The document gets the desired paper size with the `/igr/paper` → P. 8 option. Then, all four pages of the original document are imported to the new paper size. Note that the actual document content itself is not resized because letter and DIN A4 are not so very different. If needed, the content could be shrunk or enlarged easily by adding a `scale` option for the underlying `\includegraphics` macro or by using `/igr/autosize` → P. 15.

**Example 15: `incgraph-example-c.tex`**

```
\documentclass{article}
\usepackage{incgraph}

\begin{document}%
  \incmultipages[
    paper    = a4,
    bookmark = {Imported page \n\ of \nt}
  ]
  {incgraph-example-a.pdf}
  {1,...,\last}
\end{document}
```

The compiled result of this stand-alone source code is not found in this document but as a separate file in the documentation directory of the package.

## 4.4 Drawing on Full Paper Size

In the following examples, no external image is included to the document. Instead, the image (or whatever) is created inside the document and put on a separate page which could be resized or take the original document paper size.

In Example 16, a `tikzpicture` is drawn. The whole picture is put inside an `inctext` → P. 7 environment which puts the drawing on a separate page which gets the dimensions of the drawing.

**Example 16: Creation of a special text page (resized)**

```
\begin{inctext}[paper=graphics, bookmark=My special text page (resized)]
\begin{tikzpicture}%
  \coordinate (A) at (0,0); \coordinate (B) at (16,16);
  \path[use as bounding box,top color=Goldenrod!25,bottom color=Navy!25]
       (A) rectangle (B);
  \coordinate (C) at ([xshift=1cm,yshift=1cm]A);
  \coordinate (D) at ([xshift=-1cm,yshift=-1cm]B);
  \path (C) -- coordinate (E) (D);
  \draw[rounded corners=5mm,very thick,Navy] (C) rectangle (D);
  \path (C) |-
    node [pos=0.75,fill=white,draw=Navy,very thick,inner sep=3mm]
      {My Special Page \thepage} (D);
  \node[text width=10cm,align=flush center,font=\Large] at (E) {
    This is my special page. It takes the dimensions of the underlying
    |tikzpicture| as seen in the source code of Example~\ref{fullpaperdrawing1}
    on page~\pageref{fullpaperdrawing1}.};
\end{tikzpicture}
\end{inctext}
```

See the result on the following page.

This is my special page. It takes the dimensions of the underlying `tikzpicture` as seen in the source code of Example 16 on page 43.

In Example 17, nearly the same `tikzpicture` is drawn. This time, the current paper size is used which puts the drawing on a separate page but without resizing the paper. To draw seamlessly, the document paper size of 21cm to 29.7cm is used directly inside the `tikzpicture`.

**Example 17: Creation of a special text page (fitted)**

```
\begin{inctext}[paper=current, target=mytarget,
  bookmark=My special text page (fitted)]
\begin{tikzpicture}%
  \coordinate (A) at (0,0); \coordinate (B) at (21,29.7);
  \path[use as bounding box,top color=Goldenrod!25,bottom color=Navy!25]
       (A) rectangle (B);
  \coordinate (C) at ([xshift=1cm,yshift=1cm]A);
  \coordinate (D) at ([xshift=-1cm,yshift=-1cm]B);
  \path (C) -- coordinate (E) (D);
  \draw[rounded corners=5mm,very thick,Navy] (C) rectangle (D);
  \path (C) |-
    node [pos=0.75,fill=white,draw=Navy,very thick,inner sep=3mm]
      {My Special Page \thepage} (D);
  \node[text width=10cm,align=flush center,font=\Large] at (E) {
    This is my special page. It consumes the whole document paper size with
    an underlying |tikzpicture| as seen in the source code of
    Example~\ref{fullpaperdrawing2} on page~\pageref{fullpaperdrawing2}.};
\end{tikzpicture}
\end{inctext}
```

See the result on the following page.

This is my special page. It consumes the whole document paper size with an underlying `tikzpicture` as seen in the source code of Example 17 on page 45.

# 5 Basic Tool Box

In this section, some basic macros of the package are documented. It is assumed that most users will only need the macros from the user interface described in Section 2 and Section 3.

