

News from boxes and glue

TUG 2023

July 14 2023

Hotel Collegium Leoninum

Bonn

Patrick Gundlach
gundlach@speedata.de

speedata
Berlin, Germany

boxes & glue

speedata 

Let's surpass the mainstream

~~News from~~ ~~boxes and glue~~

Lessons learned

TUG 2023

July 14 2023

Hotel Collegium Leoninum

Bonn

Patrick Gundlach
gundlach@speedata.de

speedata
Berlin, Germany

boxes & glue

speedata 

Let's surpass the mainstream

What is “boxes and glue” (recap)?

boxes and glue ...

What is “boxes and glue” (recap)?

boxes and glue ...

... is a collection of software libraries

What is “boxes and glue” (recap)?

boxes and glue ...

... is a collection of software libraries

... not a ready-to-run piece of software

What is “boxes and glue” (recap)?

boxes and glue ...

- ... is a collection of software libraries
- ... not a ready-to-run piece of software
- ... written in the Go programming language

What is “boxes and glue” (recap)?

boxes and glue ...

- ... is a collection of software libraries
- ... not a ready-to-run piece of software
- ... written in the Go programming language
- ... the attempt to bring T_EX's superb typesetting quality to a modern environment

What is “boxes and glue” (recap)?

boxes and glue ...

- ... is a collection of software libraries
- ... not a ready-to-run piece of software
- ... written in the Go programming language
- ... the attempt to bring T_EX's superb typesetting quality to a modern environment
- ... and of course OpenSource

Same data structures and API

Same data structures and API

- ▣ Nodes (glyph, glue, rule, whatsit)

Same data structures and API

- ▣ Nodes (glyph, glue, rule, whatsit)
- ▣ Node packing (vpack, hpack)

Same data structures and API

- ▣ Nodes (glyph, glue, rule, whatsit)
- ▣ Node packing (vpack, hpack)
- ▣ Rectangular items

Same data structures and API

- ▣ Nodes (glyph, glue, rule, whatsit)
- ▣ Node packing (vpack, hpack)
- ▣ Rectangular items
- ▣ Line breaking

Same data structures and API

- ▣ Nodes (glyph, glue, rule, whatsit)
- ▣ Node packing (vpack, hpack)
- ▣ Rectangular items
- ▣ Line breaking
- ▣ Hyphenation

Same data structures and API

- ▣ Nodes (glyph, glue, rule, whatsit)
- ▣ Node packing (vpack, hpack)
- ▣ Rectangular items
- ▣ Line breaking
- ▣ Hyphenation
- ▣ ...

