

The ConTeXt presentation styles

Meta Graphics

fun for non-wizzards

a demonstration



```
What I will discuss
-----
Building the Meta-POST core
Including MetaPost code
Building a Meta-Graphic
Including a graphic
```

```
Building the Meta-POST core
-----
Building the Meta-POST core
Including MetaPost code
Building a Meta-Graphic
Including a graphic
```

```
Building the Meta-POST core
-----
Building the Meta-POST core
Including MetaPost code
Building a Meta-Graphic
Including a graphic
```

```
Building the Meta-POST core
-----
Building the Meta-POST core
Including MetaPost code
Building a Meta-Graphic
Including a graphic
```

green

Because this style offers a lot of space, it is one of my favourites. The few simple buttons are typical METAPOST graphics. This style supports structuring. There are three main buttons and one extra button that can be customized.

More than text alone

- graphics
- navigation
- fields
- intelligence



funny

This is a typical *one-toc with just-a-few-pages* style. The bottom half of the outline around the text shows the progress. This style is well suited for a summary from which one can launch other documents. This is how it was used first.

The Future of Documents

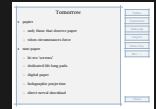


colorful

Although the title page of this style is rather random, the content can be well structure. There is a large text area available and the current position in the style is visualized at the bottom. The buttons are small METAPOST graphics.

Documents

some thoughts



fuzzy

ConTeXt users will recognize this style as being derived from the reference manual style. Indeed we use the same colors and random rectangles. At the right the main topics are shown and a close button. This style is well suited for itemized talks.

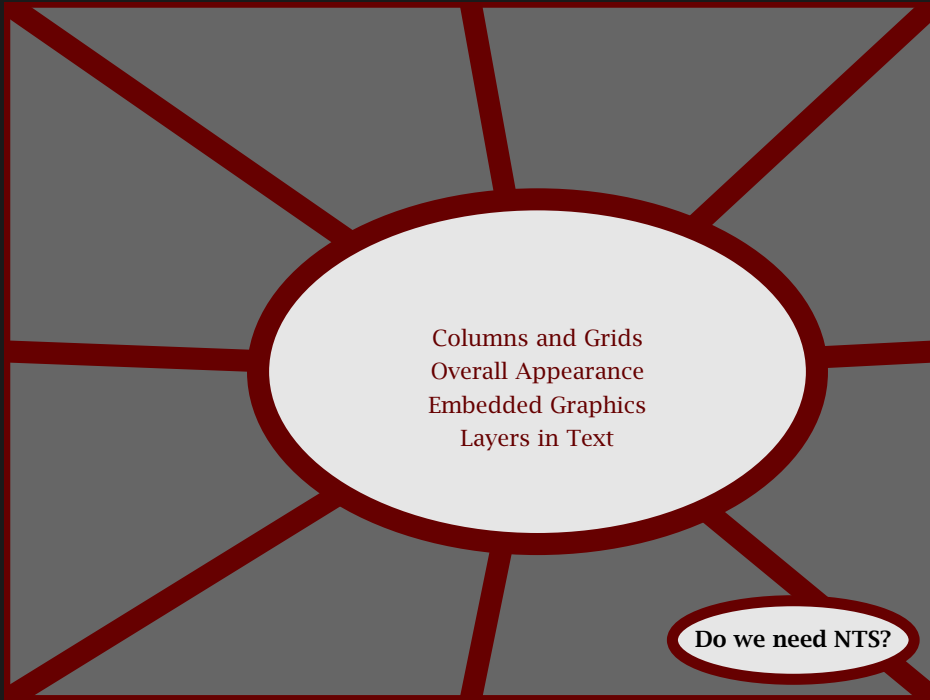
Think before you code

Quit



polish

The Antikwa Torunska font had a rather prominent backward slant. This style uses this font both for the running text and as background. The items follow the slant. This style is not meant for presentation with much text.



spider

In this style the text snippets are positioned at a random location on the page, to which the background graphic adapts itself. Each series of subtopics gets a different color. You can navigate the document by clicking on the circular shapes.

Fancy Graphics

positions and layers



wonder

This is one of the 6 styles made for the nts presentation at EuroTEX 1999. The idea was to demonstrate a couple of nasty things that one can do with pdfTEX, being an example of an extension. Afterwards it was provided that this could also be done using traditional TEX.

