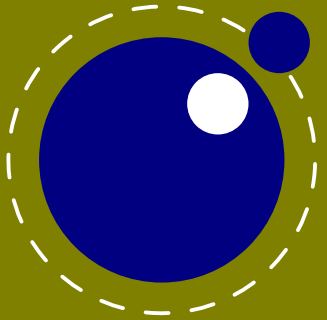
LuaTEX

0.95 – 2016

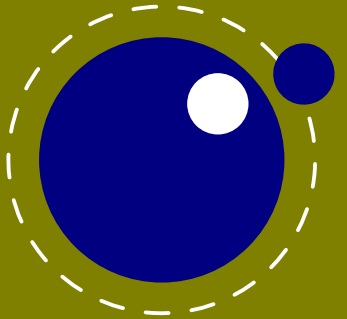
Hans Hagen – bachoTEX 2016



End March we released LuaTEX 0.95,
the first official ‘stable’ release, a
prelude to 1.00 later this year.

The interface will not change
significantly so that one
can write stable packages.

We have used LuaTEX
ourselves for over a decade
in order to figure things out.

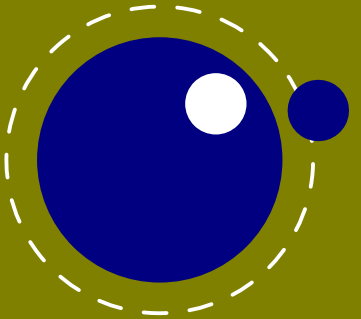


luatex

0.95 – 2016

Around 2005, after we talked a bit about it, Hartmut added the Lua scripting language to pdfTeX as an experiment.

This add-on was inspired by the Lua extension to the Scite editor that I (still) use.



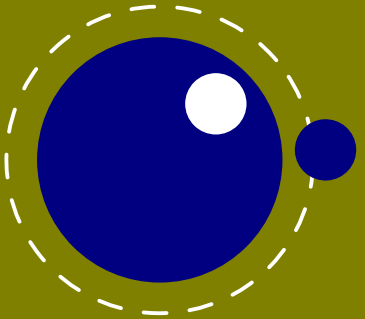
luatex

0.95 – 2016

One could query dimensions, counter registers and box dimensions and print strings to the TEX input buffer.

The Oriental TEX project then made it possible to go forward and come up with a complete interface.

For this, Taco converted the code base from Pascal to C, an impressive effort.



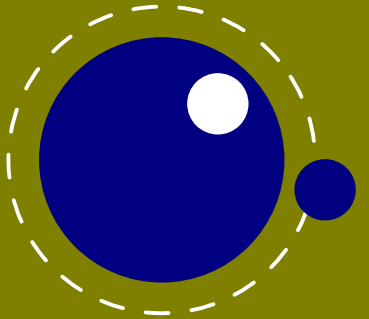
luatex

0.95 – 2016

We spend more than a year intensively discussing, testing and implementing the interface between TEX and Lua.

In successive years we polished things and extended bits and pieces.

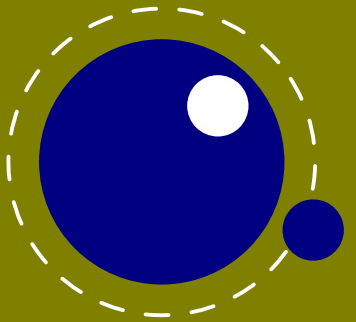
The last few years we cleaned up, filled in gaps and reached the point where we were more of less satisfied.



luatex

0.95 – 2016

The core is still traditional TEX, but extended with pdfTEX protrusion and expansion (reworked) and directional features from Aleph (cleaned up).



luatex

0.95 – 2016

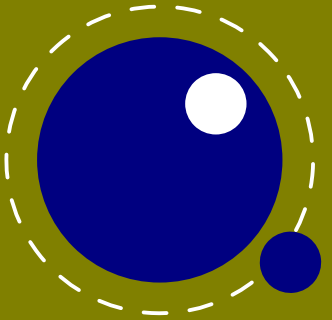
The font subsystem
accept now wide fonts.

The hyphenation machinery
can use runtime loaded
(and extended) patterns.