## 5.1 Page Size Commands

The following macros are auxiliary tools which wrap corresponding length registers from `pdflatex` and `lualatex`.

N 2021-10-12 **\igrGetPageSize**

Reads the current page dimensions and stores them into the macros \igrPageWidth and \igrPageHeight.

N 2021-10-12 **\igrPageWidth**

After using \igrGetPageSize, \igrPageWidth contains the current page width. This is \pdfpagewidth for `pdflatex`, \pagewidth for `lualatex` and a choice of \pdfpagewidth or \paperwidth for other engines.

N 2021-10-12 **\igrPageHeight**

After using \igrGetPageSize, \igrPageHeight contains the current page height. This is \pdfpageheight for `pdflatex`, \pageheight for `lualatex` and a choice of \pdfpageheight or \paperheight for other engines.

N 2021-10-12 **\igrSetPageSize**{⟨*width*⟩}{⟨*height*⟩}

Sets the current page dimensions to ⟨*width*⟩ and ⟨*height*⟩. In depends on the used engine, if a call to this command has success or not. This should work for `pdflatex` and `lualatex`.

## 5.2 Last Page Commands (Number of Pages)

The following macros are auxiliary tools which used to wrap corresponding functions from `pdflatex` and `lualatex`. Now, they rely on the L3 programming layer.

N 2021-10-18
U 2026-02-25 **\igrGetLastPage**{⟨*file*⟩}

Reads the page count from the specified (PDF-)⟨*file*⟩ and stores it in the macro \igrLastPage.

Since 2026, this command has been implemented as a wrapper around **\graphics_get_pagecount:nN** from the L3 programming layer.

N 2021-10-18
U 2026-02-25 **\igrLastPage**

After calling \igrGetLastPage, the macro \igrLastPage contains the page count of the currently processed file.

Note that since 2026 the value of \igrLastPage is assigned locally; previously, it was assigned globally.

Also note that the page count is available alternatively inside \incmultipages → P. 6 as **\last**.

### 5.3 Full Page Commands

`\igrpage{`⟨*text*⟩`}`

The ⟨*text*⟩ is put on a separate page which is resized to fit the dimensions of the ⟨*text*⟩. ⟨*text*⟩ may be single letter, an included picture, or any LaTeX code. The page number is stored into `\theigrpage` and `\igrAutoTarget` holds a hyper target value for bookmarking. The style of the separate page is set to the content of the macro `\igrpagestyle` which defaults to 'empty' but can be redefined.

`\igrcenter{`⟨*text*⟩`}`

The ⟨*text*⟩ is put in the center of a separate page which has the current document dimensions. ⟨*text*⟩ may be single letter, an included picture or any LaTeX code. The page number is stored into `\theigrpage` and `\igrAutoTarget` holds a hyper target value for bookmarking. The style of the separate page is set to the content of the macro `\igrpagestyle` which defaults to 'empty' but can be redefined.

`\igrcenterfit{`⟨*width*⟩`}{`⟨*height*⟩`}{`⟨*text*⟩`}`

The ⟨*text*⟩ is put in the center of a separate page which is resized to the given ⟨*width*⟩ and ⟨*height*⟩. ⟨*text*⟩ may be single letter, an included picture or any LaTeX code. The page number is stored into `\theigrpage` and `\igrAutoTarget` holds a hyper target value for bookmarking. The style of the separate page is set to the content of the macro `\igrpagestyle` which defaults to 'empty' but can be redefined.

`\igrtargetset{`⟨*anchor*⟩`}`

The next value for `\igrAutoTarget` is set to ⟨*anchor*⟩. This can be used for hand-made hyperlinks or bookmarks. An application for `igrtargetset` is found in Example 17 on page 45.

