The package \texttt{create-theorem} provides commands for naming, initializing and configuring theorem-like environments. All of these commands have key-value based interface and are especially useful in multi-language documents, allowing the easy declaration of theorem-like environments that can automatically adapt to the language settings.

\textbf{/ 1 /}

How to load it

First, you need a backend to provide the command \texttt{newtheorem} with the usual behaviour, for example, amsthm. After that, you can simply load the current package with:

\begin{verbatim}
usepackage{create-theorem}
\end{verbatim}

\textbf{TIP}

Since \texttt{create-theorem} uses \texttt{cleveref} internally, it should usually be placed near the last of your preamble — notably, it needs to be loaded after \texttt{varioref} and \texttt{hyperref}.

It has the following options:

- \textbf{name as context}
  - When referencing, the names correspond to the current language context. For example, the English names will be displayed when referencing a theorem-like environment in English context, no matter which language context the original environment is in.
  - Synonymous names: \texttt{name-as-context} | \texttt{nameascontext} | \texttt{regionalref}

- \textbf{name as is}
  - When referencing, the names correspond to the language contexts in which the environments are defined. For example, if the environment is defined in English context, then the English names will be displayed when referencing it, regardless of the current language context.
  - Synonymous names: \texttt{name-as-is} | \texttt{nameasis} | \texttt{originalref}

- \textbf{name in link}
  - Include the names in the hyperlinks when referencing.
  - Synonymous names: \texttt{name-in-link} | \texttt{nameinlink}

- \textbf{no preset names}
  - Disable preset names. Use this option if you want to define you own name set.
  - Synonymous names: \texttt{no-preset-names} | \texttt{nopresetnames}
2.1 Naming theorem-like environments with \NameTheorem

The syntax of \NameTheorem is as follows:

\NameTheorem\{\langle name of environment\rangle\}\{\langle key-value configuration\rangle\}

Supported keys are:

- **heading** = \langle configuration\rangle
  - The heading of the environment, where \langle configuration\rangle can be:
    * a single string in monolingual documents: \texttt{heading} = \langle string\rangle;
    * a key-value name list in multilingual documents:
      
      \texttt{heading} = {\
        \langle language name\rangle = \langle string\rangle
      }

- **heading style** = \langle style\rangle
  - The style of the heading, you can specify the font, text style, color, etc.
  - Synonymous names: heading-style | headingstyle

- **crefname** = \langle configuration\rangle
  - The name for \cref the environment, where \langle configuration\rangle can be:
    * a single string in monolingual documents: \texttt{crefname} = \{\langle name\rangle\}{\langle names\rangle};
    * a key-value name list in multilingual documents:
      
      \texttt{crefname} = {\
        \langle language name\rangle = {\langle singular name\rangle}{\langle plural name\rangle}
      }

  - Also supports the syntax of \crefthename, thus you can assign names of the form:
    
    \{\langle singular definite article\rangle\} {\langle singular name\rangle} \{\langle plural definite article\rangle\} {\langle plural name\rangle}

- **crefname style** = \langle style\rangle
  - The style of the “crefname”, you can specify the font, text style, color, etc.
  - Synonymous names: crefname-style | crefnamestyle

- **Crefname**
  - The name for \Cref the environment, its syntax is the same as that of crefname.
  - Also supports the syntax of \Crefthename.

- **Crefname style** = \langle style\rangle
  - The style of the “Crefname”, you can specify the font, text style, color, etc.
  - Synonymous names: Crefname-style | Crefnamestyle

- **numbering style** = \langle style\rangle
  - The style of numbering in the reference, you can specify the font, text style, color, etc.
  - Synonymous names: numbering-style | numberingstyle

*If you’re feeling confused, don’t worry, there is an Examples section later illustrating its usage.*
2.2 Initializing theorem-like environments with `\CreateTheorem`

The syntax of `\CreateTheorem` is as follows:

```
\CreateTheorem{(list of the name of environments)}{(key-value configuration)}
```

Supported keys are:

- **name** = `(configuration)`
  - Setting the names. Same as `\NameTheorem{(name of environment)}{(configuration)}`.
- **name style** = `(configuration)`
  - Setting the style of names. Same as using `\NameTheorem`.
  - Synonymous names: name-style | namestyle
- **style** = `(theorem style)`
  - Specifying the `\theoremstyle` for the current environment.
  - Synonymous names: apply style | apply-style | applyystyle
- **parent counter** = `(parent counter)`
  - Specifying the `(parent counter)` for the current environment, i.e., numbering will restart whenever that sectional level is encountered.
  - Synonymous names: parent-counter | parentcounter | number within | number-within | numberwithin
- **shared counter** = `(shared counter)`
  - Specifying the `(shared counter)` for the current environment, i.e., numbering will progress sequentially for all theorem-like environments using this counter.
  - Synonymous names: shared-counter | sharedcounter | number like | number-like | numberlike
- **numberless**
  - Defining the current environment to be unnumbered.
- **create starred version**
  - Defining a corresponding starred (unnumbered) version of the current environment.
  - Synonymous names: create-starred-version | createstarredversion | create numberless version | create-numberless-version | createnumberlessversion
- **copy existed** = `(existed environment)`
  - Defining the current environment to be the same as `(existed environment)`.
  - This key is usually useful in the following two situations:
    1. To use a more concise name. For example, with `\CreateTheorem{thm}{copy existed = theorem}`, one can then use the name thm to write theorems.
    2. To remove the numbering of some environments. For example, one can remove the numbering of the remark environment with `\CreateTheorem{remark}{copy existed = remark*}`.
  - Synonymous names: copy-existed | copyexisted

*If you’re feeling confused, don’t worry, there is an Examples section later illustrating its usage.*
The names for the following environments have been preset: assertion, assumption, axiom, conclusion, conjecture, convention, corollary, definition, definition-proposition, definition-theorem, example, exercise, fact, hypothesis, lemma, notation, observation, postulate, problem, property, proposition, question, remark and theorem. If you are fine with the preset names, then you don’t need to specify the key “name” while creating them, otherwise you shall have to use the package option “no preset names” to disable the presets and then define your own ones.