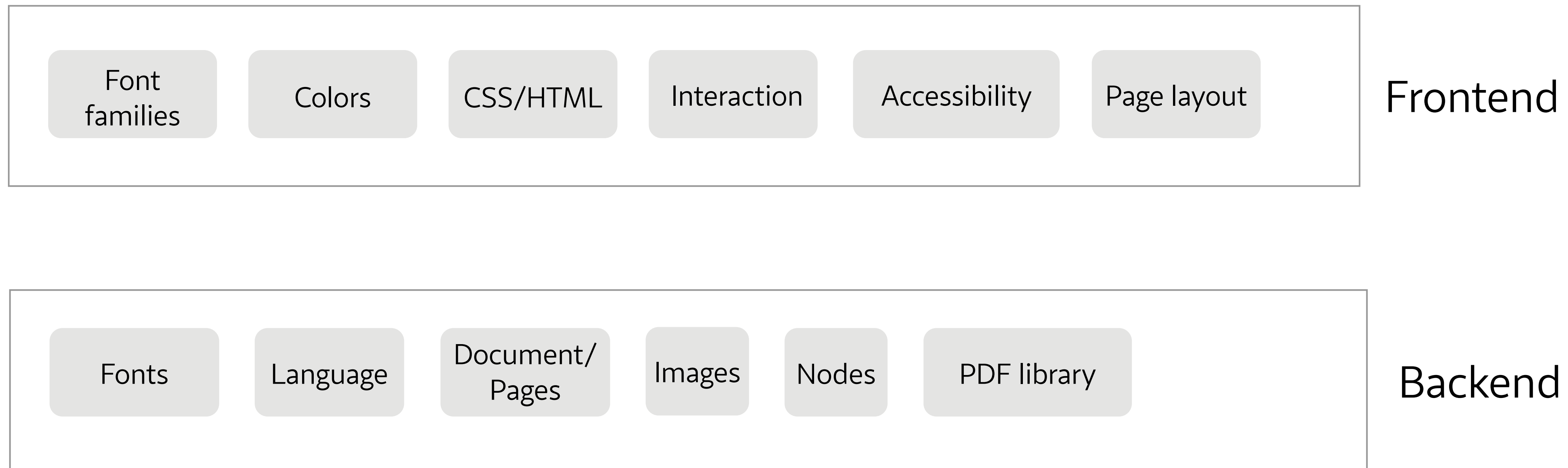
Architecture of boxes and glue

Architecture of boxes and glue



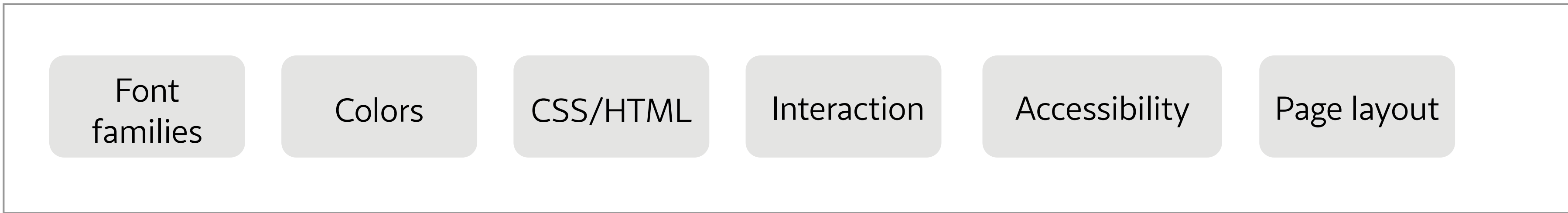
Backend

Architecture of boxes and glue

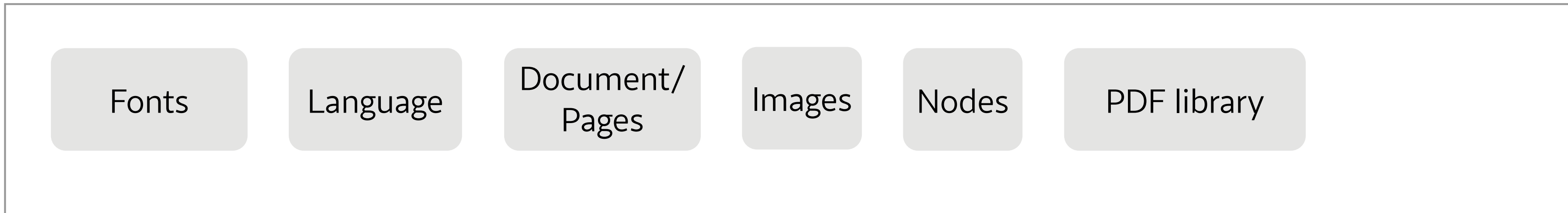


Architecture of boxes and glue

Application



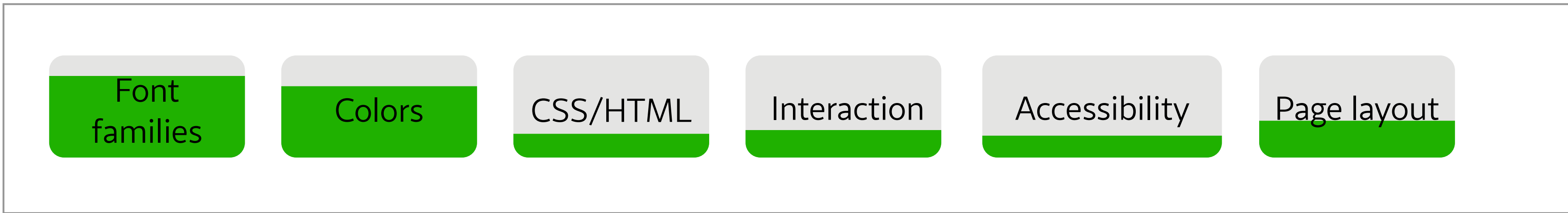
Frontend



Backend

Architecture of boxes and glue

Application



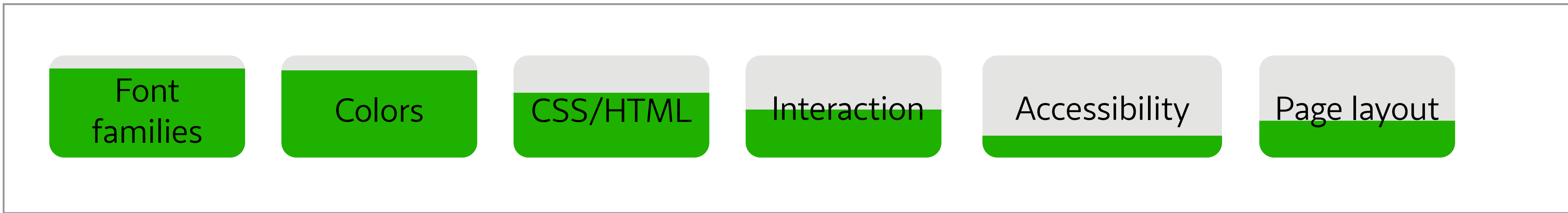
Frontend



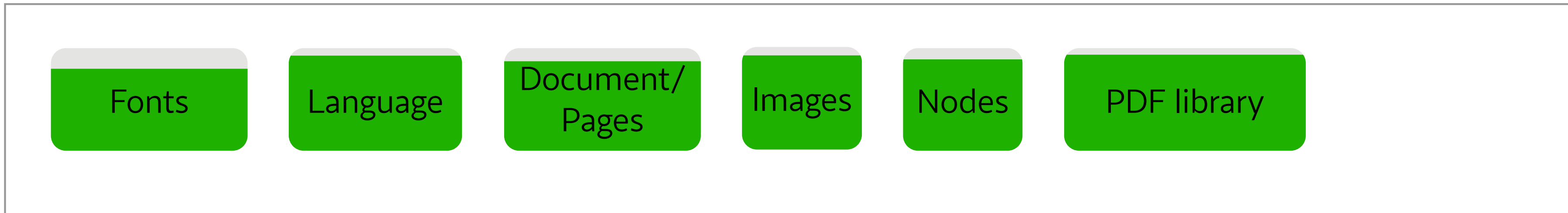
Backend

Architecture of boxes and glue

Application



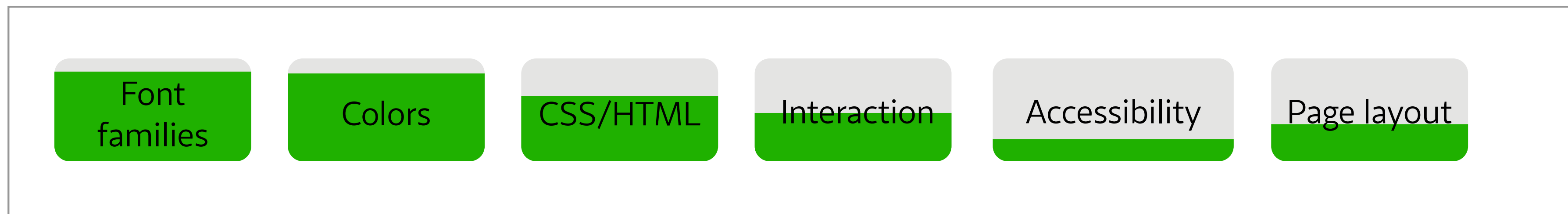
Frontend



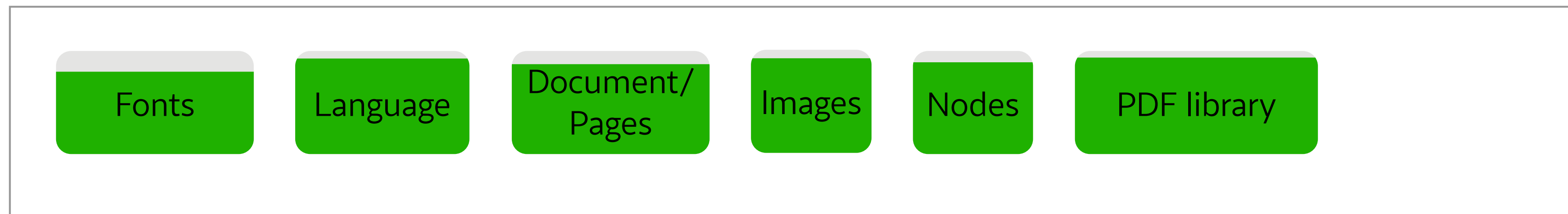
Backend

Architecture of boxes and glue

Application

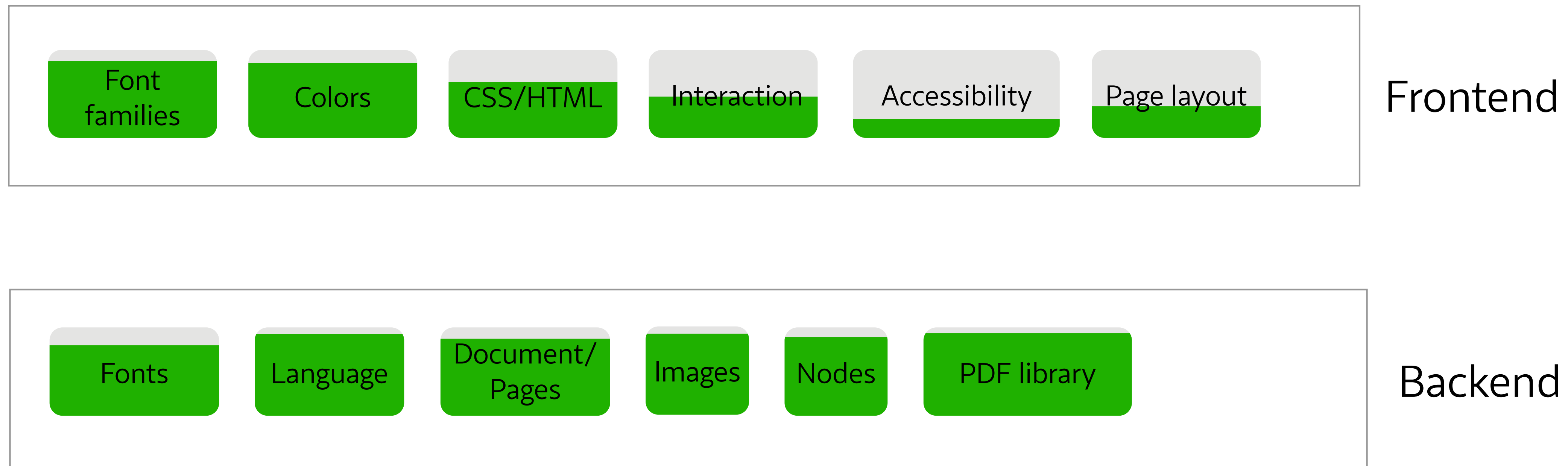
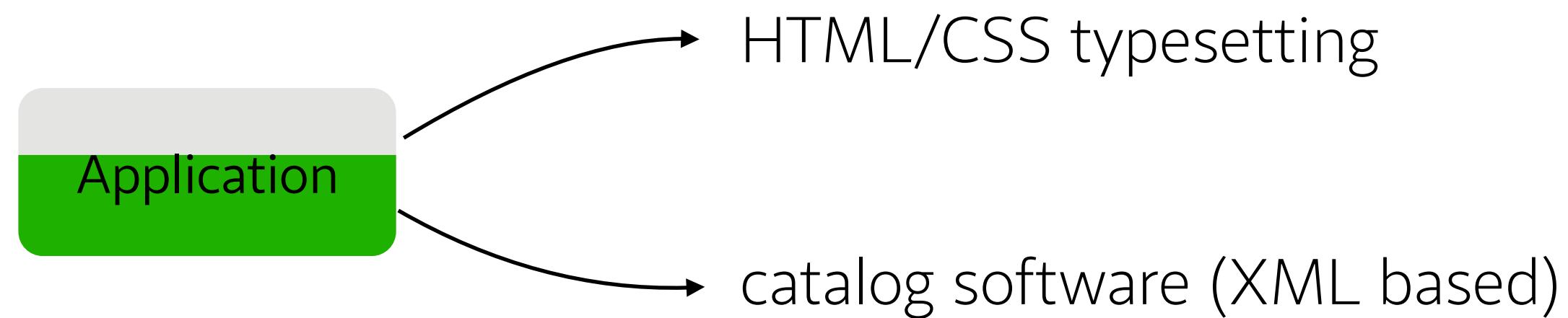


Frontend



Backend

Architecture of boxes and glue



Time line



raphink on Sep 8, 2016



Are you going to port LuaTeX to Go? 😄



pgundlach on Sep 8, 2016 Author Member



It's a long term project and only parts of TeX will be ported (only the algorithms and node stuff, but not the input language)



raphink on Sep 8, 2016



The input language is what I'd love to see disappear in TeX. After 10 years of using it, I still get headaches from trying to write simple logic in TeX packages. I'd love a new language with real modern expressivity.