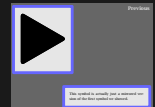
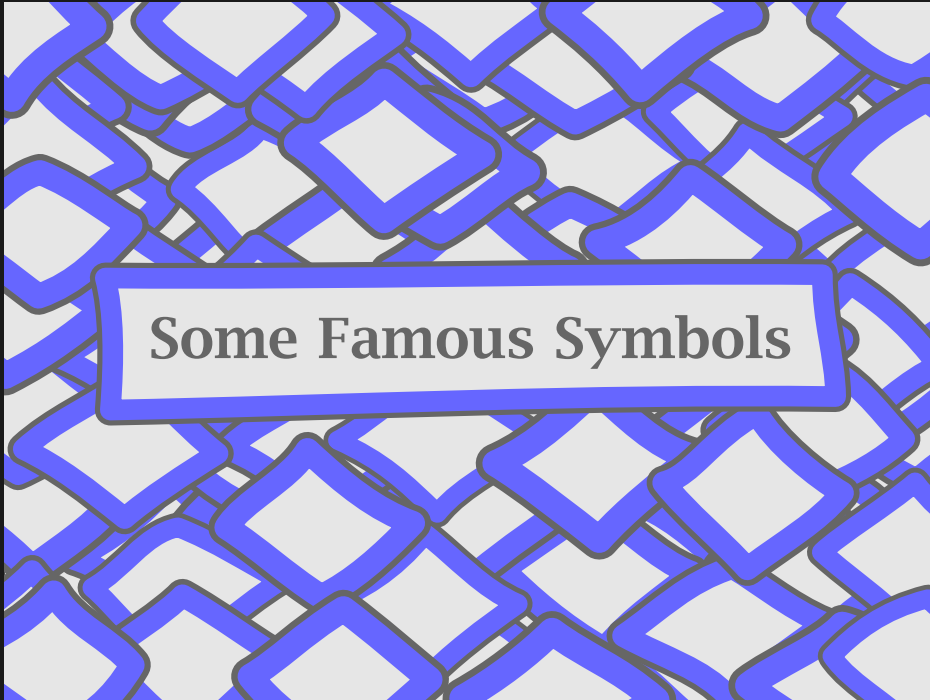
The MAPS Bibliography

from XML to PDF



window

In order to explain the way the Maps bibliography was made, we needed a style to present both a sample and explanation. This style presents pairs as well as single screens.



grow

This style looks like the previous one, but this time we build up the page. The dimensions of the text window are determined by the width of the sample window. As with its precursor, this style highly structures your talk.

Are we an endangered species?

(some thoughts and questions)

Hans Hagen



writing

The Lucida Handwriting font is a pretty good font, and this style is meant to accompany it. The text frame is drawn by METAPOST using a penshape as are the small symbols that identify items and navigation buttons.



Facsimile Snapping Breaking Notes

1



split

This simple style can be used for relative short presentations with much text per screen. The colors are rather dark in order to let the text area shine bright.

The Future of NTS

Hans Hagen



balls

This is a special kind of style. You can stepwise build up a statement and present the individual steps as a piecewise built graphical paragraph. Each statement gets a different color.

Definitions

$\text{T}_{\text{E}}\text{X}$
 $\text{T}_{\text{E}}\text{X}$
MetaPost
Portable Document Format

JavaScript
eXtensible Markup Language
Con $\text{T}_{\text{E}}\text{X}$ t



knot

When you are in need of a quick and dirty presentation, this style can be your friend. It provides a minimum of structure and the scetchy outline add to the hasty job. Given the bit of randomness, it is also a typical example of a Con $\text{T}_{\text{E}}\text{X}$ t style.

The screenshot shows a web application interface with a large white content area on the left and a yellow sidebar on the right. The content area contains the title "Educational Typesetting" and the author "Hans Hagen, GUT 2001". The sidebar contains five buttons: "Playground", "Conditions", "Observations", "Solutions", and "ConTeXt". A "close" button is located at the bottom right of the content area.

Thumbnail of the 'Playground' page, showing a list of items and a sidebar with navigation buttons.

Thumbnail of the 'Conditions' page, showing a list of items and a sidebar with navigation buttons.

Thumbnail of the 'Observations' page, showing a list of items and a sidebar with navigation buttons.

Thumbnail of the 'Solutions' page, showing a list of items and a sidebar with navigation buttons.

organic

This style is more subtle than it looks at first sight. The buttons at the right, adapt themselves to the shape around the text, which can be a random one. This style was developed in the process of writing the [MetaFun](#) manual, and is described in detail in there. The outline comes in three flavors, one of them being a random shape.

Postprocessing PDF

It is not uncommon to postprocess the files
to make them look like real PDFs, making A5

What is PDF

For long DVI was TeX's native output format. This format can be converted to for instance POSTSCRIPT or PDF. The later format has the advantage that fonts and graphics are embedded which make the file portable across platforms. We start this day with a short explanation of what PDF is.

Hans Hagen

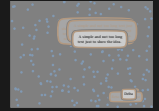
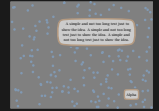


cycle

This style was used at tug 2000 to introduce the pdfTeX related talks. It cycles through the topics and summaries by moving them to the front one by one.

