

This document tries to show how the package alist.tex may be used

copy

The control sequence `\copy` copies the *definition* of the *control sequence* in #2 and changes the *definition* of the *control sequence* in #1.

```
\def\aa{a}
\def\bb{b}
\copy\aa\bb
\aa → b
```

Note that since the *definition* is actually copied it stays the same even if the *control sequence* it was copied from is redefined.

```
\def\bb{c}
\aa → b
```

In contrast if you were to just redefine `\aa` as `\bb` its value would be bound to the *definition* of `\bb`

```
\def\aa{\bb}
\aa → c
\def\bb{d}
\aa → d
```

You can also copy a *control sequence's definition* into itself.

```
\copy\aa\aa
\aa → d
```

append

The control sequence `\append` appends the *definition* of the *control sequence* in #2 to the end of the *definition* of the *control sequence* in #1.

```
\def\aa{a}
\def\bb{b}
\append\aa\bb
\aa → ab
```

You can also append a *control sequence's definition* to itself.

```
\append\aa\aa
\aa → abab
```

contextwidth

The *control sequence* `\contextwidth` takes a `\dimen` register in #1 and anything in #2. It then stores the width #2 would have, were it written next to the arguments this control sequence previously accepted in #2.

```
\newdimen\dimena
\newdimen\dimenb
\contextwidth{\dimena}{\advance\dimenb10pt}
\the\dimena → 0pt
\contextwidth{\dimena}{\vrule width\dimenb}
\the\dimena → 10pt
```

list

The *control sequence* `\list` provides an adjustable method of typesetting lists it's argument #1 defines the bullet's to be used for the content given in #2. The simplest form of list only uses `\item`'s. These are automatically aligned behind the specified bullets. `\list{-\ }{`

```
\item{This is the content of the first item.}
\item{This is the content of the second item.}
```

Items may consist of multiple rows.

```
For a text of sufficient length I had to strain my imagination.}}
```

The previous code produces the following list:

- This is the content of the first item.
- This is the content of the second item. Items may consist of multiple rows. For a text of sufficient length I had to strain my imagination.

`\bullet` is defined such that bullets may be influenced by previous ones in the same list. Bullets may also be redefined within a list. This can happen either in bullet's or inbetween items. The content of items are encapsulated and can therefore not influence `\bullet`.

```
\newcount\bulletcount
\list{\advance\bulletcount1 \number\bulletcount)\ }{
```

```

\item{Using this method one can number items.
This item is labeled the first.}
\item{This item is labeled second.
The number conveniently changes without manual any alteration.}
{\def\bullet{$\bullet$} }
\item{Bullets may also be redefined inbetween any two items.}}
\item{When counting}
\item{high enough}
\item{you will see,}
\item{that}
\item{the list}
\item{automatically}
\item{choses}
\item{the appropriate spacing.}}
The previous code produces the following list:
1) Using this method one can number items. This item is labeled the first.
2) This item is labeled second. It's number conveniently increased without extra work.
• Bullets may also be redefined inbetween any two items.
3) When counting
4) high enough
5) you will see,
6) that
7) the list
8) automatically
9) choses
10) the appropriate spacing.
One can't just redefine \bullet, inbetween \item's but also put regular text.
\list{-\ }{
\item{The text below this item is not an item itself.}
\noindent This is not an item\hfil\break
\item{The text above this item is not an item itself.}
}
The previous code produces the following list:
- The text below this item is not an item itself.
This is not an item.
- The Text above this item is not an item itself.
    Lists can also be nested in other lists.
\list{$\bullet$ }{
\item{\list{-\ }{
    \item{Some text.}
    \item{\dots}
}}
\item{Second item.}}
The previous code produces the following list:
• - Some text.
  - ...

• Second item.

```