

Contents

You can go ahead through the complete file or jump to any of the example pages with the following links.

[Table builds](#) [Filling tables](#) [Building tables by column](#)
[More filled tables](#) [Forward and/or backward](#) [Taking a tour](#)
[Highlighting itemized lists](#) [Going through a list](#) [More](#)
[highlighting effects](#) [Still more list effects](#) [Including pictures](#)

Table builds

Building	your	table
----------	------	-------

Table builds

Building	your	table
line	by	line,

Table builds

Building	your	table
line	by	line,
entry		

Table builds

Building	your	table
line	by	line,
entry	by	

Table builds

Building	your	table
line	by	line,
entry	by	entry,

Table builds

Building	your	table
line	by	line,
entry	by	entry,

growing	up,	too.
---------	-----	------

Table builds

Building	your	table
line	by	line,
entry	by	entry,
from	the	bottom
growing	up,	too.

Table builds

Building	your	table
line	by	line,
entry	by	entry,
from	the	bottom
growing	up,	too.

Did you notice, that we have a footer from the very beginning and not only, when the page is complete?

Filling tables

Filling tables

Fill		
the		
first		
column,		

Filling tables

Fill	then	
the	the	
first	second,	
column,	and	

Filling tables

Fill	then	finally,
the	the	also
first	second,	the
column,	and	third.

Building tables by column

Building tables by column

Show
the
first
column,

Building tables by column

Show	then
the	the
first	second,
column,	and

Building tables by column

Show	then	finally,
the	the	also
first	second,	the
column,	and	third.

More filled tables

More filled tables

Fill		
	the	
		table

More filled tables

Fill		at
	the	
		table

More filled tables

Fill		at
	the	
random		
	positions	table

More filled tables

Fill	leave	at
some	the	
random		empty
	positions	table

More filled tables

Fill	leave	
some	the	well,
random	almost	empty
empty	positions	table

More filled tables

Fill	leave	
some	the	well,
random	almost	empty
empty	positions	table

Did you notice, that the element in the upper right corner has vanished?

Forward and/or backward

Writing

backwards.

Forward and/or backward

Writing sentences

write backwards.

Forward and/or backward

Writing sentences word

write backwards.

also

Forward and/or backward

Writing sentences word by

write backwards.

that, also

Forward and/or backward

Writing sentences word by word

write backwards.

need that, also

Forward and/or backward

Writing sentences word by word has
you need that, also
write backwards.

Forward and/or backward

Writing sentences word by word has always
if you need that, also
write backwards.

Forward and/or backward

Writing sentences word by word has always been
can, if you need that, also
write backwards.

Forward and/or backward

Writing sentences word by word has always been possible
you can, if you need that, also
write backwards.

Forward and/or backward

Writing sentences word by word has always been possible with [word by word](#) now you can, if you need that, also write backwards.

Forward and/or backward

Writing sentences word by word has always been possible with PPower4. But now you can, if you need that, also write backwards.

Taking a tour

1

Taking a tour

1 

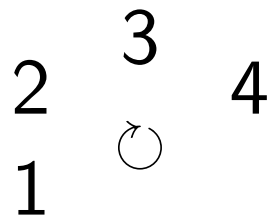
Taking a tour

2
1 ↻

Taking a tour

2 3
1 ↻

Taking a tour



Taking a tour

2 3 4
1 ↻ 5

Taking a tour

2 3 4
1 ① 5
6

Taking a tour

2 3 4
7 7 5
6

Taking a tour

8 3 4
7 ① 5
6

Highlighting itemized lists

- This is an important topic.

Highlighting itemized lists

- This is an important topic.
- But only, until the next appears.

Highlighting itemized lists

- This is an important topic.
- But only, until the next appears.
- And both are less important, when the third item is here.
And of course we can highlight also long items now.

Going through a list

- You can also walk through a list,
- which is visible from the very beginning,
- and highlights the item, that is currently most important.

Going through a list

- You can also walk through a list,
- which is visible from the very beginning,
- and highlights the item, that is currently most important.

Going through a list

- You can also walk through a list,
- which is visible from the very beginning,
- and highlights the item, that is currently most important.