## 5.4 Box Commands

`\igrboxset{`⟨*text*⟩`}`

The ⟨*text*⟩ is put into a TEX box named `\igrbox`. Additionally, some auxiliary macros are defined:

- `\igrAutoTarget`: unique value for a hyper target.
- `\igrBoxWidth`: width of the `\igrbox`.
- `\igrBoxHeight`: total height of the `\igrbox`.
- `\igrBoxht`: height of the `\igrbox`.
- `\igrBoxdp`: depth of the `\igrbox`.

```
\igrboxset{This is an example}
|\igrAutoTarget| = \igrAutoTarget, |\igrBoxWidth| = \igrBoxWidth,
|\igrBoxHeight| = \igrBoxHeight,\\
|\igrBoxht| = \igrBoxht, |\igrBoxdp| = \igrBoxdp;
```

`\igrAutoTarget` = igr-24, `\igrBoxWidth` = 76.42221pt, `\igrBoxHeight` = 7.95pt,
`\igrBoxht` = 6.2pt, `\igrBoxdp` = 1.75pt;

`\igrboxcenter`

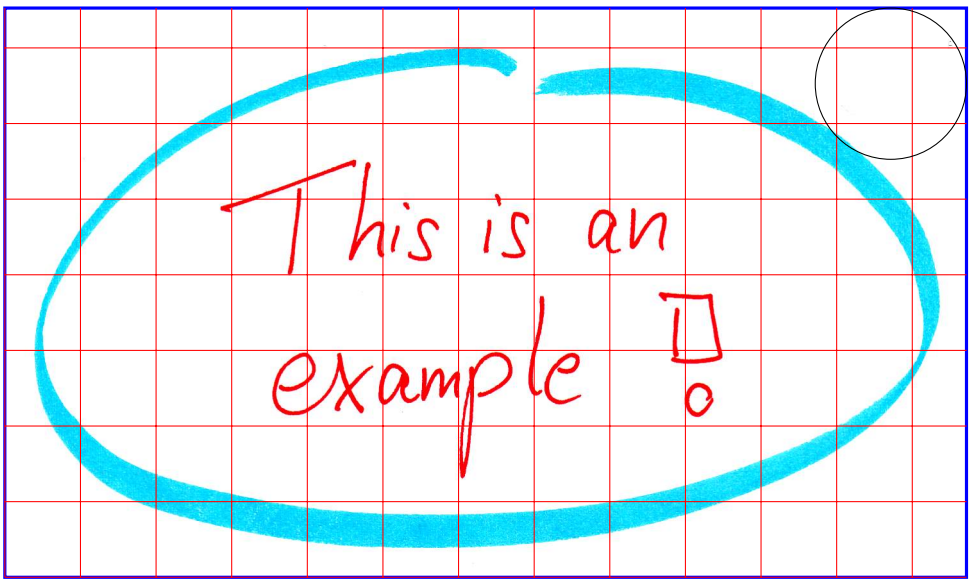
The current content of the `\igrbox` is put in the center of a separate page which has the current `pdfpage` dimensions.
The style of the separate page is set to the content of the macro `\igrpagestyle` which defaults to 'empty' but can be redefined. Note that a `\clearpage` or similar has to be inserted before this command.

### \igrboxtikz

The current content of the `\igrbox` is embedded into a `\node` command from the `tikz` package [4] which has to be loaded separately. Also, the bounding box is adjusted to the `\igrbox`.