Please note that, by default, a normal environment \textlangle} env \textrangle and its starred relative \textlangle} env \textrangle* do not share the same set of names, for the sake of generality. However, with proper usage of create starred version and copy existed, you are already able to produce all of the following combinations that shares the same set of names: 1) numbered \textlangle} env \textrangle, numbered \textlangle} env \textrangle*; 2) numbered \textlangle} env \textrangle, unnumbered \textlangle} env \textrangle*; 3) unnumbered \textlangle} env \textrangle, numbered \textlangle} env \textrangle*; and 4) unnumbered \textlangle} env \textrangle, unnumbered \textlangle} env \textrangle*. I left it as an easy exercise for you ;-)

2.3 Configuring theorem-like environments with \SetTheorem

The previous two commands are especially useful for package or class writers, while this one is more for the users. If you are not satisfied with preset name styles or numbering settings, then even after initializing the environments, you can still further configure them by means of \SetTheorem, the syntax of which is as follows:

\SetTheorem{(list of the name of environments)}{(key-value configuration)}

\textbf{TIP}
\SetTheorem should only be used in the preamble of your document.

Supported keys are:

- name = \textlangle} configuration \textrangle and name style = \textlangle} configuration \textrangle
  - Same as \NameTheorem\textlangle} name of environment \textrangle\textrangle{(configuration)}.
  - Note that this configuration can overwrite those already specified in \NameTheorem.
  - Synonymous names: name-style | namestyle

- parent counter = \textlangle} parent counter \textrangle
  - Specifying the \textlangle} parent counter \textrangle for the current environment, \textit{i.e.}, numbering will restart whenever that sectional level is encountered.
  - Note that this configuration can overwrite those already specified in \CreateTheorem.
  - Synonymous names: parent-counter | parentcounter | number within | number-within | numberwithin

- shared counter = \textlangle} shared counter \textrangle
  - Specifying the \textlangle} shared counter \textrangle for the current environment, \textit{i.e.}, numbering will progress sequentially for all theorem-like environments using this counter.
  - Note that this configuration can overwrite those already specified in \CreateTheorem.
  - Synonymous names: shared-counter | sharedcounter | number like | number-like | numberlike
3.1 | The environment idea

First, let's getting familiar with these two commands by creating the environment idea.

\NameTheorem{idea}{
  heading  = Idea,
  crefname = {idea}{ideas},
  Crefname = {Idea}{Ideas},
}
\CreateTheorem{idea}{ parent counter = section }

or to do it in one turn:

\CreateTheorem{idea}{
  name = {
    heading  = Idea,
    crefname = {idea}{ideas},
    Crefname = {Idea}{Ideas},
  },
  parent counter = section,
}

This is not exciting at all. Now, let's say we are writing a bilingual note in English and French. (I shall omit the \NameTheorem version and do it all at once in \CreateTheorem."

\CreateTheorem{idea}{
  name = {
    heading  = { english = Idea,  
                   french = Idée, },
    crefname = { english = {idea}{ideas},  
                 french = [l']{idée}[les]{idées}, },
    Crefname = { english = {Idea}{Ideas},  
                 french = [L']{idée}[Les]{idées}, },
  },
  parent counter = section,
}

With this, if you use \selectlanguage{french}, the idea environment shall be automatically displayed as “Idée”. And if you \crefthe it, the definite article and the name showed up properly just as expected.

Next we shall deal with the numbering problem. Let's continue to use this environment idea for demonstration – suppose that we have set the names up with \NameTheorem.

3.2 | Let's play with numbering

Remember the exercise I left you in the previous section? Let's do it together now.
3.2.1 Numbered idea and numbered idea*

This is easy, copy existed suffices:

\CreateTheorem{idea}\{parent counter = section\}
\CreateTheorem{idea*}\{copy existed = idea\}

3.2.2 Numbered idea and unnumbered idea*

This is the easiest, create starred version will do.

\CreateTheorem{idea}\{
    parent counter = section,
    create starred version,
\}

Notice that you cannot use \CreateTheorem{idea*}\{numberless\} here, since we don’t have names defined for idea*.

3.2.3 Unnumbered idea and numbered idea*

This is a bit tricky: by default we can only create numbered idea or unnumbered idea*, and the question is how to switch them. We shall need an intermediary.

\CreateTheorem{idea}\{create starred version\}
\CreateTheorem{idea-temp}\{copy existed = idea*\}
\CreateTheorem{idea*}\{copy existed = idea\}
\CreateTheorem{idea}\{copy existed = idea-temp\}

3.2.4 Unnumbered idea and unnumbered idea*

This is the combination of the first two cases — we need to create idea* first and then copy it to idea:

\CreateTheorem{idea}\{create starred version\}
\CreateTheorem{idea}\{copy existed = idea*\}

In each case, the two environments idea and idea* share the same set of names.

/ 4 /

Known issues

- The current mechanism does not work well for German, a problem originated in the package crefthe. The author plans to adopt a more refined approach in a later version so as to support the various grammatical situations in German.
- There may be inaccuracies in the translation of those preset names.

If you run into any issues or have ideas for improvement, feel free to discuss on:

https://github.com/Jinwen-XU/create-theorem/issues

or email me via ProjLib@outlook.com.