Time line



Time line

web → go
experiments



Time line

web → go
experiments

first T_EX
algorithm



Time line

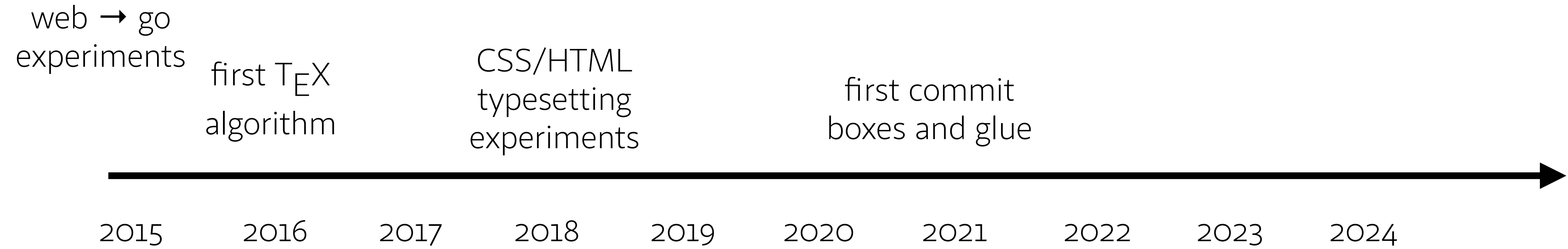
web → go
experiments

first T_EX
algorithm

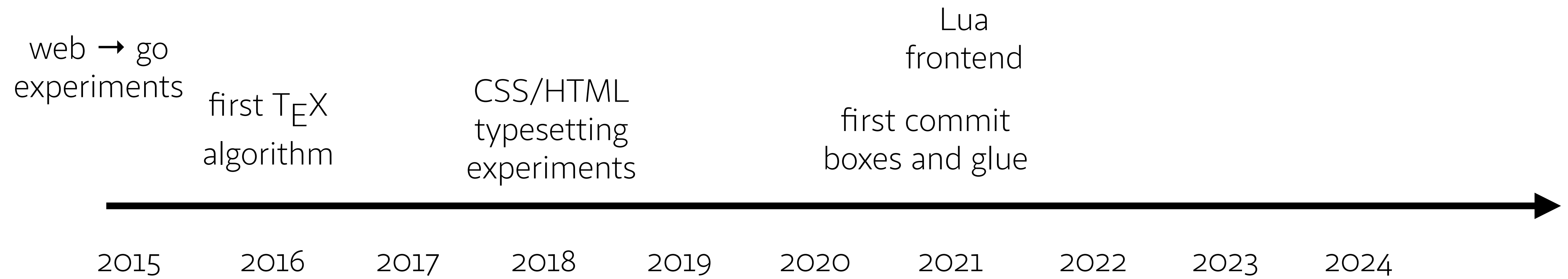
CSS/HTML
typesetting
experiments



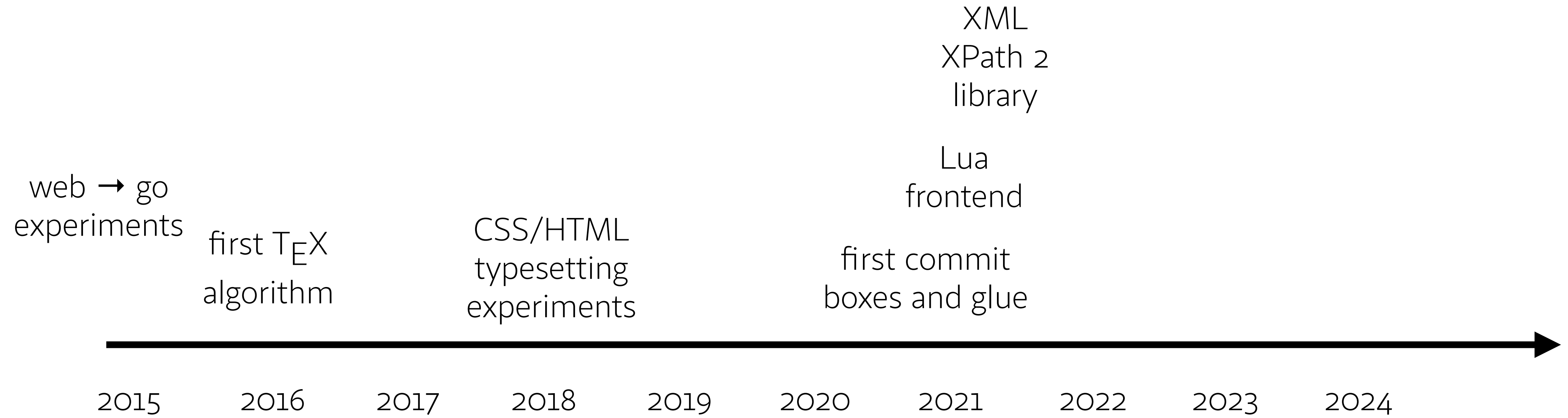
Time line



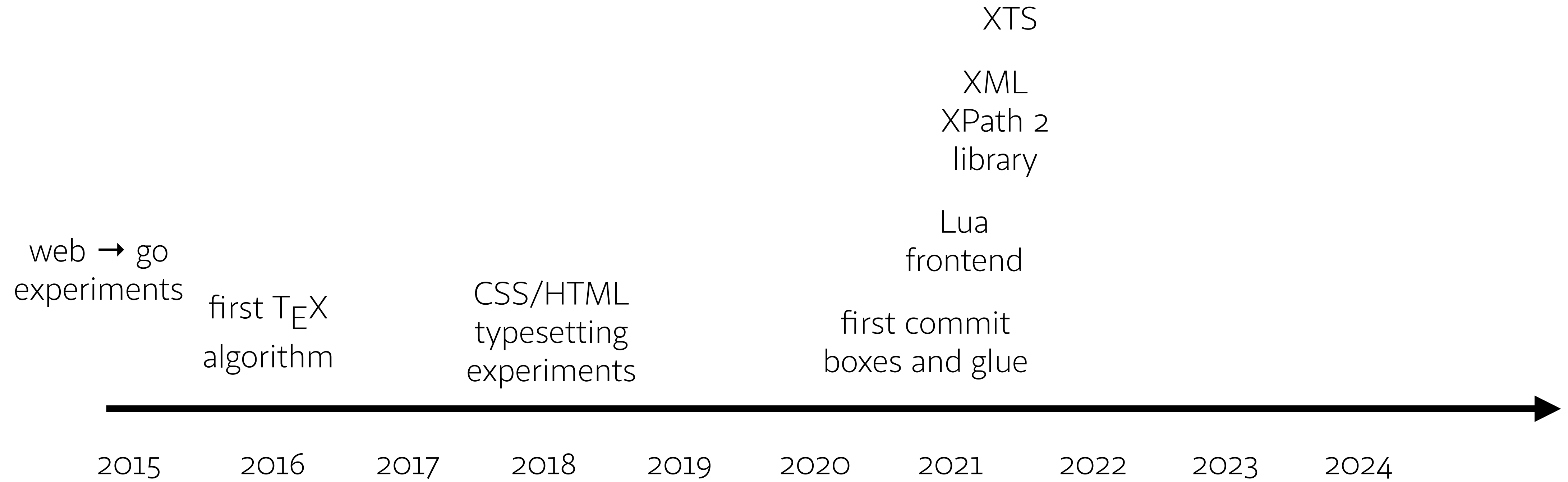
Time line



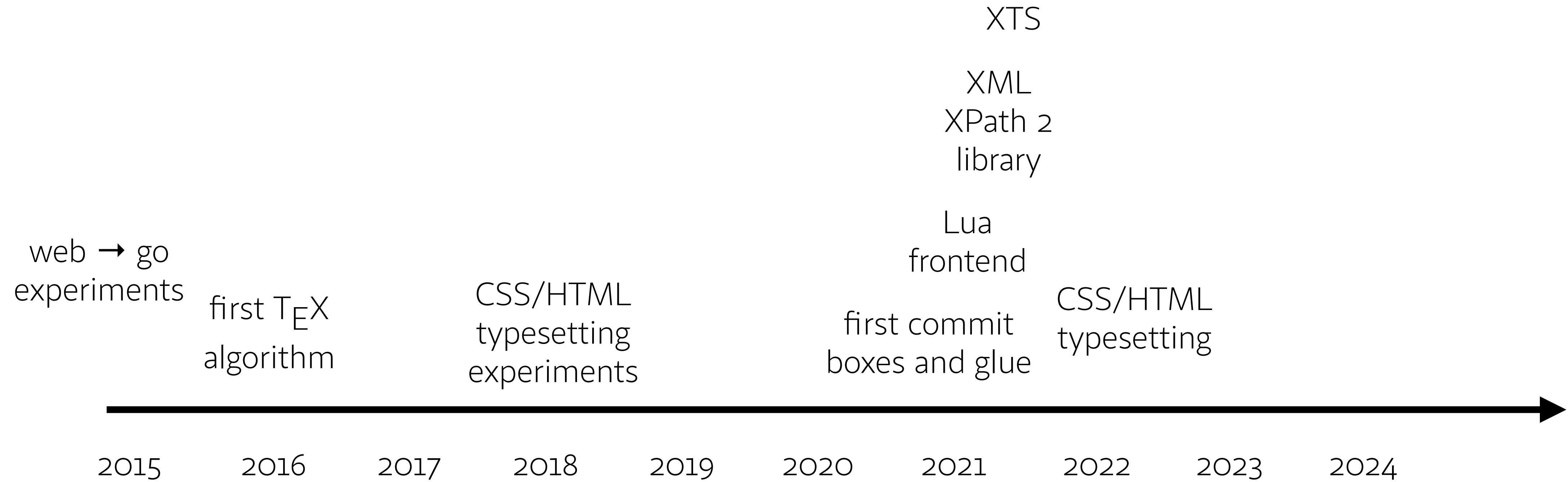
Time line



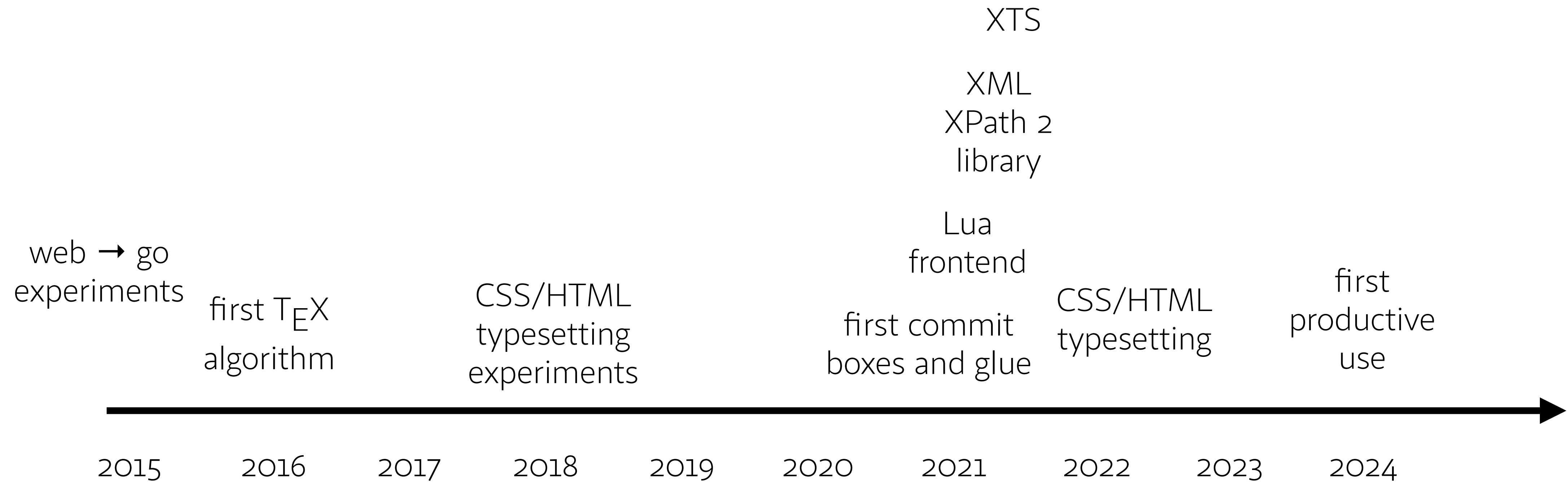
Time line



Time line



Time line



Next steps / wishes

Next steps / wishes

- ▣ Right to left (and mixed) typesetting

Next steps / wishes

- ▣ Right to left (and mixed) typesetting
- ▣ Graphics library like MetaPOST

Next steps / wishes

- ▣ Right to left (and mixed) typesetting
- ▣ Graphics library like MetaPOST
- ▣ Paragraph shape

Next steps / wishes

- ▣ Right to left (and mixed) typesetting
- ▣ Graphics library like MetaPOST
- ▣ Paragraph shape



boxes and glue: design goals

boxes and glue: design goals

- ▣ T_EX alike typography and output quality

boxes and glue: design goals

- ▣ T_EX alike typography and output quality
- ▣ Performance

boxes and glue: design goals

- ▣ T_EX alike typography and output quality
- ▣ Performance
- ▣ T_EX's data structures

boxes and glue: design goals

- ▣ T_EX alike typography and output quality
- ▣ Performance
- ▣ T_EX's data structures
- ▣ Arabic et. al. (Unicode, LTR/RTL, Bidi)

boxes and glue: design goals

- ▣ T_EX alike typography and output quality
- ▣ Performance
- ▣ T_EX's data structures
- ▣ Arabic et. al. (Unicode, LTR/RTL, Bidi)
- ▣ PDF standards

boxes and glue: design goals

- T_EX alike typography and output quality
- Performance
- T_EX's data structures
- Arabic et. al. (Unicode, LTR/RTL, Bidi)
- PDF standards

Lessons learned (so far)

Lessons learned (so far)

▣ It works!

Lessons learned (so far)

- ▣ It works!
- ▣ Writing PDF can be difficult

Lessons learned (so far)

- ▣ It works!
- ▣ Writing PDF can be difficult
- ▣ Err and err and err again

It works!

In olden times when wishing still helped one, there lived a king whose daughters were all beautiful; and the youngest was so beautiful that the sun itself, which has seen so much, was astonished whenever it shone in her face. Close by the king's castle lay a great dark forest, and under an old lime-tree in the forest was a well, and when the day was very warm, the king's child went out into the forest and sat down by the side of the cool fountain; and when she was bored she took a golden ball, and threw it up on high and caught it; and this ball was her favorite plaything.

The frog king

In olden times when wishing still *helped* one, there lived a king whose daughters were all beautiful, but the *youngest* was so beautiful that the sun itself, which has seen so much, was *astonished* whenever it shone in her face.

Close by the king's castle lay a great dark forest, and under an old lime-tree in the forest was a well, and when the day was very warm, the king's child went out into the forest and sat down by the side of the cool *fountain*, and when she was bored she took a golden ball, and threw it up on high and caught it, and this ball was her favorite plaything.

- Now it so happened that on one occasion the princess's golden ball did not fall into the little hand which she was holding up for it,
 - but on to the ground beyond, and rolled straight into the water.
1. The king's daughter followed it with her eyes, but it vanished, and the well was deep, so deep that the bottom could not be seen.
 2. At this she began to cry, and cried louder and louder, and could not be comforted.

```
<Layout xmlns="urn:speedata.de:2009/publisher/en"
  xmlns:sd="urn:speedata:2009/publisher/functions/en">

  <Record element="data">
    <ProcessNode select="article"/>
    <Message select="$nr"/>
  </Record>

  <Record element="article">
    <SetVariable variable="nr" select="@number"/>
  </Record>
</Layout>
```



Print Company & Office
61556 W
20th Ave
Seattle King
WA 98104
206-711-6498
206-395-6284
jbiddy@printcompany.com
www.printcompany.com

James Butt
6649 N Blue Gum St
New Orleans, LA 70116

November 6, 2014

Dear James Butt,

but I must explain to you how all this mistaken idea of denouncing pleasure and praising pain was born and I will give you a complete account of the system, and expound the actual teachings of the great explorer of the truth, the master-builder of human happiness. No one rejects, dislikes, or avoids pleasure itself, because it is pleasure, but because those who do not know how to pursue pleasure rationally encounter consequences that are extremely painful. Nor again is there anyone who loves or pursues or desires to obtain pain of itself, because it is pain, but because occasionally circumstances occur in which toil and pain can procure him some great pleasure. To take a trivial example, which of us ever undertakes laborious physical exercise, except to obtain some advantage from it? But who has any right to find fault with a man who chooses to enjoy a pleasure that has no annoying consequences, or one who avoids a pain that produces no resultant pleasure?

