Introduction to extending

Printing values

I often use display-scheme-music and pretty-print.
Worth mentioning?
OTOH, there are so many printing possibilities...

Probs

For your example #(display (ly:music-property #{ c'8 #} 'pitch))
The output for #(display-scheme-music ...) would be different, worth explaining?

Glossary of important object types

Books, bookparts, scores

If not explicetely specified, bookparts are not in the hierarchy afaik:

```
bk = \book { \score { R1 } }
#(display (ly:book-book-parts bk))
->()
```

As opposed to scores:

```
bk = \book { { R1 } }
#(display (ly:book-scores bk))
```

The whole implementation of bookpart is a mess and buggy.
Maybe don't mention bookpart at all?

Stencils

"Every grob has a stencil"
Not every but most.

Markups

Probably add "Mostly used outputting text"

LilyPond and Scheme

Table of contents is missing in the side bar.

I think the sentence
"As a conclusion, you should generally use the hash, with a few exceptions for the dollar in case you want to play syntax tricks or copy a value conveniently."
is not always correct.
In music-functions, when calling a variable containing music .$ is preferable, because otherwise a subsequent call of the same variable will not see the original.

Music objects

Music function basics

In def of nrepeats probably use index? not integer?
Actually it requires an exact index...

The \etc shortcut

Not limited to music-function, see:
\markup foo = \markup \with-color #red \etc
\markup \foo "arg"
Worth mentioning?

Music transformation tools

(music-map function music)
."... function is applied to the bottom music objects ..."
This is misleading, maybe better:
function is first applied to the bottom music objects

The translation process

The context hierarchy

Well, the Global-context has no parent...

Writing an engraver

Basics

"For engravers defined in Scheme, the argument is no longer a string, but a procedure:"
Maybe "For user-defined not registered scheme-engravers, ..."

I'm not comfortable with the pseudo-code following
"Here is a template for writing an engraver:"
Obviously some hooks are missing.
I suggest to either show them all, with correct arguments:

```
(acknowledgers
  ((grob-interface-1 engraver grob source-engraver)
    ...)
```

as you do lateron
or something like

```
(make-engraver
  ((hook-1 args)
    ...)
  (hook-2
    ((args)
     ...)
     ...)
```

...)
and explain later what they are supposed to be.

Backend programming

The causation chain

"Look for the grob’s cause, then its cause if a grob, etc., until an event is found as cause."
I don't understand this sentence, admittedly I'm not a native speaker...

Items, spanners and columns

My point of view is different:
All grobs are either items or spanners, this is the first criterion to distuingish, imho
What you call column-grobs are a subset of items and sort of container-grobs in my thinking.
Hence I'd say: grobs are devided into spanners and items.
Column- (or container-) grobs are a subset of items.
Breakable items as well.
(NonMusical)PaperColumns are a subset of said column-grobs.

"Spanners extend horizontally between two columns."
Glissando acts a little different, afaik.

Unpure-pure containers

A plethora of TODOs ;)
Let me add another one: a meaningful code example for a custom unpure-pure container, unpure/pure funtions written in scheme...