Hyphenation, ligaturing,
kerning are separated.

Most steps in processing node lists
can be intercepted using callbacks.

The math machinery has
opentype math code paths.



luatex

0.95 – 2016

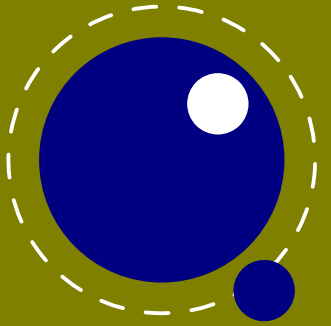
All in- and output can be
controlled and intercepted.

The backend code has
been separated better.

You can write (simple) parsers.

Nodes can be accessed
and manipulated.

Images and reuseable
boxes are now native.



luatex

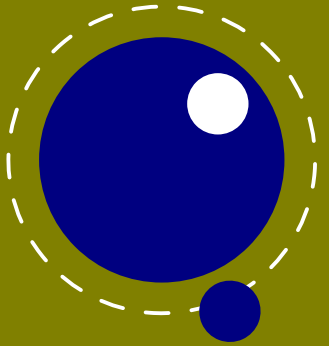
0.95 – 2016

The project is driven by ConTEXt users and development.

Right from the start ConTEXt supported LuaTEX.

This means that most mechanisms have been tested in production.

Raw performance is less that 8 bit pdfTEX but in practice and on modern machines LuaTEX behaves well.



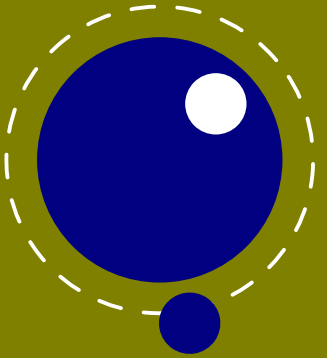
luatex

0.95 – 2016

We will continue development,
but functionality will stay
stable within versions.

The code will be further
streamlined and documented.

The manual will be improved over time.



luatex

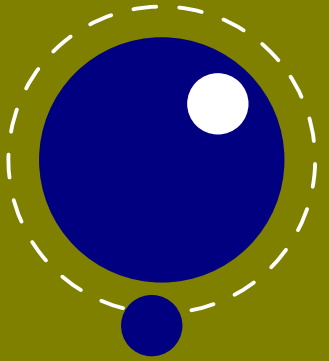
0.95 – 2016

Hans Hagen

Hartmut Henkel

Taco Hoekwater

Luigi Scarso



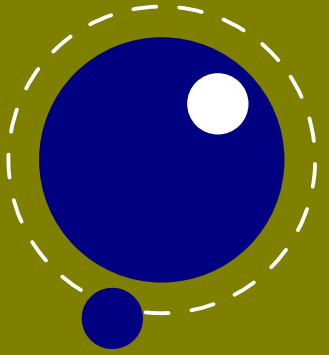
luatex

0.95 – 2016

In just over a decade of development
and extensive usage we added
a couple of new primitives.

Also over time many helpers were
introduced at the Lua end that access
TEX properties and manipulate lists.

The next pages only mention
recent additions and changes.

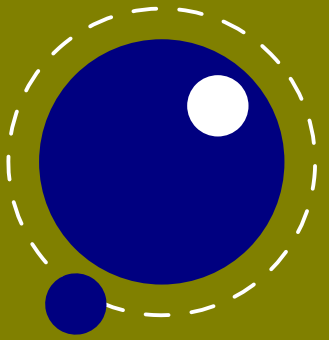


luatex

0.95 – 2016

Forms and images are now stored in rule nodes and not longer extension whatsits:

```
\saveboxresource
\saveimageresource
\useboxresource
\useimageresource
\lastsavedboxresourceindex
\lastsavedimageresourceindex
\lastsavedimageresourcepages
```

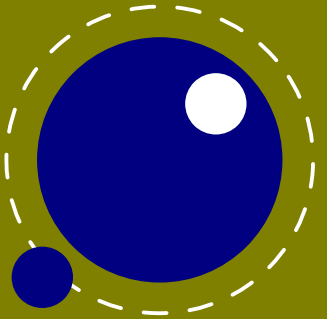


luatex

0.95 – 2016

Direction nodes are now core nodes and we only have a few directions left:

TLT
TRT
LTL
RTT

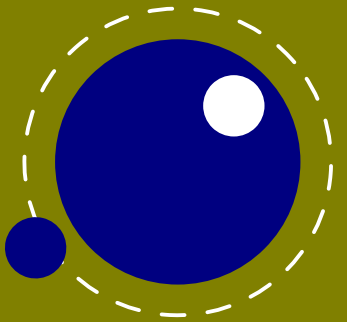


luatex

0.95 – 2016

Some pdfTeX commands
have been promoted:

\pagewidth \pageheight
\adjustspacing \protrudechars
\ignoreligaturesinfont
\expandglyphsinfont \copyfont
\normaldeviate \uniformdeviate
\setrandomseed \randomseed
\ifabsnum \ifabsdim \ifprimitive
\primitive \savepos \lastxpos
\lastypos \outputmode
\draftmode \pxdimen \insertht

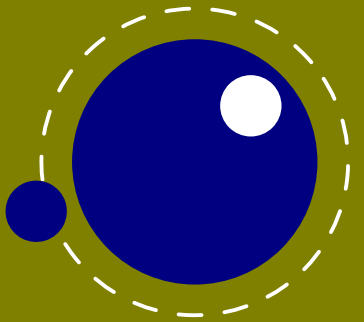


luatex

0.95 – 2016

As a side effect of better separation the backend commands are collapsed:

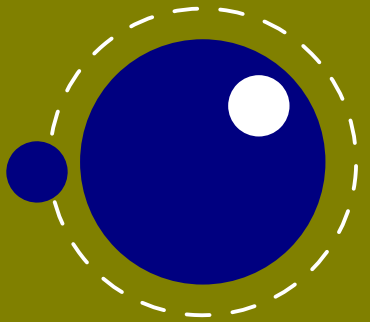
```
\pdfextension keyword
   [some follow up]

\pdfvariable keyword

\pdffeedback keyword
```



luatex

0.95 – 2016

Redundant or experimental commands and functionality has been removed.



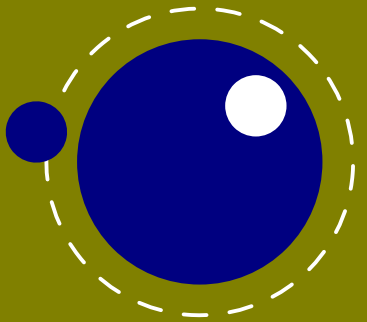
luatex

0.95 – 2016

Internally rule nodes are used for rules, reusable content, images.

In addition we have empty rules: `\novrule` and `\nohrule`.

At the Lua end we can have user rules that trigger a callback in the backend.



luatex

0.95 – 2016

We added a few new low level helpers:

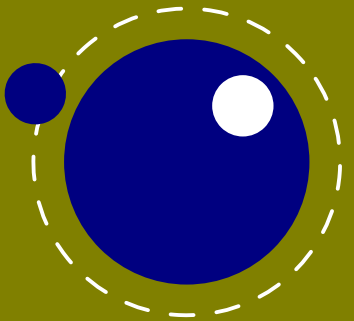
`\nospaces` for preventing spaces
(1 = no spaces, 2 = zero skips)

`\[e]toksapp` `\[e]tokspre`
for extending token registers

`\letcharcode` just to be nice

`\begincsname`
`\lastnamedcs` for less code

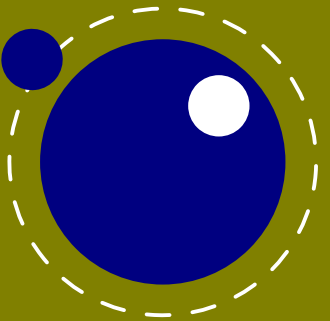
`\hpack` `\vpack` `\tpack`
to avoid callbacks



luatex
0.95 – 2016

There are a few error suppressors:

```
\suppresslongerror
\suppressmathparerror
\suppressoutererror
\suppressifcsnameerror
```