Yours faithfully,

Jani Biddy

Lessons learned (so far)

- ▣ It works!
- ▣ **Writing PDF can be difficult**
- ▣ Err and err and err again

Writing PDF can be difficult

Writing PDF can be difficult

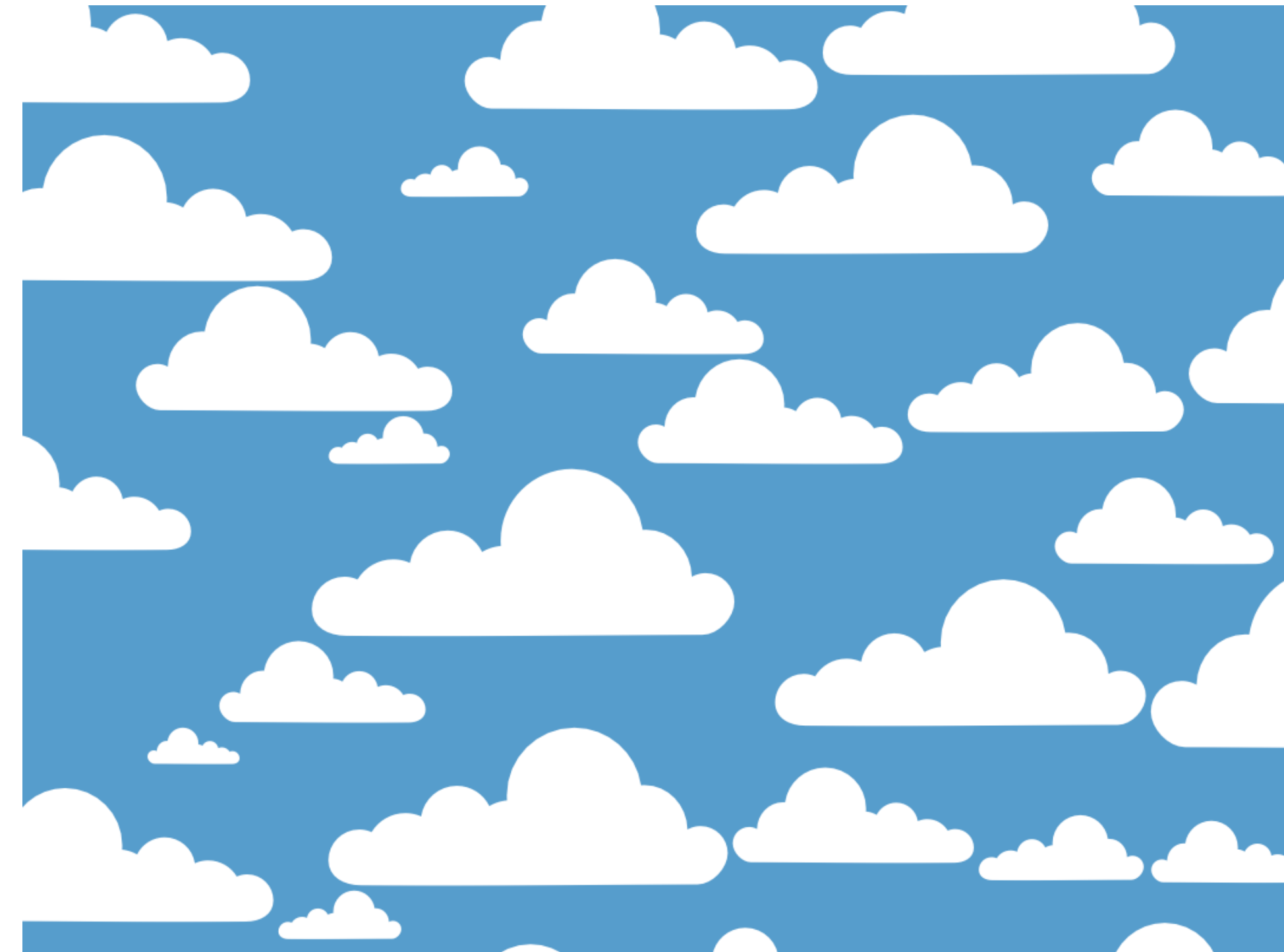
- ▣ Looks like an innocent file format
(plain text + some binary data)

Writing PDF can be difficult

- ▣ Looks like an innocent file format (plain text + some binary data)
- ▣ Spec has more than 1000 pages

“Innocent file format”

```
5 0 obj
<<
  /Type /XObject
  /Subtype /Image
  /DecodeParms << /Columns 1072 /Predictor 15 >>
  /BitsPerComponent 8
  /ColorSpace [ /Indexed /DeviceRGB 128 6 0 R ]
  /Filter /FlateDecode
  /Height 804
  /Length 22877
  /Width 1072
>>
stream
...
  <compressed data>
...
endstream
endobj
```



“Innocent file format”



“Innocent file format”

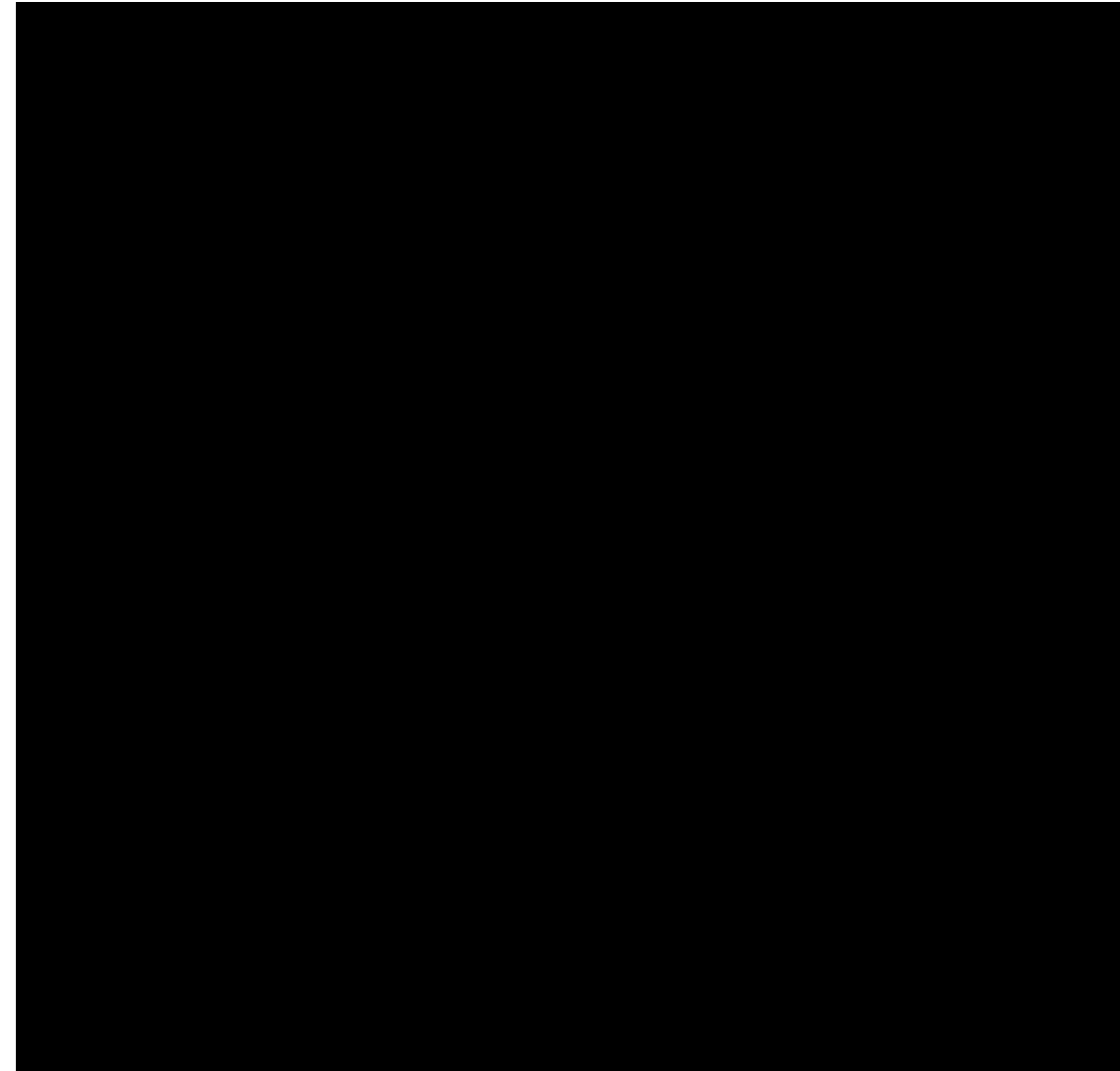


“Innocent file format”

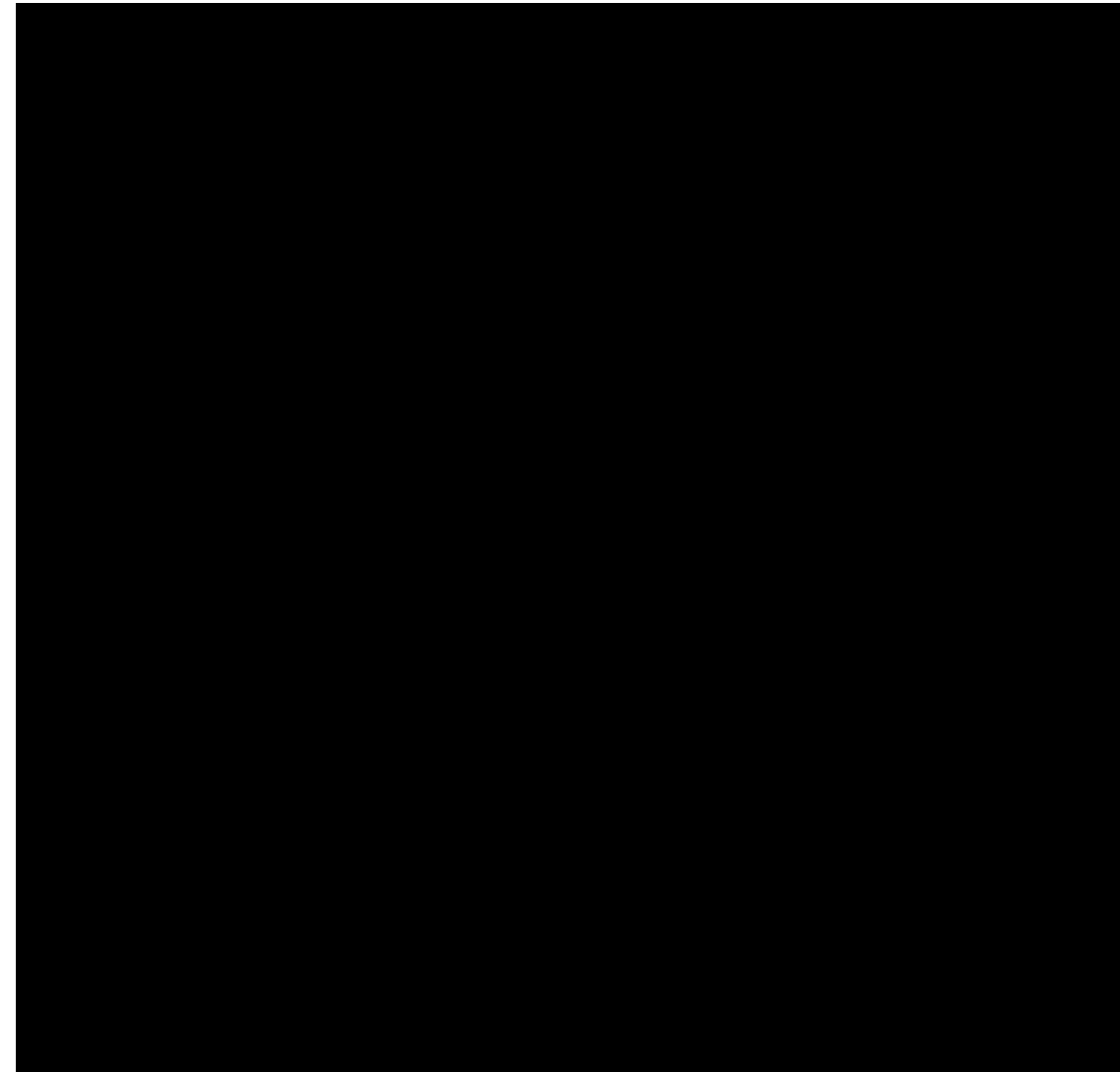
“Innocent file format”