Something
Very Important

August 2000



speckle

Sometimes a presentation is just a bunch of quotes. In that case this presentation style can be used to put them upfront. Each quote (or summary) is added to the previous page. By clicking on a quote one can go back to the page were it was presented. At the bottom we present the title.

The ConTEXT Test Quotes

August 7, 2001



more

This is another example of a style that stepwise builds up a screen. We cycle through the corners of the page with slightly random windows. It's one of my favourites.

The ConTeXt Story

a quick tour

What output do...
What input do...
What system do...

The next . . . minutes

What output do we want
What input do we prefer
What T_X do we run
What system do we use

What output do we want

- high quality paper output
 - one input — multiple out-put
 - high level of abstraction
- optimal screen output
 - high quality typography
 - dedicated navigation
- some examples
 - local references
 - intelligent menus
 - text as program
 - lots of graphics
 - screens and paper
 - multiple fonts
 - simple but effective
 - lots and indices
 - cross linked source
 - typical T_X
 - program as text

What input do we prefer

- highly structured ASCII
- medium neutral coding
- short file

What T_X do we run

- DVI output
 - DVI generation
 - DVI viewing
- PDF output
 - pdf generation
 - pdf viewing GhostScript
 - pdf viewer Acrobat

What system do we use

- Starting point
- Pitfalls
- Structure
- Typography
- Navigation
- User interface
- Environment
- Resources

Starting point

- typographic tradition
- eternal use
- optimal navigation
- future media
- new visions

Pitfalls

- structure
- fonts
- graphics
- compression
- portability
- searching
- interactivity
- quality
- flexibility
- protection

Structure

- multiple layouts
- reordering of data
- degrees of freedom
- easy maintenance
- fill in forms
- JavaScript
- sound and video

Typography

- nature (text, math, theme)
- character
- fragmentation
- floating bodies
- adaptive layout
- number substitutes
- enhanced pagebody
- color and backgrounds
- integrated METAFONT graphics

Navigation

- extensive sectioning
- local/global referencing
- long/cross references
- linked indices
- link-di lists
- navigation bars
- subpageing
- menus and buttons
- intelligence
- parallel lists
- reader profiles
- version control

User interface

- parameter driven
- project organisation
- multi lingual
- minimal backing needed
- books into main routines
- object oriented structure
- selective processing
- isolated specials

Environment

- programs
 - weh2, context, yandtex, pdftex
 - dvipson, GhostView, Dvifilter, Reader
 - T_XMitt
- formats
 - Plain T_X
 - ConTeXt

Resources

- documentation
- examples
- www.tex.nl/context
- http://context.org.nl
- look for: Gilbert, Hans, Taco, Tobias

original / pre-01

More than text alone

- graphics
- navigation
- fields
- intelligence

The calculator demo

Combining `tk`, `swtwidgets`, `tkinter` and `javaScript`.

Just a few examples

- in-line fill-in fields
- parents, children, clones and copies
- field characteristics
- entering vs text
- all kinds of fields
- advanced referencing
- popping up information
- and some more
- figures and fields
- and more of those

Some new concepts

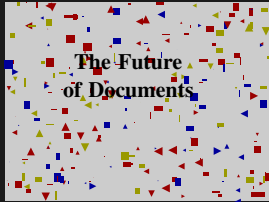
- creating objects
- selective inclusion
- one pass dilemma
- generating datastructures

Advanced referencing

Just some alternatives.

```
\goto[reference]  
\goto[outer reference:]  
\goto[outer reference:|inner reference]  
\goto[operation[argument]]  
\goto[operation[action[argument,argument]]]  
\goto[action]  
\goto[action[argument]]  
... chained or not.
```

funny / pre-03



Some thoughts . . .

- The current state of documents
- The future state of documents
- The current state of typesetting
- The future state of typesetting
- Managing data and authoring content
- But . . .
- So . . .

The current state of documents

- paper
 - wandering eyes
 - you see what you buy
 - external rewording
 - a pleasure to use
 - crisp for the bin
- screens
 - typoed on keyboard
 - typoed on the fly
 - complete line format
 - reasonable constraints

The future state of documents

- paper
 - only those that deserve paper
 - when circumstances force
- non-paper
 - holographic projection
 - direct neural download
 - hi-res 'screens'
 - dedicated life long paths
 - digital paper

The current state of typesetting

- made up by hand
- made up in back
- keyed / scanned in
- generated on the fly

The future state of typesetting

- typesetting on demand
- intuitive authoring
- internal reuse
- cheat watching

Managing data and authoring content

- what you think is what is stored
- what you think is what you get
- integrity testing / checking
- author / user / design driven
- paid for giving and using
- personal storage / retrieval

But . . .