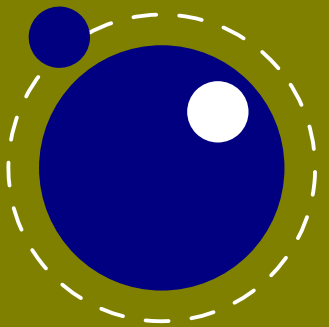


luatex

0.95 – 2016

\Uleft \Uright \Umiddle

[height <dimension>]
[depth <dimension>]
[axis]
<delimiter>

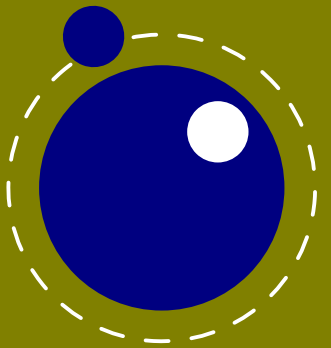


luatex

0.95 – 2016

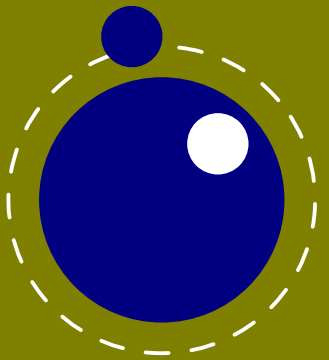
\Uvextensible

[height <dimension>]
[depth <dimension>]
[axis]
[exact]
<delimiter>



luatex

0.95 – 2016



\Uhextensible
\U[over|under]delimiter
\Udelimiter[over|under]

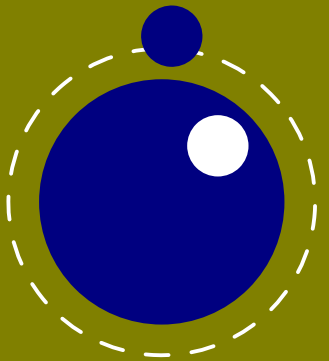
[height <dimension>]
[depth <dimension>]
[left|middle|right]
<family>
<slot>

luatex

0.95 – 2016

\Umathaccent

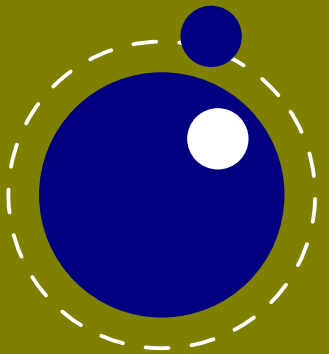
[top|bottom|overlay]
[fixed]
[fraction <number>]
<delimiter>
{content}



luatex

0.95 – 2016

```
{ {1}
  \Uskewed /
[exact|noaxis]
  {2} }

{ {1}
\Uskewedwithdelims / ()
    [exact|noaxis]
  {2} }
```

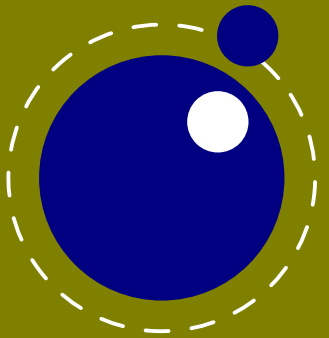


luatex

0.95 – 2016

A `\matheqnogapstep` factor that determines the gap between formula and equation number.

A `\mathdisplayskipmode` directive that controls display skips: 1 = always, 2 = only when not zero, 3 = never.



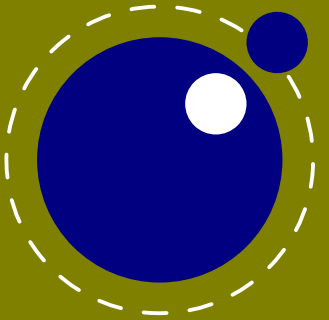
luatex

0.95 – 2016

a \mathscriptsmode parameter
that determines how to apply

\Umathsubshiftdown (d)
\Umathsupshiftup (u)
\Umathsubsupshiftdown (s)

0	dynamic	dynamic
1	d	u
2	s	u
3	s	u + s − d
4	d + (s − d)/2	u + (s − d)/2
5	d	u + s − d



luatex

0.95 – 2016