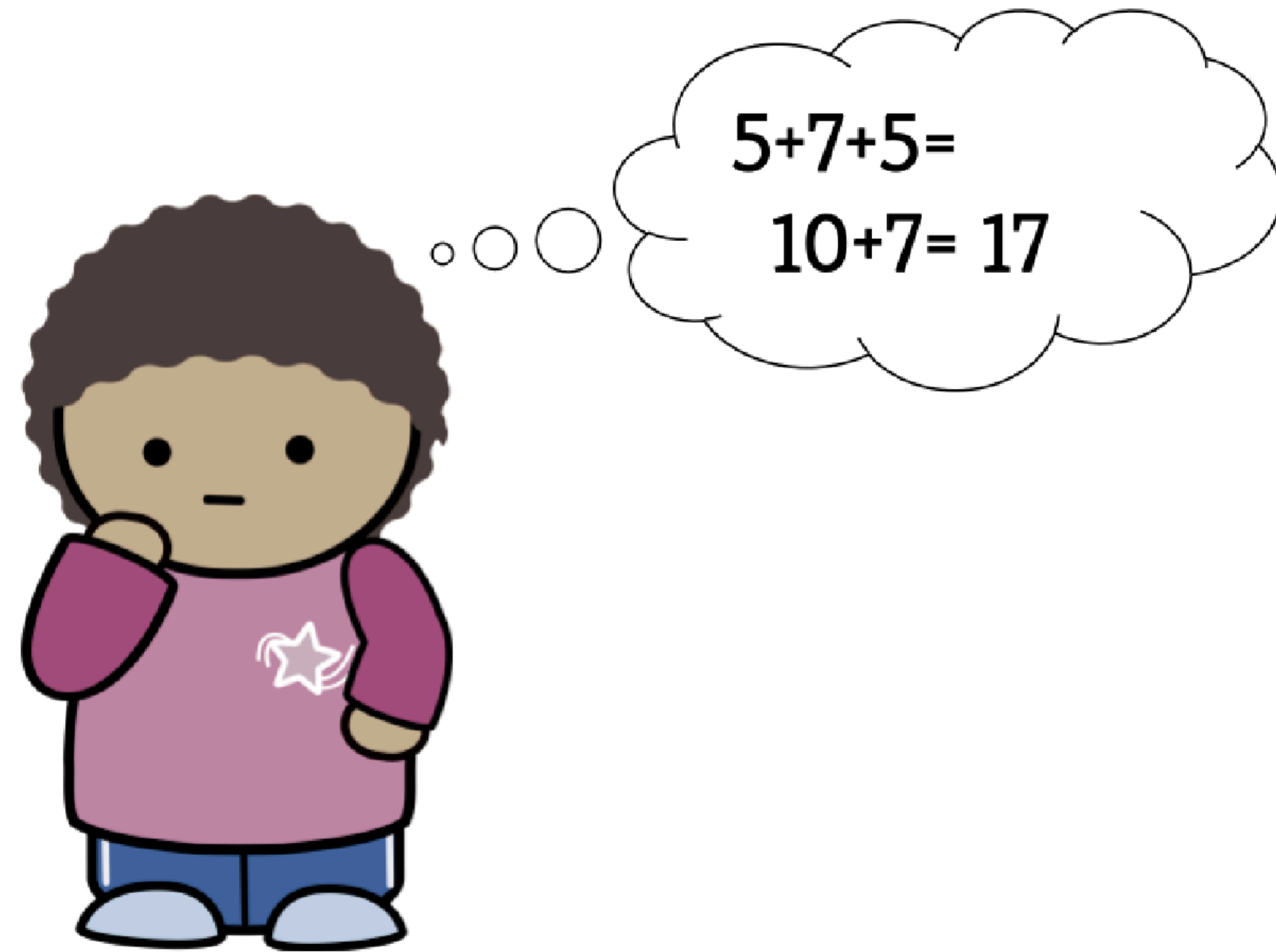
“Innocent file format”



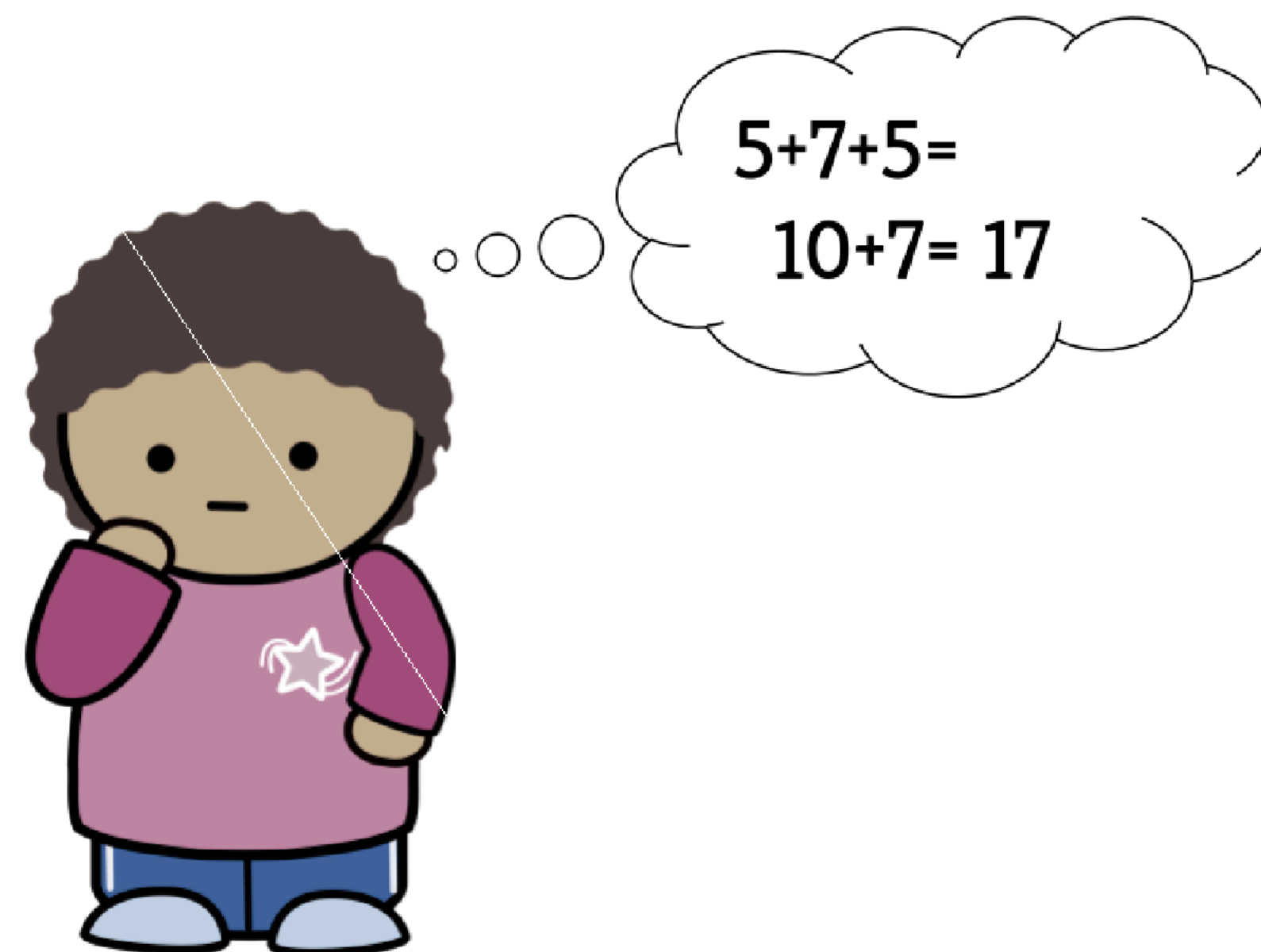
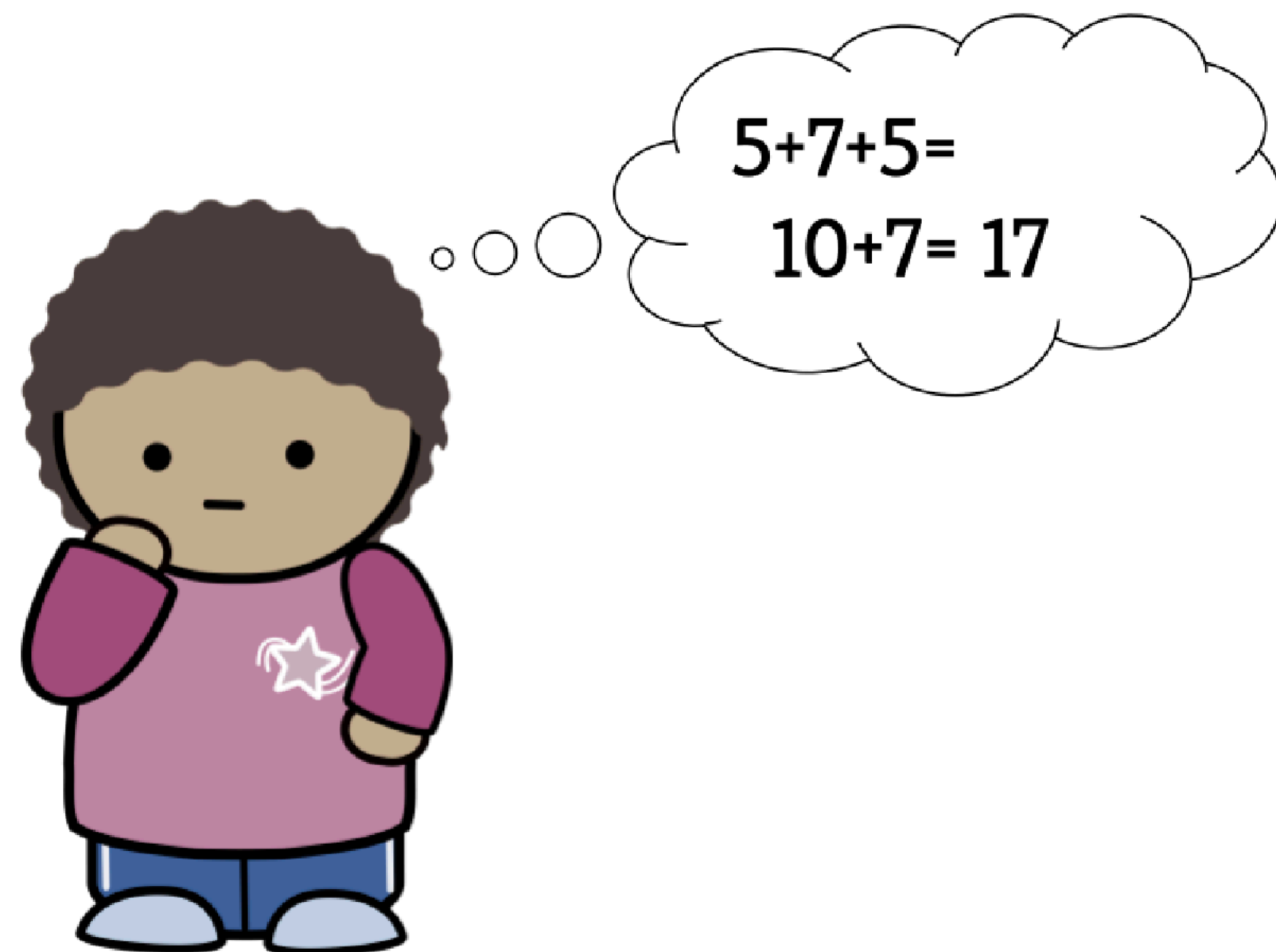
“Innocent file format”