- there is nothing decent yet around
- we must make the best of it

So . . .

- why are we doing these things
- how to deal with the real feeling
- do we want to live with these temporary hacks
- will I ever play a role in this
- when will the real games stand up



Today

- paper
 - instant impressions
 - you see what you buy
 - can be a pleasure to see
 - more and more crap
- screen
 - typeset on forehead
 - typeset on the fly
 - complete free format
 - reasonable constraints

Buttons: Today, Tomorrow, Make-up, Targets, Some Day, But ... Show

Tomorrow

- paper
 - only those that deserve paper
 - when circumstances force
- non-paper
 - hi-res "screens"
 - dedicated life long pads
 - digital paper
 - holographic projection
 - direct neural download

Buttons: Today, Tomorrow, Make-up, Targets, Some Day, But ... Show

Make-up

- made up by hand
- made up automatically
- keyed and/or scanned in
- generated on the fly

Buttons: Today, Tomorrow, Make-up, Targets, Some Day, But ... Show

Targets

- intuitive authoring
- typesetting on demand
- eternal reuse
- clever searching

Buttons: Today, Tomorrow, Make-up, Targets, Some Day, But ... Show

Some Day

- thinking documents and content
- what you think is what's get stored
- what you think is what you get
- integrity testing / indexing
- author / user / design driven
- give and take, paid for or not
- personal life-long storage and retrieval

Buttons: Today, Tomorrow, Make-up, Targets, Some Day, But ... Show

But ...

- there is nothing decent yet around
- we must make the best of it
- what role can T_X play

Buttons: Today, Tomorrow, Make-up, Targets, Some Day, But ... Show

fuzzy / pre-05

Think before you code

Quit

Topics

- document structure
- data abstraction
- optimal typography
- potential complications

Quit

Document structure

- provide non traditional sectioning
- use name spaced cross references
 - tables of contents must be able to adapt themselves"
 - anything can and will be reused"
 - many things can become a marginal, foot- or endnote"
 - anything goes into headers, footers and sidebar"
 - there are more uses than we can think of.

Quit

Data abstraction

- think of future demands and tools
- there is a difference between local and global data
- never code something more than once"
- force consistency by using abbreviations and alle
- avoid hard coded names in changing documents"

Quit

Optimal typography

- it's in the small points, never compromise on kerning"
- tables can be typeset differently, flows can come back"
- math, physical units and chemicals are not the same
- languages have different demands and conventions
- we seldom know in advance what it will look"

Quit

Potential complications

- lack of structure and consistency
- input, output and glyph encodings, forms and searching
- use of periods, capitals, quotes and spaces"

Quit

polish / pre-06

Columns and Grids
Overall Appearance
Embedded Graphics
Lines in Text

we need it?!

First of all we want our columns to be perfectly balanced. This is critical for page text, but imagine lots of white space, like display math.

Columns and Grids

We want floats to be moved to the best available location. Of course we want floats to span more than one column, and even spanning one and a half columns with a nice flowing around the figure should be possible.

Columns and Grids

We definitely don't want to end up with two lines of text on the opposite pages separated. When we think the page number is such a way that we get full pages? Of course we will need more than paragraph and page reorganization for this, we are dealing with the document as a whole.

Columns and Grids

Columns may differ in width. Think of two columns, spanning "one-third and two-thirds" of a page. In the middle of each row columns we will want to represent an illustration, and the text should follow the circular shape of this illustration.

Columns and Grids

Talking of illustrations, instead of being something with fixed dimensions, the whole may be adjusted, or even constructed, to suit the overall document appearance (grid, vertical, and more).

Columns and Grids

Are you still thinking from left to right? Text can go in all directions, and will be moved into the width of columns may change in the meantime. Anyone who has seen traditional Jewish religious documents, will see the challenge in mixed columns with floatrows. Having general parallel columns.

Columns and Grids

A case study of columns.

Columns and Grids

Typing is more than manipulating sources. That's why we need a typesetting system that looks at the glyphs themselves, the small graphics!

Overall Appearance

People tend to disagree on what looks best, but experts often agree on what looks worse. Why not build an expert knowledge, or even better, build a system that learns from the user's rating?

Overall Appearance

How is graphics calculated? Does it use an expert like learned lists of glyphs, or does it first build a category? At least there it knows how the pages come out. In the validation a fraction of an output device? Will the change of glyphs depends on the rating? Will "X" and "x" become one?

Overall Appearance

In validating the appearance, a model of the page needed, as terms of meaningful areas? If so, how is such a model defined? Do we need pattern recognition?

