Guile-Present and Texinfo

Works better than you’d think

Andy Wingo

Why Texinfo?

Easier thinking in plain text, without distractions of graphical tools like OpenOffice

Programmatically consistent slides

Texinfo pretty good for describing programming languages

Because we can!

First, a preamble

```
\input texinfo
@documentencoding UTF-8
@settitle Example presentation
```

Makes the document valid texinfo

Next, the title slide

```
@titlepage
@title Example presentation
@subtitle Subtitle 1
@subtitle Subtitle 2
@author Jane Doe
@end titlepage
```

Sections are slides

Slides start with `@section` or other sectioning commands like `@subsection`. For example, this slide starts with:

```
@section Sections are slides
```

Text after the `@section` is the body of the slide.

Normal texinfo rules govern paragraph breaks: a blank line separates paragraphs, as do block elements like section headers or itemized lists.

Text markup

Anywhere you have text, you can have inline text markup.

```
@code{code}, @emph{emphasized text},
@strong{strong words}
```

`code`, *emphasized text*, **strong words**

There is also `@var`, which looks the same as `@emph`.

Inline text commands

Anywhere you have text, you can have inline commands.

- `@email{a@@b.com}`: `a@b.com`
- `@uref{http://gnu.org/}`: `http://gnu.org`
- `@result{}`: ⇒

We will add more with time.

Standalone Images

An image alone in its own paragraph is scaled to fit the available space, and centered horizontally.

```
@image{guile-banner-small.png}
```

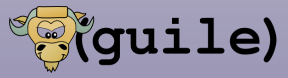
Manually specified constraints

Limiting width:

```
@image{guile-banner-small.png,400}
```

Limiting height:

```
@image{guile-banner-small.png,,25}
```



Limiting both:

```
@image{guile-banner-small.png,400,25}
```

Inline images

Images in paragraphs are scaled to the `text-height` and wrap the paragraph as needed.

```
Some Guile logos:
@image{guile-banner-small.png,400},
@image{guile-banner-small.png,400},
@image{guile-banner-small.png,400},
@image{guile-banner-small.png,400}
```

Some Guile logos:

,

,

,

Pre-formatted text

```
@example
Text with fixed-width fonts.
@end example
```

produces

```
Text with fixed-width fonts.
```

The same with `smallexample` will use a smaller font; don’t expect people to be able to read it, though!

```
Text in @smallexample is smaller.
```

Escaping

`@`, `{`, and `}` are special to Texinfo and need to be escaped.

To write them in Texinfo, prefix them by another `@`: `@@`, `@{`, and `@}`.

These characters need escaping even in `@example`.

However they don’t need escaping in `@verbatim`, so you can use that if you don’t need embedded formatting.

Itemized lists

```
@itemize
@item Item 1
@item Item 2
bla bla
@item Item 3
@end itemize
```

- ❧ Item 1
- ❧ Item 2 bla bla
- ❧ Item 3

Options

After the `@settitle`, a sectioning command, or any begin/end kind of command like `@example`, you can set some options.

Layout in Guile-Present is as with CSS: there are properties that cascade. For example if you wanted all slides in your deck to be in Georgia and have a gradient background from an SVG, put this after the `@settitle`:

```
@c +background: background.svg
@c +font-family: Georgia
```

Available properties (1)

`margin-left`, `margin-right`, `margin-top`, `margin-bottom`, all of which default to 64 pixels

`text-height`, `line-spacing` and `block-spacing`, defaulting to 42, 1.1 and 1.4, respectively

`font-family`, `font-slant`, `font-weight`, defaulting to `Serif`, “normal”, and “normal”

`bullet-string`, `bullet-font-family`, defaulting to ❧ and `Sans`

Available properties (2)

`header-height`, defaulting to 64 pixels

`background`, which should be an image file, and defaults to nothing (no image)

`background-color` and `foreground-color`, defaulting to black on white (`#xFFFFFF` and `#x000000`)

`width`, and `height`, defaulting to 1024 by 768.

Slide notes

Text in `@iftex` blocks will not be rendered in slides, but will be rendered by `texi2pdf`. The reverse is true for `@ifnottex`.

```
@iftex
Here are some things to remind myself to say
@end iftex
```

End with `@bye`

@bye