More highlighting effects

- We start this highlighted and keep it for a while.
- To show the effects we need more items.
- Still more text.
- And the final text on this slide.

More highlighting effects

- We start this highlighted and keep it for a while.
- To show the effects we need more items.
- Still more text.
- And the final text on this slide. Something to appear is here.

More highlighting effects

- We start this highlighted and keep it for a while.
- To show the effects we need more items.
- Still more text.
- And the final text on this slide. Something to appear is here.

More highlighting effects

- We start this highlighted and keep it for a while.
- To show the effects we need more items.
- Still more text.
- And the final text on this slide.

Still more list effects

- Hi! Again this starts highlighted here. And we want to come back to this later.

Still more list effects

- Hi! Again this starts highlighted here. And we want to come back to this later.
- Sorry for these boring items.

Still more list effects

- Hi! And we want to
come back to this later.
- Sorry for these boring items.
- But somehow we must fill this slide.

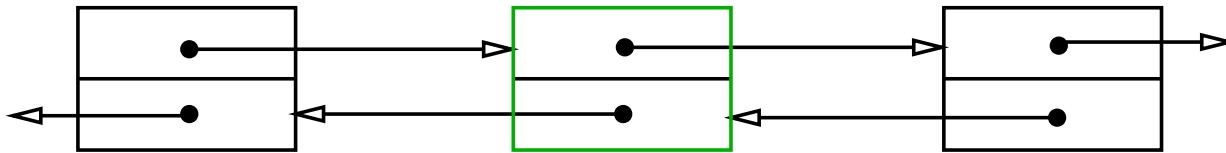
Still more list effects

- Hi! And we want to
come back to this later.
- Sorry for these boring items.
- But somehow we must fill this slide.
- And we got it.

Including pictures

```
void dlink::append( dlink *p ) {  
    p->suc = suc;  
    p->pre = this;  
    suc->pre = p;  
    suc = p;  
}
```

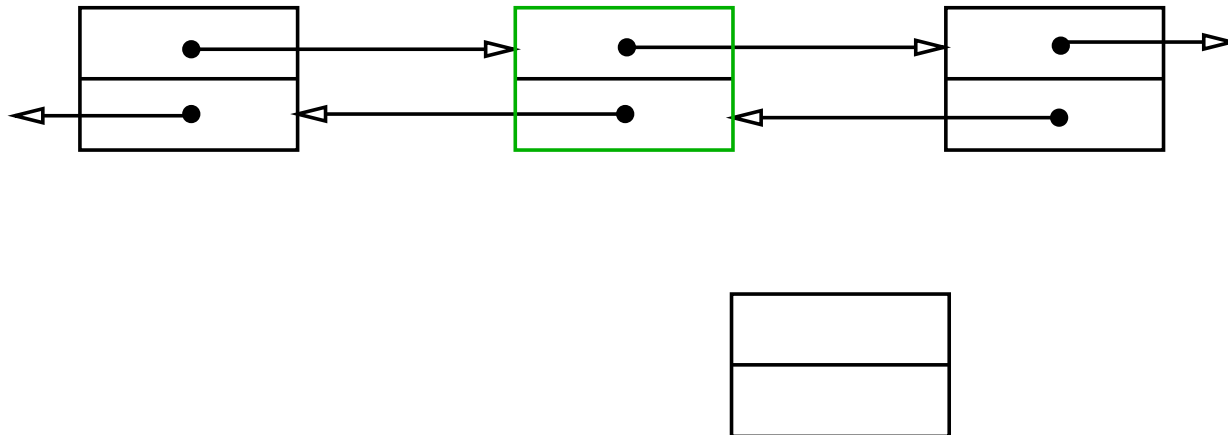
We can also highlight pieces of program code and present a corresponding illustration, which shows the resulting changes in a data structure.