Overall Appearance

We can think a graphic object, or maybe even several. Models for reordering of illustrations between processes dealing with page repositioning and drawing images need to be developed. Such architectures should cooperate naturally with the paragraph and page breaking in text.

Embedded Graphics

Typing along curves, having shapes into outlines, and applying arbitrary filling and shading, is all about curves.

Embedded Graphics

TeX is strong in math, but has about 1000 characters? Not much satisfactorily results can be reached. Sure is possible. That's why we all need documents, that make an order how to represent that in TeX. Lots of thinking needs to go into that area.

Embedded Graphics

For some languages putting together glyphs is not enough. Actually drawing glyphs, or even better words or sentences can be an alternative. From experience one can make it into a system text. Strong handwriting oriented graphics has to meet expensive coding.

Embedded Graphics

Some examples of embedding graphics.

Embedded Graphics

First of all, the user system needs some more understanding about the layout text, support for macros, nested glyph names is needed.

Lines in Text

When we step through a document, some knowledge we want to language the text we're dealing with makes sense. Not only the (user) languages of a text, but the direction also plays a role. Complex text features should be recognized properly.

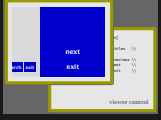
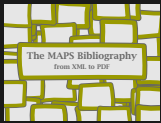
Lines in Text

In many document documents, like HTML-forms, interaction with a typesetting engine is not a luxury, especially not in complex and custom languages. We can see a plug in, but the document itself should contain the information needed to let us do it by itself. A document is more than a collection of graphics and glyphs, and typesetting more than organizing them.

Lines in Text

As we've already demonstrated using TeX to embed special characters like pre-compiled and text files in a browser. Although heavily dependent of features of browsers, we will benefit from a direct model of browser which we present as well as concepts of information hidden in the output but showing up at wish.


Lines in Text



window / pre-09


Some Famous Symbols

Symbols




This symbol can be used to indicate a hyperlink to a previous page.

Previous




This symbol can be used to indicate a hyperlink to a previous page.

Previous



This symbol can be used to indicate a hyperlink to a previous page.
As one can expect there is also a symbol for going to the next page.


Previous




This symbol is actually just a mirrored version of the first symbol we showed.

Is this one or not?


First and Last



First and Last



First and Last




A few screens back, we saw this symbol.

First and Last



A few screens back, we saw this symbol.
This symbol represents the beginning of something.

First and Last



A few screens back, we saw this symbol.
This symbol represents the beginning of something.
Just like this one represents an end.

First and Last




A few screens back, we saw this symbol.
This symbol represents the beginning of something.
Just like this one represents an end.
They look just like the symbols found on audio and video players.

Summary



So we have a symbol for previous ...

Summary




So we have a symbol for previous ...
... and one for next ...
... and one for next ...

Summary



So we have a symbol for previous ...
... and one for next ...
... and yet another for first ...
... and yet another for first ...

Summary



So we have a symbol for previous ...
... and one for next ...
... and yet another for first ...
... and of course for last.

Are we an endangered species?

(some thoughts and questions)

Hans Hagen

What we're going to discuss

TeX users
The publishers
The readers
Time
TeX and MetaPost
The future authors
The future publishers
The medium
TeX and MetaPost again
The ending
Conclusions

TeX users

- They are readers as well as authors
- They often like writing.
- They have strong opinions on how things should look.
- Most of them are not trained in typesetting.
- Many of them are pretty well aware of how to save time.
- They like to concentrate on writing instead of formatting.
- They use TeX because of its quality.
- They can easily (and routinely) update their documents.
- They want to be (and often operate) at the front of technology.
- They don't know how to convince people to use TeX.

The publishers

- They need to consider typesetting to be a specialization.
- They are no longer primarily driven by content.
- They have to change, will change, and are changing.
- They want the highest quality for the lowest price.

The readers

- They move and move buy in WWW bookstores.
- They will start using these dedicated devices soon.
- Many of them still want to own books.
- They still prefer "look and feel over "style".
- They want to see in advance what they buy.
- They keep old copies of articles and books.
- They will finally get tired of updating.

Now ...

- Will there be proof?
- Will there be design?
- Will there be quality?
- What type of content will we deal with?
- Who will be in control?

TeX and MetaPost

- Both can do a lot but are seldom used to full power.
- They are among the most powerful tools available.
- They demonstrate that typesetting can largely be automated.
- Like cats, TeX and MetaPost have many lives.

The future authors

- They must learn to think in structure and reason.
- More and more they have to be aware of readers and finalization.
- They had better not only accept that something cannot be done.
- They have to learn to think in mixed text and graphics.
- They should be aware of "eternal" use and reuse.

The future publishers

- They will publish, but, as well as typical on demand.
- They will use techniques that go far beyond current technologies.
- They will have to decide between good and need (and quality).
- They have to combine psychology, ergonomics and technology.