“Innocent file format”



“Innocent file format”



Lessons learned (so far)

- ▣ It works!
- ▣ Writing PDF can be difficult
- ▣ **Err and err and err again**

Lessons learned (so far)

- ▣ It works!
- ▣ Writing PDF can be difficult
- ▣ **Err and err and err again**

but not less and less and less

Different kind of errors

Cell 1	Cell 2	Cell 3
Cell 4	Cell 5	Cell 6

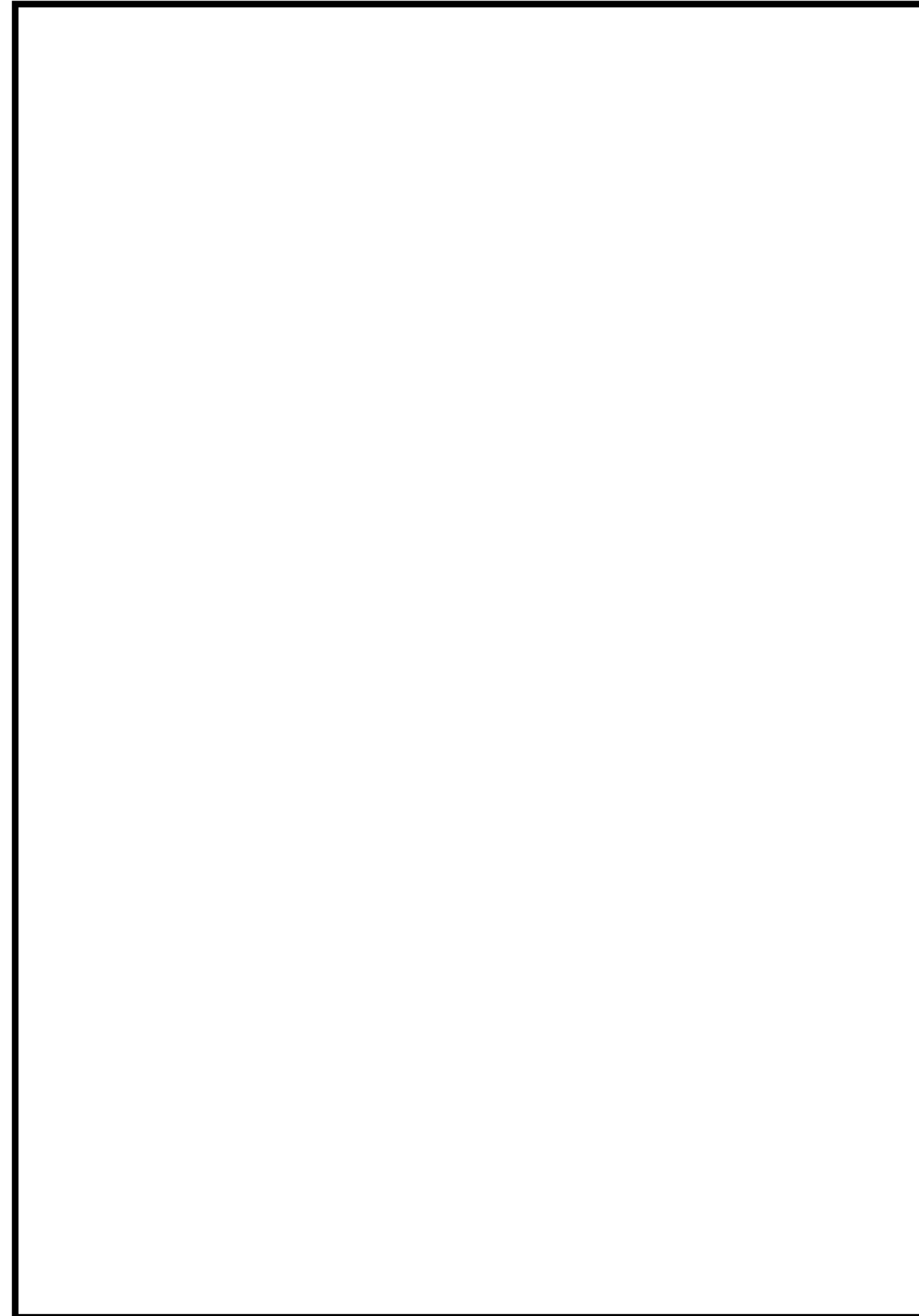
Different kind of errors

Cell 1	Cell 2	Cell 3
Cell 4	Cell 5	Cell 6

In olden times when
 here lived a king
 beautiful; and he
 had the sun in self,
 was astonished when

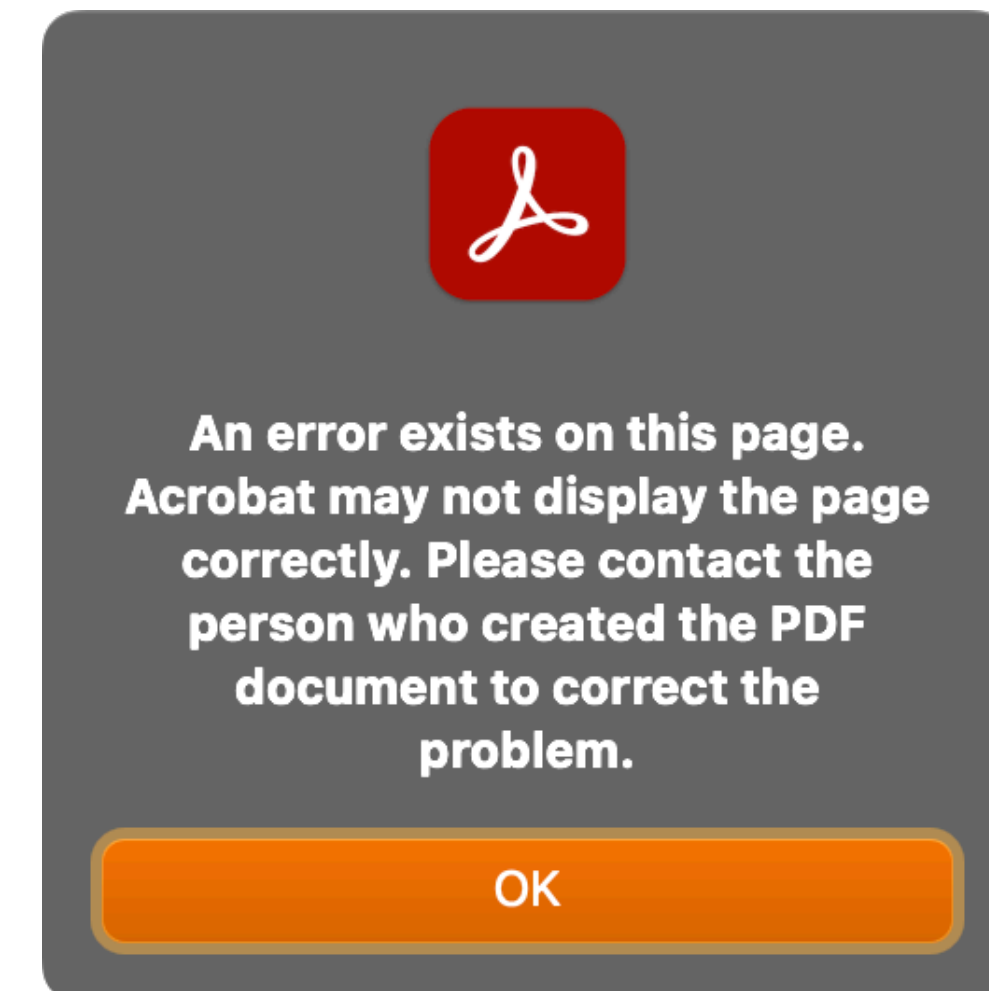
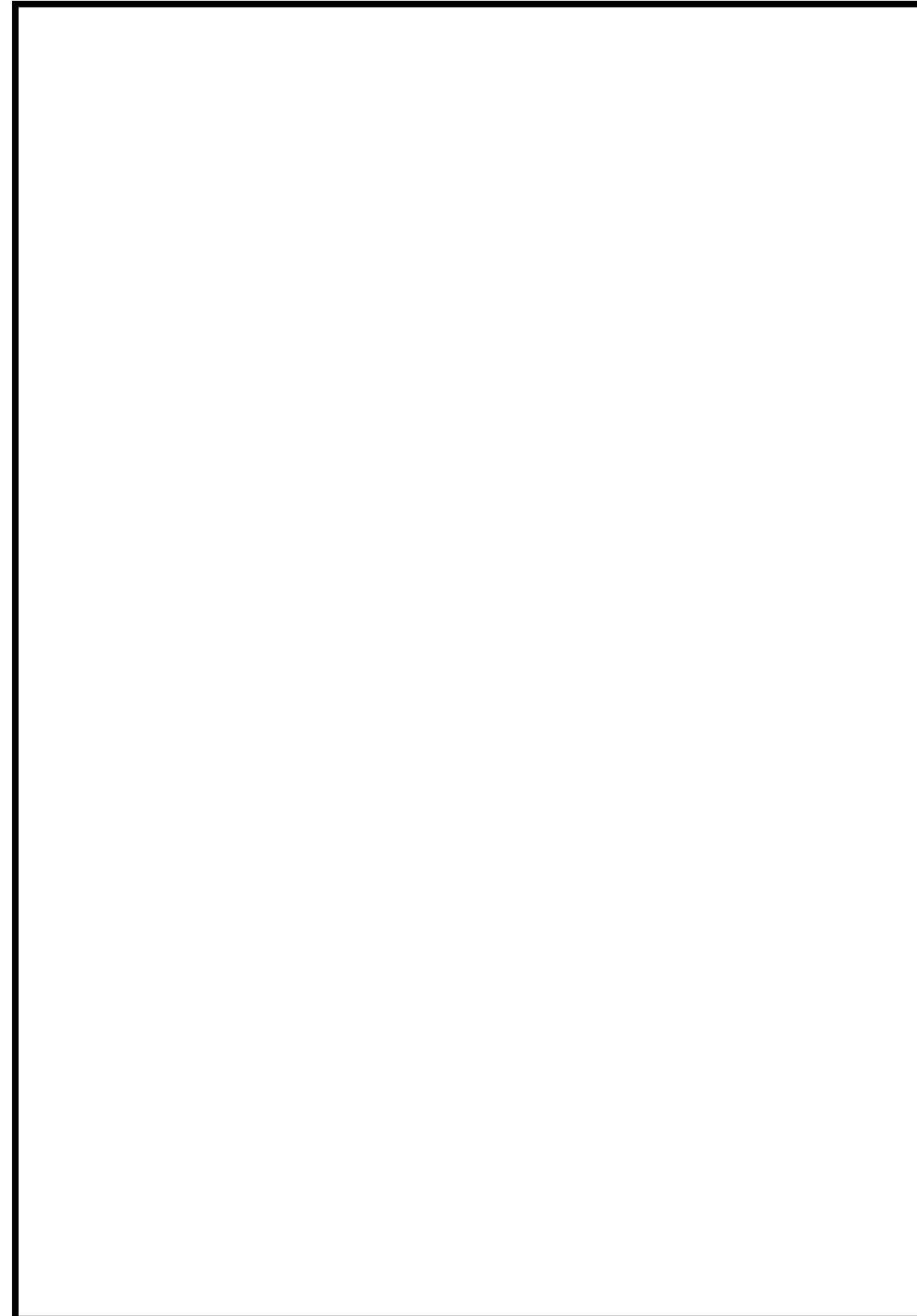
Different kind of errors