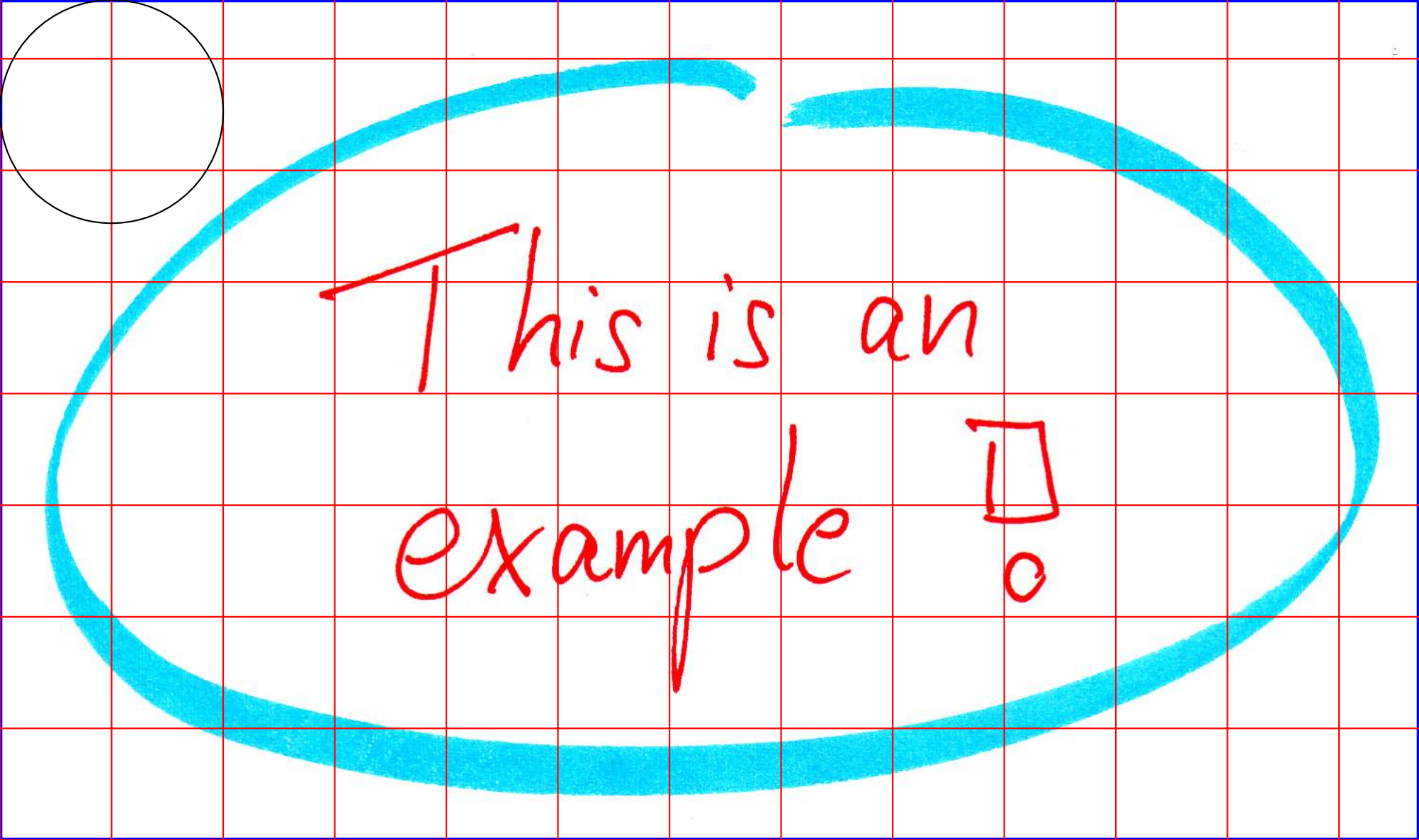
To support positioning inside the picture, two `tikz` nodes named `box` and `page` are defined which both take the dimensions of the `\igrbox`.

```
\igrboxset{\includegraphics{example.jpg}}%
\begin{tikzpicture}%
  \igrboxtikz%
  \draw[blue,very thick] (0,0) rectangle (\igrBoxWidth,\igrBoxHeight);
  \draw[red] (0,0) grid (\igrBoxWidth,\igrBoxHeight);
  \draw[black] ([xshift=-1cm,yshift=-1cm]page.north east) circle (1cm);
\end{tikzpicture}
```




The boxing macros can also be used nested (see the result on the following page):

```
\igrpage{\igrboxset{\includegraphics{example.jpg}}%
  \begin{tikzpicture}%
  \igrboxtikz%
  \draw[blue,very thick] (0,0) rectangle (\igrBoxWidth,\igrBoxHeight);
  \draw[red] (0,0) grid (\igrBoxWidth,\igrBoxHeight);
  \draw[black] ([xshift=1cm,yshift=-1cm]page.north west) circle (1cm);
\end{tikzpicture}}
```


This is an
example !