The medium

- Are publishers willing to distribute document source code?
- How many holy grails do we expect to find?
- What technologies can we expect?
- We will move from paper to screen to mind.

TeX and MetaPost again

- TeX is no easy job for TeX and MetaPost.
- In many areas they lead competitors.
- They are still the most accurate tools available.
- Both provide cutting edge technology.
- Their users want to be in control.
- We still need to put out some problems.

The coding

- DMH showed a way but DMG got away with it.
- We can do math, (and), so we can do them all.
- It's not only in the coding, but also in the methods.
- We often do we want to rule the job.
- We will probably soon learn what we did forget.
- We had better learn to look into the future.

Conclusions

- Nothing that real surprises is happening.
- We are already 15 years ahead but often unaware of it.
- Coding is not the main issue, the (change in) content is.
- There will be documents, but we will not recognize them as such.
- We have the tools and new tools are coming.
- We have stability and finally people will use it.
- TeX is one of the few life-long tools around.
- For a long time, TeX will be around but nobody will notice.
- There are some illusions to come, so we can take some time.
- We must not forget the past and not ignore the future.

1

Facsimile
Snapping
Breaking
Notes

1

2

Facsimile
Snapping
Breaking
Notes

2

3

Facsimile
Snapping
Breaking
Notes

3

4

Facsimile
Snapping
Breaking
Notes

4

split / pre-14

The Future of NTS

Hans Hagen

The production of NTS packages

The production of NTS packages

The production of NTS packages

The production of NTS packages

The production of NTS packages

The production of NTS packages

The production of NTS packages

The production of NTS packages

The alternative path has brought

The alternative path has brought

The alternative path has brought

The alternative path has brought

The alternative path has brought

The challenge will be to

The challenge will be to

The challenge will be to

Together a NTS

Together a NTS

Together a NTS

Together a NTS

Together a NTS

Together a NTS

Together a NTS

After 1910000

After 1910000

After 1910000

After 1910000

After 1910000

After 1910000

After 1910000

After 1910000

After 1910000

After 1910000

balls / pre-15

Definitions

TeX
LaTeX
MetaPost
Portable Document Format

TeX

- It was the most TeX to be a programming language that is possible. The intention was to be able to describe a document and its layout.

- Both are referred to as TeX and LaTeX. The program interprets the source and produces a layout of the document.

- Although TeX originates in the beginning of the 1980s, it is still one of the most powerful tools available for typesetting complex documents.

- Attempts to make TeX to be a TeX can only lead to more.

TeX

We use the most TeX to be a programming language and a program. The language can be used to describe a document and its layout. Both are referred to as TeX and LaTeX. The program interprets the source and produces a layout of the document.

Although TeX originates in the beginning of the 1980s, it is still one of the most powerful tools available for typesetting complex documents. Attempts to make TeX to be a TeX can only lead to more.

MetaPost

This program is derived from MetaPost. Like TeX, both interpret a language and a program. When MetaPost generates layout, MetaPost is using its own internal graphics.

The graphic capabilities of both programs are limited. The latest MetaPost version (2.85) has a number of extensions that can be used for other graphics than MetaPost can do.

Portable Document Format

Professional printers and other typesetters often have a hard time to convert the graphic page description language PostScript to other PDF. The portable document format, also known as PDF, is a standard way of describing a document.

The standard uses a number of extensions and support for document layout and content. The PDF standard is a good standard for describing document layout and content.

JavaScript

Also known as the ECMAScript language, JavaScript is an object-oriented programming language. It is primarily used to be embedded in documents.

Although it looks a lot like Java, JavaScript is primarily used for client-side scripting. It is primarily used to be embedded in documents.

eXtensible Markup Language

Even since computer-aided typesetting, people have been trying to describe their documents in a way that the computer can deal with their structure and content. XML is a standard, based on the principles of SGML, and the extensible markup language XML and TeX.

These languages can be recognized by the language-independent XML. The languages are used to describe documents in the Document Type Definition (DTD) and the extensible markup language XML and TeX.

ConTeXt

A collection of general purpose macros written in TeX, ConTeXt is used to be a TeX program. ConTeXt uses a number of extensions and can be used for typesetting paper and layout documents.

MetaPost support is implemented in ConTeXt, allowing the most advanced graphic capabilities. An extension to the TeX program, ConTeXt is a good standard for describing document layout and content.