(empty PDF)



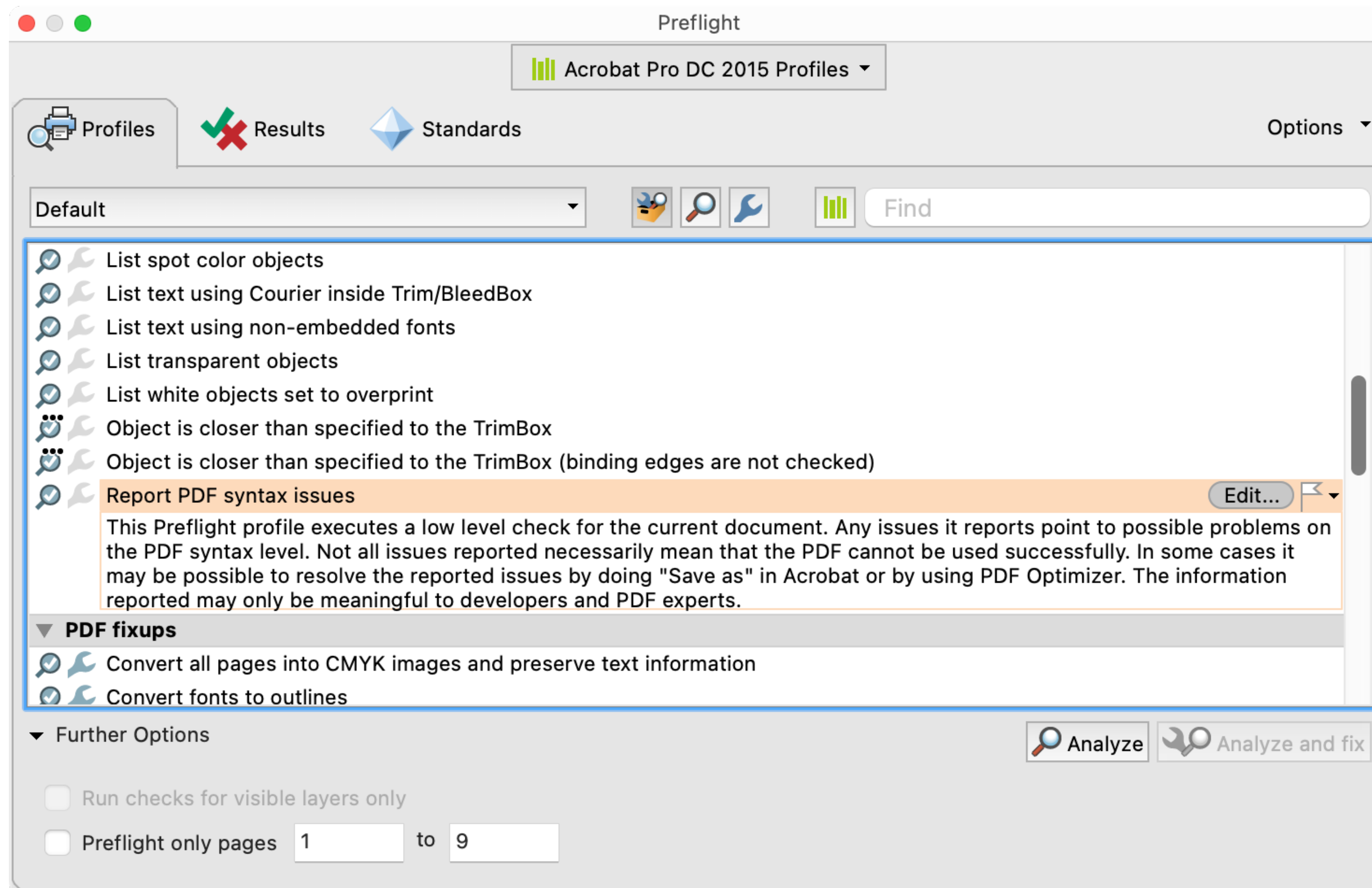
Different kind of errors

(empty PDF)



Writing PDF (error checking)

Writing PDF (error checking)



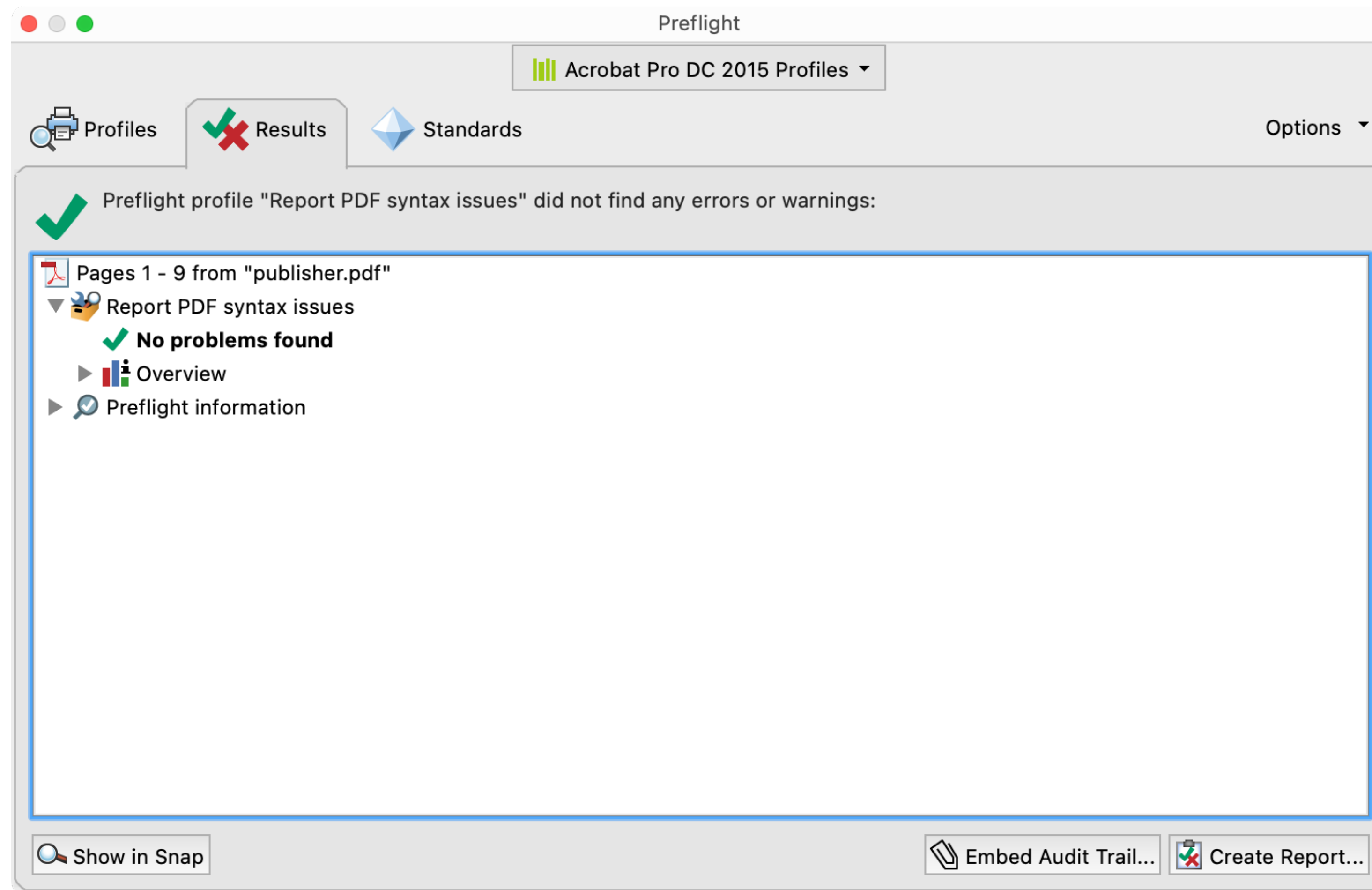
Writing PDF (error checking)

The screenshot shows the Adobe Acrobat Pro DC Preflight window. The title bar reads "Preflight" and the menu bar includes "Acrobat Pro DC 2015 Profiles". The main area displays the results of a preflight check using the "Report PDF syntax issues" profile. A red 'X' icon indicates that errors were found. The results are organized into a tree view:

- Page 1 from "publisher.pdf"
 - Report PDF syntax issues
 - Page Description Errors
 - Error in PDF syntax
 - Trigger values
 - Unable to analyze the content stream on page 1; parse error: "Too few operands."
 - Invalid operand(s) for operator (1 match on 1 page)
 - Page 1 (cm): 21.0 by 29.7
 - Page details
 - MediaBox (cm): 21.0x29.7 (0.0/0.0/21.0/29.7)
 - TrimBox (cm): 21.0x29.7 (0.0/0.0/21.0/29.7)
 - Page scaling factor (UserUnits): Not present
 - Page rotation (degrees counterclockwise) 0
 - Thumbnail: Not present
 - Annotations: Not present
 - Page is a separated plate: No
 - Page Description Errors
 - Invalid operand(s) for operator
 - Operator Name: Tm
 - Stream State: TextObject
 - Operators Read: 3
 - Traversal Path: ->Pages->Kids->[0]
 - Invalid operand(s) for operator
 - Operator Name: Tm
 - Stream State: TextObject
 - Operators Read: 7
 - Traversal Path: ->Pages->Kids->[0]
 - Invalid operand(s) for operator
 - Trigger values
 - Overview
 - Preflight information

At the bottom of the window, there are buttons for "Show in Snap", "Embed Audit Trail...", and "Create Report...".