Including pictures

```
void dlink::append( dlink *p ) {  
    p->suc = suc;  
    p->pre = this;  
    suc->pre = p;  
    suc = p;  
}
```

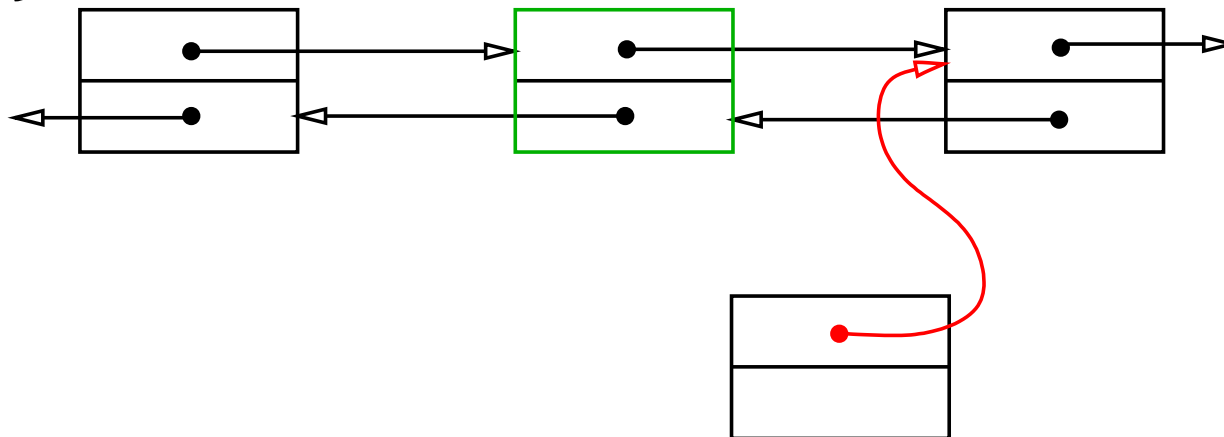
We can also highlight pieces of program code and present a corresponding illustration, which shows the resulting changes in a data structure.



Including pictures

```
void dlink::append( dlink *p ) {  
    p->suc = suc;  
    p->pre = this;  
    suc->pre = p;  
    suc = p;  
}
```

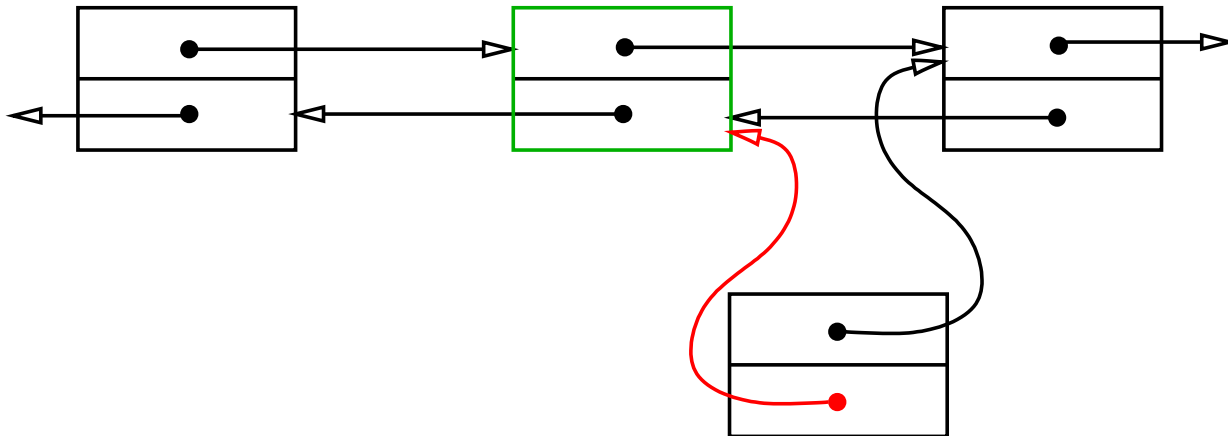
We can also highlight pieces of program code and present a corresponding illustration, which shows the resulting changes in a data structure.