Educational Typesetting

- authors want control over their text
- designers want to make fancy things
- publishers want to **publish content**

Hans Hagen, GUT 2001

Buttons: **Playground**, **Conditions**, **Observations**, **Solutions**, **Context**, **Home**

Playground

- authors want control over their text
- designers want to make fancy things
- publishers want to **publish content**

Buttons: **Playground**, **Conditions**, **Observations**, **Solutions**, **Context**, **Home**

Conditions

- materials should follow changes in educational programs
- content must be available in many forms and depths
- information to be retained for formal and job related education
- content should be coded in one source or come from databases
- consistency should be guaranteed within authors teams
- workflows should be automated as much as possible

Buttons: **Playground**, **Conditions**, **Observations**, **Solutions**, **Context**, **Home**

Observations

- 130 years ago demands were not that high, and it was not that hard to hire good **processors (publishing & books)**. **TeX** permits you to reach a very high level of abstraction
- today **TeX** is the **hobbyist**, so whatever you produce, it should become **TeX**, **one day** also there are some limitations, so a **way live** has become **much more easier**
- **flexible standards** pop up faster than one can support them, **fighting them will become more and more important**
- the publishing world is about to change fast: publishers become **brokers of information (ohai)**

Buttons: **Playground**, **Conditions**, **Observations**, **Solutions**, **Context**, **Home**

Solutions

- convertible input **format**
- structured content **processors & maintain**
- high quality output **(Adobe & Adobe)**
- controlling the process **(Adobe & Adobe)**
- **typesetting on demand (and [progress.pdf.com](#))**

Buttons: **Playground**, **Conditions**, **Observations**, **Solutions**, **Context**, **Home**

Context

- general purpose **source package** is necessary in major upgrade stages
- extensive support for **structuring and reuse** (in **TeX** and **HTML**)
- **reusable blocks and extendible source frameworks** will be provided
- **evolving into a full-featured toolkit** (**script, shell, style**)
- will soon provide **UTF-8** file control and features

Buttons: **Playground**, **Conditions**, **Observations**, **Solutions**, **Context**, **Home**

organic / pre-19

Postprocessing PDF

What is PDF

For long time was \LaTeX 's native output format. This format can be converted to for instance POSTSCRIPT or PDF. The latter format has the advantage that fonts and graphics are embedded which make the file portable across platforms. We start this day with a short explanation of what PDF is.

Hans Hagen

Postprocessing PDF

The history of PDF \LaTeX

The PDF \LaTeX project started ... years ago. In its current incarnation, this program is rather stable and mature. However, it took quite some development, discussion and testing, and the PDF \LaTeX mailing list has played an important role in this. In this regard, this project can be considered one of the most innovative \LaTeX related activities of the end of the previous century. How did it all evolve?

Sebastian Rahtz

Postprocessing PDF

Fonts in PDF \LaTeX

Since PDF \LaTeX provides its own backend, it also has to deal with font inclusion. PDF \LaTeX supports type 1 as well as true-type and bitmap fonts. Some can be included directly, others needs special treatment. Fonts can be embedded completely, partially, or not at all. Also, users have to set up some map files. Although font support is rather straightforward, some basic knowledge can be handy.

Erik Fraibach

Postprocessing PDF

How PDF \LaTeX can improve your paper

It may have gone unnoticed in many paper users, but one of the main reasons for developing PDF \LaTeX was the wish to improve the visual appearance of the page. The current status of \LaTeX The Program, limits this improvement to the individual paragraphs and pages. Carefully PDF \LaTeX provides several methods to improve the look and feel of a page. Systematic experiments and research were the basis for the evolution of PDF \LaTeX .

Hin Tho Thinh

Postprocessing PDF

Graphics in PDF \LaTeX

A consequence of being its own backend, is that PDF \LaTeX must include graphics itself. PDF \LaTeX supports the PDF, EPS , PNG and vector graphic formats. EPS graphics can be converted to PDF. Because PDF \LaTeX gives you access to low level PDF, it can also support that resolution graphics. When embedding graphics one has to consider resolution and color.

Hans Hagen

Postprocessing PDF

PDF \LaTeX in a workflow

Since PDF is one of the major file formats, PDF \LaTeX is a good candidate for acting as a backend in processing data. How does that work, and what is needed to get it working?

Ed Cashin

Postprocessing PDF

Going beyond static documents

The last few years, the world of documents has changed drastically. Color has become natural on the desktop and screen documents go beyond their static counterparts. One way to enhance documents is to use advanced hypertext links. A more drastic deviation from traditional documents is embedding program code, like JAVASCRIPT . One can use this scripting language to provide comfortable navigation and intelligence to documents. PDF \LaTeX provided the hooks to embed such scripts into the document. In a similar way, one can use PDF \LaTeX to make advanced forms.

Hans Hagen

Postprocessing PDF

Setting up PDF \LaTeX

Since PDF \LaTeX is an all-in-one tool, the \LaTeX user no longer has to deal with a multi-stage source to paper process. Installation is not that complicated, but there a few thing you should know about the configuration.