### \igrboxtikzpage

This is an alias for \igrboxtikz → P. 50.

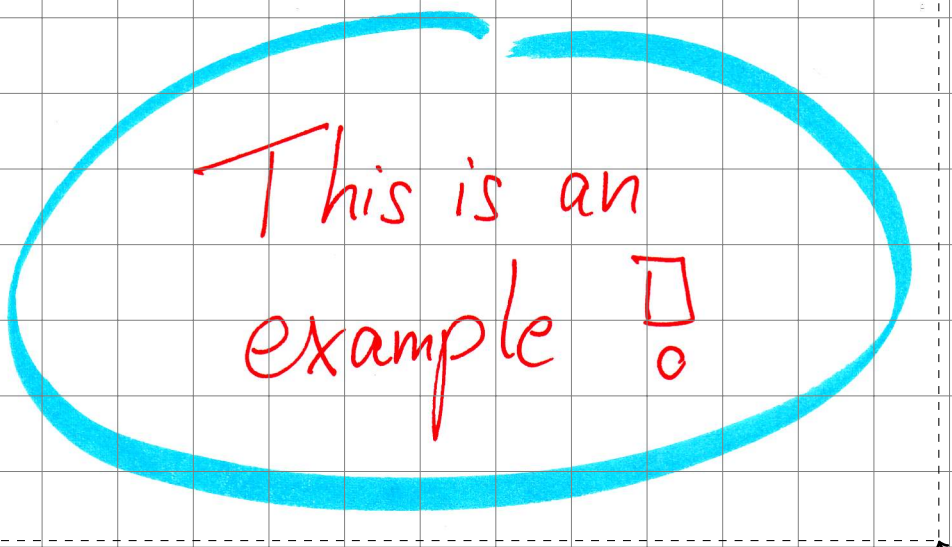
### \igrboxtikzcenter

The current content of the \igrbox is embedded into a \node command from the tikz package [4] which has to be loaded separately. This node is placed in the center of a bounding box which takes the current page dimensions. Afterwards, \igrBoxWidth and \igrBoxHeight are redefined to the dimensions of the total page.
To support positioning inside the picture, two tikz nodes named box and page are defined. box takes the dimensions of the \igrbox and page takes the dimensions of the tikzpicture.

```
\igrcenter{\igrboxset{\includegraphics{example.jpg}}%
  \begin{tikzpicture}%
  \igrboxtikzcenter%
  \draw[help lines] (0,0) grid (\igrBoxWidth,\igrBoxHeight);
  \draw[dashed] (box.south west) rectangle (box.north east);

  \draw[very thick,<->] (page.north west)--(box.north west);
  \draw[very thick,<->] (page.north east)--(box.north east);
  \draw[very thick,<->] (page.south west)--(box.south west);
  \draw[very thick,<->] (page.south east)--(box.south east);
\end{tikzpicture}}
```

See the result on the following page.


This is an
example

## 5.5 Map and Match Commands

**\igrsetmatchvalue**{⟨*key*⟩}{⟨*value*⟩}

The given ⟨*key*⟩ is mapped to the given ⟨*value*⟩. Later, this ⟨*value*⟩ can be retrieved by \igrifmatch.

```
\igrsetmatchvalue{my key A}{my value A}
\def\keytester#1{\igrifmatch{#1}{Hurray: '\igrmatchvalue'}{'#1' unknown}}

\keytester{something}\\
\keytester{my key A}
```

'something' unknown
Hurray: 'my value A'

**\igrsetmatches**{⟨*list*⟩}

The ⟨*list*⟩ is a comma separated list of ⟨*key*⟩=⟨*value*⟩ pairs. On every pair, the \igrsetmatchvalue macro is applied.

```
\igrsetmatches{my key A = my value A, bar = Shakespeare}
\def\keytester#1{\igrifmatch{#1}{Hurray: '\igrmatchvalue'}{'#1' unknown}}

\keytester{something}\\
\keytester{bar}\\
\keytester{my key A}
```

'something' unknown
Hurray: 'Shakespeare'
Hurray: 'my value A'