Including pictures

```
void dlink::append( dlink *p ) {  
    p->suc = suc;  
    p->pre = this;  
    suc->pre = p;  
    suc = p;  
}
```

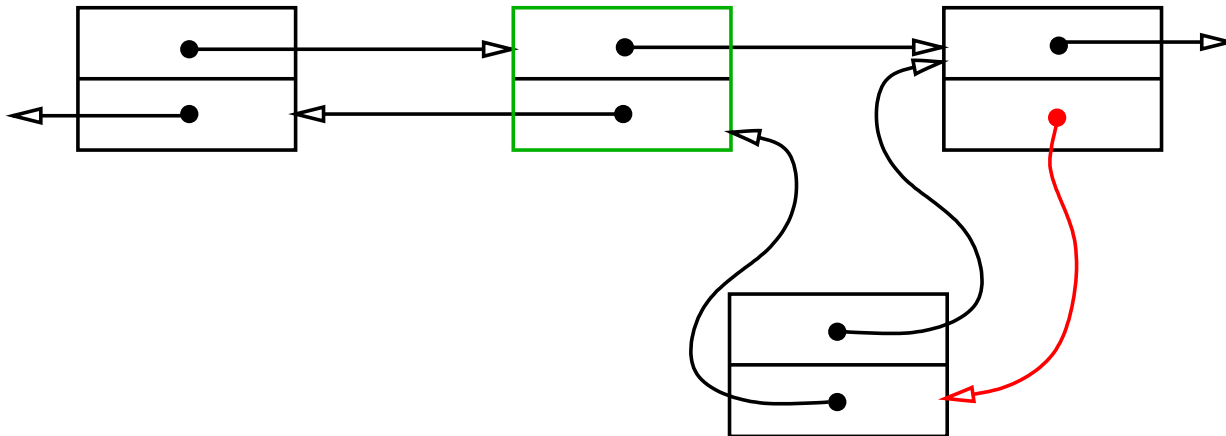
We can also highlight pieces of program code and present a corresponding illustration, which shows the resulting changes in a data structure.



Including pictures

```
void dlink::append( dlink *p ) {  
    p->suc = suc;  
    p->pre = this;  
    suc->pre = p;  
    suc = p;  
}
```

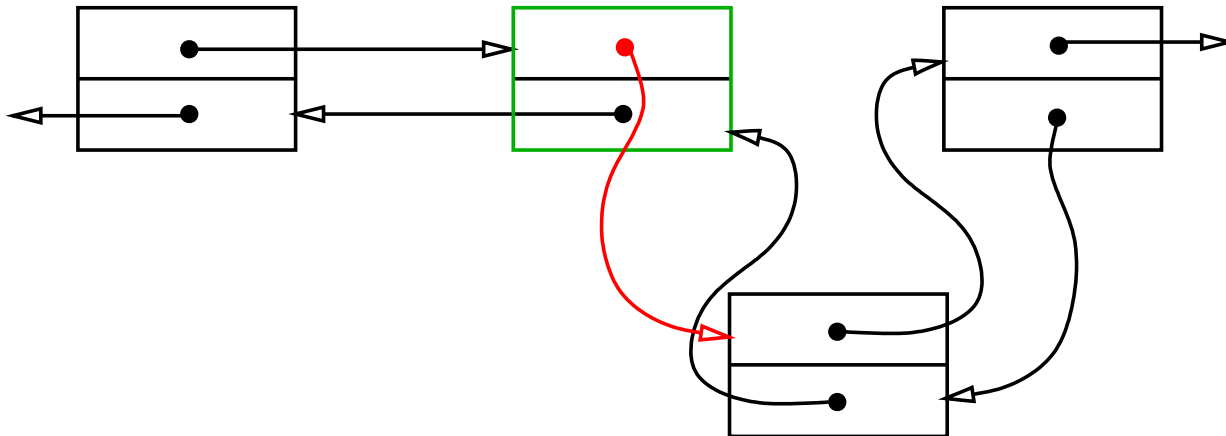
We can also highlight pieces of program code and present a corresponding illustration, which shows the resulting changes in a data structure.



Including pictures

```
void dlink::append( dlink *p ) {  
    p->suc = suc;  
    p->pre = this;  
    suc->pre = p;  
    suc = p;  
}
```

We can also highlight pieces of program code and present a corresponding illustration, which shows the resulting changes in a data structure.



Thanks for having a look

The features demonstrated here can be created with `pdflatex`, `vtex` and the combination of `latex` and `dvipdfm` using the post processor PPower4.

If you would like to check this out, see the [homepage](#) of PPower4. Please send comments concerning features and the documentation. I would appreciate also suggestions for more examples.

Thank you for your cooperation!