Example of the `xkeyval` package for package writers

J. op den Brouw*

July 17, 2021

This simple test package is conceived to demonstrate the use of *key/value* option processing for packages and macros. The test package can be loaded with:

```
\usepackage[⟨options⟩]{xkeyval-package}
```

where *options* is a list of options:

`parindent`=*skip*, default 0pt
`color`=*color*, default is black
`style`=*up*|*sl*|*sc*|*it*, default is up
`showindent`=*true*|*false*, default is false

As an example:

```
\usepackage[parindent=15pt,color=blue,style=sc,showindent]{xkeyval-package}
```

The package has one macro:

```
\mybox[⟨options⟩]{ text }
```

that typesets some simple text. The options are the same as the package options. The package options serve as the default for the macro, but can be overridden with per-macro invocations. As an example:

```
\mybox[parindent=15pt,style=sc,color=blue,showindent=false]{\blindtext}
```

Best is to examine the package file in detail.

*The Hague University of Applied Sciences, J.E.J.opdenBrouw@hhs.nl

Some text before ... `parindent` = 0.0pt

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

Some text after ... `parindent` = 0.0pt

```
%%
%% xkeyval-package.sty
%%
%% Xkeyval Test Package to demonstrate xkeyval
%%
%% (c)2021, Jesse op den Brouw <J.E.J.opdenBrouw@hhs.nl>
%%
%% License: LPPL, https://www.latex-project.org/lppl/

% Credentials
\def\fileversion{0.1}
\def\filedate{2021/07/09}

% TeX format needed
\NeedsTeXFormat{LaTeX2e}[1995/06/01]

% Provides package
\ProvidesPackage{xkeyval-package}[\filedate\space Version
                           \fileversion\space xkeyval Test Package]

% Read in xkeyval
\RequirePackage{xkeyval}
% Read in xcolor
\RequirePackage{xcolor}

% Text parindent length from package and macro options
\newlength{\package@parindent}
\setlength{\package@parindent}{0pt}
\newlength{\macro@parindent}
\setlength{\macro@parindent}{0pt}

% Text color from package and macro options
\def\package@textcolor{black}
\def\macro@textcolor{black}

% The selected style as a number
\def\package@stylenr{0}
\def\macro@stylenr{0}

% The showindent options as booleans
\newif\ifpackage@showindent
\newif\ifmacro@showindent


% Declare package options
% The package parindent option
\DeclareOptionX{parindent}[15pt]{\setlength{\package@parindent}{#1}}

% The package color option
\DeclareOptionX{color}[black]{\def\package@textcolor{#1}}

% The package text style as a number
% Text style is one of up (upright), sc (small caps), it (italic) or sl
% (slanted). The style is saved as a number: up=0, sc=1, it=2, sl=3.
% Note that \DeclareOptionX wil not work for choice keys, so we use a
```

```
% nice trick by defining a choice key with the package name as family
\define@choicekey{xkeyval-package.sty}{style}[\val\temp@stylenr]{up,sc,it,sl}
                          {\def\package@stylenr{\temp@stylenr}}

% Boolean for showing a visual parindent, again with a trick as it is
% not possible to use DeclareOptionX for boolean keys.
\define@boolkey{xkeyval-package.sty}{showindent}[true]{
    % Now we have to set the macro \package@showindent to false or true
    % but that's is a bit complicated, because xkeyval only sets
    % \ifKV@xkeyval-package.sty@showindent to false or true. Note the
    % - and . in the macro name. These characters are not letters so
    % we can't use them in simple macro names. We have to use the
    % \csname ... \endcsname construction but that does not work within
    % \define@boolkey. So we do that later on (about 10 lines below)
}


% Print out unrecognized options ...
\DeclareOptionX*{\PackageError{xkeyval-package}{Invalid option '\CurrentOption'}}

% Process the options
\ProcessOptionsX\relax

% Process the showindent package key, note that the - and the .
% are not letters so we must use a csname construction
\csname ifKV@xkeyval-package.sty@showindent\endcsname
    \package@showindenttrue
\fi


% Define keys for the macro mybox (famlily name)
% Text parindent is a skip (dimension)
\define@key{mybox}{parindent}[\package@parindent]{\setlength\macro@parindent{#1}}

% Text color is a color defined by xcolor
\define@key{mybox}{color}[\package@textcolor]{\def\macro@textcolor{#1}}

% Text style is one of up (upright), sc (small caps), it (italic) or sl (slanted)
% The style is saved as a number: up=0, sc=1, it=2, sl=3
\define@choicekey{mybox}{style}[\val\temp@stylenr]{up,sc,it,sl}
                                          {\def\macro@stylenr{\temp@stylenr}}

% Boolean for showing a visual parindent, as a boolean key.
\define@boolkey{mybox}{showindent}[true]{%
    \ifKV@mybox@showindent
        \macro@showindenttrue
    \else
        \macro@showindentfalse
    \fi
}


% The command to test
\newcommand\mybox[2][]{%
\begingroup% Start a new group
```

```
    % Inherit from package options
    \setlength{\macro@parindent}{\package@parindent}%
    \edef\macro@textcolor{\package@textcolor}%
    \edef\macro@stylenr{\package@stylenr}
    \macro@showindentfalse
    \ifpackage@showindent\macro@showindenttrue\fi
    % Parse keys
    \setkeys{mybox}{#1}%
    % Set indent and color
    \setlength{\parindent}{\macro@parindent}\color{\macro@textcolor}%
    % Create a visual bucket for the parindent skip
    \ifmacro@showindent%
      \begingroup% Start a new group
        \hspace{-\parindent}\rule{0.4pt}{1ex}\hspace{-0.4pt}%
                      \rule{\parindent}{0.4pt}\hspace{-0.4pt}\rule{0.4pt}{1ex}%
      \endgroup% End the current group
    \fi%
    % Set style
    \ifcase\macro@stylenr\upshape\or\scshape\or\itshape\or\slshape\else\upshape\fi%
    % The text
    #2%
\endgroup% End the current group
}

\endinput
```