**\igrifmatch**{⟨*key*⟩}{⟨*then*⟩}{⟨*else*⟩}

If a ⟨*key*⟩ was defined by \igrsetmatchvalue or \igrsetmatches, the corresponding value is put in the macro **\igrmatchvalue** and the ⟨*then*⟩ code is executed. If the ⟨*key*⟩ is unknown, the ⟨*else*⟩ code is executed.

```
\igrsetmatches{1 = January, 2 = February, 3 = March, apr = April}
\def\monthname#1{\igrifmatch{#1}{The name of month #1\ is \igrmatchvalue.}{%
  You are kidding.}}

\monthname{1} \monthname{foo} \monthname{2}\\
\monthname{3} \monthname{apr} \monthname{35}
```

The name of month 1 is January. The name of month foo is bar. The name of month 2 is February.
The name of month 3 is March. The name of month apr is April. You are kidding.

## 5.6 Zero Filling Commands

`\igrmakezerofill{`⟨*macro*⟩`}{`⟨*digits*⟩`}`

With this command, a new ⟨*macro*⟩ can be defined which takes a non negative number as parameter. This number is filled up with leading zeros until the ⟨*digits*⟩ count is reached. If ⟨*digits*⟩ is 0 or 1, nothing is added. A ⟨*digits*⟩ value greater than 10 is treated as 10 which is the maximum number of possible digits.

```
\igrmakezerofill{\myfill}{0}
\myfill{7}, \myfill{12}, \myfill{934}, \myfill{665234}.\\
\igrmakezerofill{\myfill}{3}
\myfill{7}, \myfill{12}, \myfill{934}, \myfill{665234}.\\
\igrmakezerofill{\myfill}{5}
\myfill{7}, \myfill{12}, \myfill{934}, \myfill{665234}.\\
\igrmakezerofill{\myfill}{9}
\myfill{7}, \myfill{12}, \myfill{934}, \myfill{665234}.\\
\igrmakezerofill{\myfill}{30}
\myfill{7}, \myfill{12}, \myfill{934}, \myfill{665234}.
```

7, 12, 934, 665234.
007, 012, 934, 665234.
00007, 00012, 00934, 665234.
000000007, 000000012, 000000934, 000665234.
0000000007, 0000000012, 0000000934, 0000665234.

```
\igrmakezerofill{\threedigits}{3}
\threedigits{1}%
\foreach \n in {2,...,100} {, \threedigits{\n}}.
```

001, 002, 003, 004, 005, 006, 007, 008, 009, 010, 011, 012, 013, 014, 015, 016, 017, 018, 019, 020, 021, 022, 023, 024, 025, 026, 027, 028, 029, 030, 031, 032, 033, 034, 035, 036, 037, 038, 039, 040, 041, 042, 043, 044, 045, 046, 047, 048, 049, 050, 051, 052, 053, 054, 055, 056, 057, 058, 059, 060, 061, 062, 063, 064, 065, 066, 067, 068, 069, 070, 071, 072, 073, 074, 075, 076, 077, 078, 079, 080, 081, 082, 083, 084, 085, 086, 087, 088, 089, 090, 091, 092, 093, 094, 095, 096, 097, 098, 099, 100.

## References


[1] David Carlisle and The LaTeX Project Team. *Packages in the graphics bundle*. Version 1.2e. Comprehensive TeX Archive Network. May 17, 2026.
https://ctan.org/pkg/graphicx.

[2] Andreas Matthias. *The pdfpages Package*. Version 0.6h. Comprehensive TeX Archive Network. Mar. 30, 2026.
https://ctan.org/pkg/pdfpages.

[3] Heiko Oberdiek. *The bookmark Package*. Version 1.32. Comprehensive TeX Archive Network. May 21, 2026.
https://ctan.org/pkg/bookmark.

[4] Till Tantau, Christian Feuersänger, Henri Menke, and The PGF/TikZ Team. *The TikZ and PGF Packages*. Version 3.1.11a. Comprehensive TeX Archive Network. Aug. 29, 2025.
https://ctan.org/pkg/pgf.

# Index