Ed Cashin

Postprocessing PDF

Postprocessing PDF

It is not uncommon to postprocess the files produced by \LaTeX , for instance making AT binders out of \LaTeX documents. Since PDF \LaTeX can process PDF graphics, it can do its own advanced postprocessing, sometimes going far beyond what's common in the \LaTeX world. Another kind of postprocessing involves converting PDF into a textual format. An example of this application is an experimental utility that converts \LaTeX into HTML, in a rather natural way.

Brend de Boer

Something
Very Important

August 2000

A simple and not too long text just to show the idea. A simple and not too long text just to show the idea. A simple and not too long text just to show the idea.

Alpha

A simple and not too long text just to show the idea. A simple and not too long text just to show the idea.

Beta and Gamma

A simple and not too long text just to show the idea.

Delta

A simple and not too long text just to show the idea. A simple and not too long text just to show the idea. A simple and not too long text just to show the idea.

Epsilon

A simple and not too long text just to show the idea. A simple and not too long text just to show the idea. A simple and not too long text just to show the idea.

Zeta, Eta and Theta

A simple and not too long text just to show the idea. A simple and not too long text just to show the idea. A simple and not too long text just to show the idea. A simple and not too long text just to show the idea. A simple and not too long text just to show the idea.

Omega


```
module pre-01
colors BackgroundColor InteractionColor ContrastColor
```

```
\TitlePage{}
```

```
\Topics{}
```

```
\Topic{}
```

```
\Subject{}
```

```
module pre-02  
colors BackgroundColor OrnamentColor
```

```
\TitlePage{}
```

```
\Topics{}
```

```
\Topic{}
```

```
\Subject{}
```

```
module pre-03  
colors PageColor BackgroundColor ContrastColor
```

```
\TitlePage{}
```

```
\Topics{}
```

```
\Topic{}
```



```
module pre-04
colors red green blue yellow gray
```

```
\TitlePage{}
```

```
\Topics{}
```

```
\Topic{}
```

```
module pre-05  
colors BackgroundColor OrnamentColor
```

```
\TitlePage{}
```

```
\Topic{}
```

```
module pre-06  
colors PageColor BackgroundColor ContrastColor
```

```
\TitlePage{}
```

```
\Topics{}
```

```
\Topic{}
```

```
module pre-07
colors PageColor TextColor LineColor linecolor 1-6
```

```
\TitlePage{}
```

```
\Topics{}
```

```
\StartIdeas
```

```
  \Topic{}
```

```
  \StartIdea
```

```
  \StopIdea
```

```
\StopIdeas
```

```
module pre-08
colors PageColor TextColor LineColor [ColorPage ColorLine]
layers text sample idea
modes demo
```

```
\TitlePage{}
\StartIdea
  \StartSample \StopSample
  \StartText   \StopText
\StopIdeas
```

```
module pre-09
colors PageColor TextColor LineColor InteractionColor
modes demo
```

```
\TitlePage{}
\Topic{}
\StartIdea
  \Topic{}
  \StartSample \StopSample
  \StartText \StopText
\StopIdeas
\StartText
\StopText
```

```
module pre-10
colors PageColor TextColor LineColor InteractionColor
modes demo
```

```
\TitlePage{}
\Topic{}
\StartIdea
  \Topic{}
  \StartSample \StopSample
  \StartText \StopText
  \StartSubText \StopSubText
\StopIdeas
\StartSample
\StopSample
\StartText
\StopText
\StartSubText
\StopSubText
```

```
module pre-13
colors   TextColor PageColor LineColor SymbolColor
```

```
\TitlePage{}
```

```
\Topics{}
```

```
\Topic{}
```


module pre-14
colors One Two

\TitlePage{}

\Topic{}

\Subject{}

```
module pre-15
colors TextColor PageColor LineColor linecolor 1-6
modes demo
```

```
\TitlePage{}
\StartIdea
  \StartItem
  \StopItem
\StopIdea
```

```
module pre-16
colors TextColor PageColor LineColor
modes demo
```

```
\TitlePage{}
\StartIdea
  \Topic{}
  \NextIdea
\StopIdea
```

```
module pre-19
colors TextColor PageColor OrnamentColor InteractionColor ContrastColor
modes demo
```

```
\TitlePage{}
\Topic{}
```

module pre-22
colors TopColor BotColor DotColor
modes demo

\TitlePage{}{}
\StartSummary{}{}
\StopSummary

module pre-23
colors TopColor BotColor DotColor
modes demo

\TitlePage{}{}
\StartSummary{}{}
\StopSummary

```
module pre-26  
modes demo reverse
```

```
\TitlePage{}  
\StartTopic  
\Title{}  
\StopTopic  
\ColofonPage{}
```