Writing PDF (error checking)



Tools for PDF debugging

Tools for PDF debugging

- ▣ Adobe Acrobat

Tools for PDF debugging

- ▣ Adobe Acrobat

- ▣ less

(use qpdf to decompress):

```
qpdf --qdf --object-streams=disable in.pdf out.pdf
```

Tools for PDF debugging

- ▣ Adobe Acrobat
- ▣ less
(use qpdf to decompress):
qpdf --qdf --object-streams=disable in.pdf out.pdf
- ▣ veraPDF (OpenSource)

Tools for PDF debugging

- ▣ Adobe Acrobat
- ▣ less
(use qpdf to decompress):
qpdf --qdf --object-streams=disable in.pdf out.pdf
- ▣ veraPDF (OpenSource)
- ▣ PAC (PDF Accessibility Checker) from PDF/UA foundation

Tools for PDF debugging

- ▣ Adobe Acrobat
- ▣ less
(use qpdf to decompress):
qpdf --qdf --object-streams=disable in.pdf out.pdf
- ▣ veraPDF (OpenSource)
- ▣ PAC (PDF Accessibility Checker) from PDF/UA foundation

Development cycle

implement feature
and write PDF → open in Adobe Acrobat → check for errors



Debugging...

Debugging...

Cell 1	Cell 2	Cell 3
Cell 4	Cell 5	Cell 6

Debugging...

- ▣ Use a step by step debugger

Debugging...

- Use a step by step debugger
- Visual debugging

1 □ A short story.

A wonderful serenity has taken possession of my entire soul, like these sweet mornings of spring which I enjoy with my whole heart. I am alone, and feel the charm of existence in this spot, which was created for the bliss of souls like mine. I am so happy, my dear friend, so absorbed in the exquisite sense of mere tranquil existence, that I neglect my talents¹.

- one.

- two.

- three.

TEX

Hello.

World.

Debugging...

- ▣ Use a step by step debugger
- ▣ Visual debugging
- ▣ “Printf” debugging

```
a := readInput()  
printf("input is:", a)  
doSomething(a)  
...
```

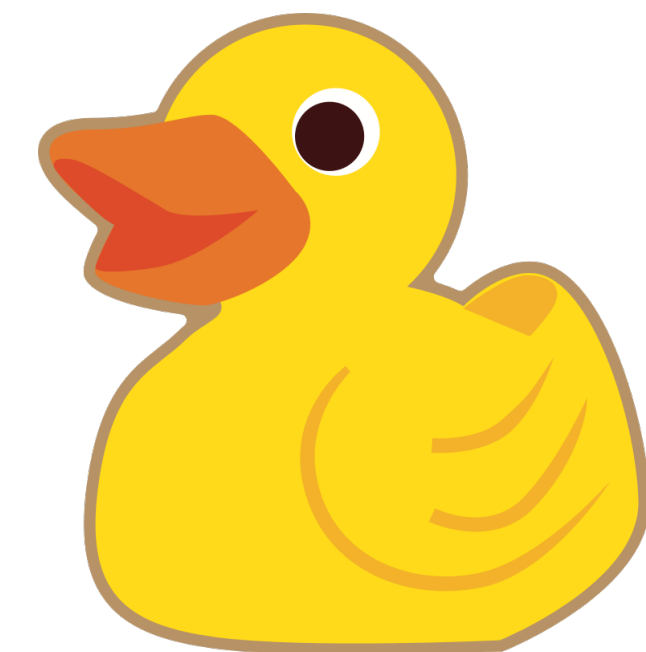
Debugging...

- ▣ Use a step by step debugger
- ▣ Visual debugging
- ▣ “Printf” debugging
- ▣ Rubber duck debugging

Debugging...

- Use a step by step debugger
- Visual debugging
- “Printf” debugging
- Rubber duck debugging

```
    dmin < math.MaxInt {  
        lb.appendBreakpointHere(n, dmin, dc, ac, .  
    }  
    }  
    if dmin == math.MaxInt && lb.activeNodesA == nil {  
        W, E, Y, Z := lb.computeSum(n)  
        lastInactive := lb.inactiveNodesP  
        width := lb.sumW  
        var pre Node  
        switch v := n.(type) {  
        case *Penalty:  
            width += v.Width  
        case *Disc:  
            width += 5 * bag.Factor  
            pre = v.Pre
```



Printf debugging...

Printf debugging...

- ▣ Works only for simple cases

Printf debugging...

- ▣ Works only for simple cases
- ▣ Too much output for data structures used in typesetting

Printf debugging...

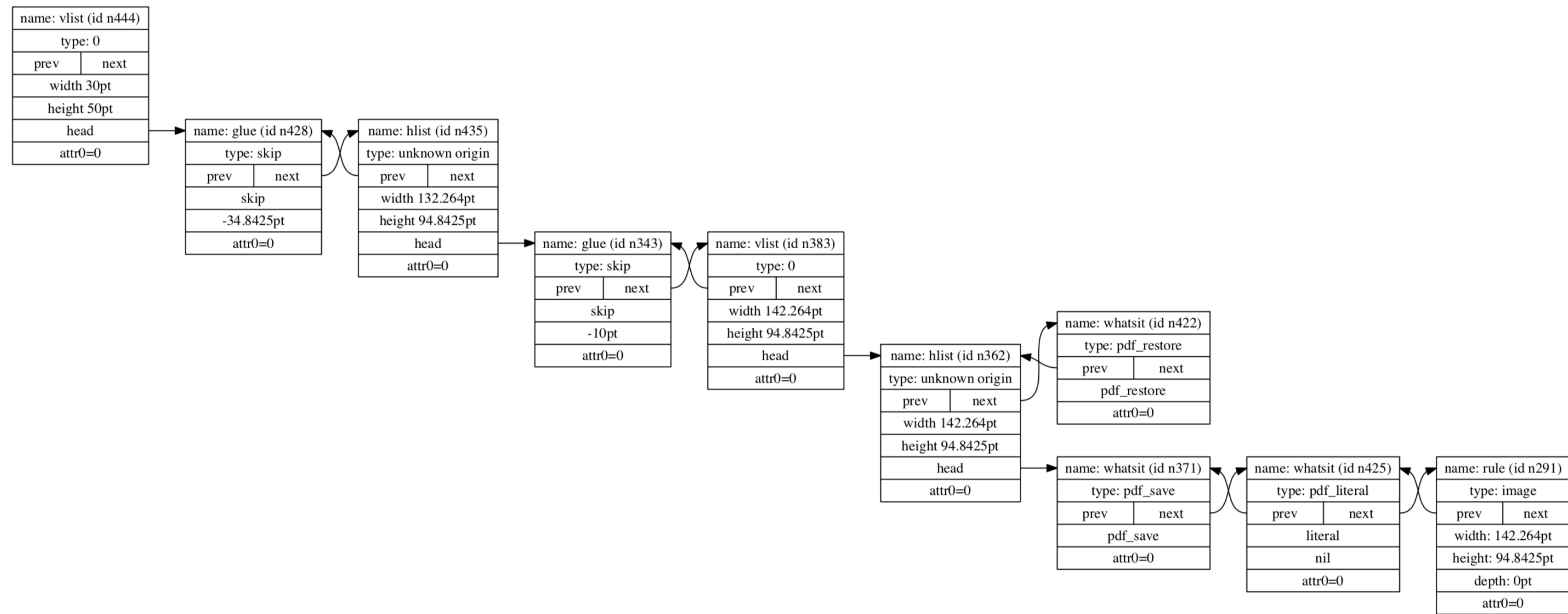
- ❑ Works only for simple cases
- ❑ Too much output for data structures used in typesetting
- ❑ How do I understand the nested node lists?

Printf debugging...

- ❑ Works only for simple cases
- ❑ Too much output for data structures used in typesetting
- ❑ How do I understand the nested node lists?

Structured output is important!

Node debugging



viz nodelist

Node debugging

```

└─VLIST dir: TLT
  └─head:
    └─WHAT SIT subtype: pdf_literal, mode: 1, data: data
      └─VLIST
        └─VLIST dir: TLT
          └─head:
            └─GLUE width: 28.35pt
              └─HLIST width: 566.94pt, depth: 3pt, height: 9pt, dir: TLT
                └─head:
                  └─GLUE width: 28.35pt
                    └─VLIST width: 538.59pt, depth: 3pt, height: 9pt, dir: TLT
                      └─head:
                        └─VLIST width: 538.59pt, depth: 3pt, height: 9pt, dir: TLT
                          properties: {'origin' = par:format}
                            └─head:
                              └─HLIST subtype: line, width: 538.59pt, depth: 3pt, height: 9pt, dir: TLT, attr: 405=0
                                └─head:
                                  └─RULE depth: 3pt, height: 9pt, attr: fontfamily=2
                                    properties: {'kernafter' = -1973.451}
                                  └─GLYPH char: fi, width: 5.35pt, height: 7.12pt, attr: fontfamily=2
                                    properties: {'kernafter' = -1973.451}
                                  └─KERN kern: -0.03pt
                                    properties: {'kernafter' = -2631.268}
                                  └─GLYPH char: s, width: 3.67pt, height: 4.48pt, depth: 0.12pt, attr: fontfamily=2
                                    properties: {'kernafter' = -2631.268}
                                  └─KERN kern: -0.04pt
                                  └─GLYPH char: h, width: 5.34pt, height: 7.12pt, attr: fontfamily=2
                                  └─PENALTY penalty: 10000
                                    properties: {'origin' = finishpar}
                                  └─GLUE subtype: parfillskip, stretch: +1fil
                                  └─GLUE subtype: rightskip

```

nodetree
(by Josef Friedrich)

Structured debugging output

```

<vlist id="567" wd="538.58" ht="48" dp="0" origin="textblock">
  <vlist id="566" wd="538.58" ht="48" dp="0">
    <hlist id="564" wd="538.58" ht="48" dp="0" r="1" origin="line">
      <glue id="563" wd="0" stretch="0" stretchorder="0" shrink="0" shrinkorder="0" subtype="0"/>
      <vlist id="556" wd="538.58" ht="48" dp="0">
        <vlist id="555" wd="538.58" ht="48" dp="0" origin="prepend in HTML mode" x="0" y="0">
          <hlist id="553" wd="538.58" ht="8.02" dp="1.98" r="15.671976806286716" origin="line">
            <glue id="552" wd="0" stretch="0" stretchorder="0" shrink="0" shrinkorder="0" subtype="0"/>
            <glyph id="4" components="L" wd="5.56" ht="8.02" dp="1.98" codepoint="72" face="0"/>
            <glyph id="5" components="o" wd="5.56" ht="8.02" dp="1.98" codepoint="82" face="0"/>
            <glyph id="6" components="r" wd="3.33" ht="8.02" dp="1.98" codepoint="97" face="0"/>
            <kern id="7" kern="-0.1"/>
            ...
            <glyph id="482" components="." wd="2.78" ht="8.02" dp="1.98" codepoint="89" face="0"/>
            <penalty id="533" penalty="10000" width="0"/>
            <glue id="534" wd="0" stretch="1" stretchorder="1" shrink="0" shrinkorder="0" subtype="0"/>
            <glue id="539" wd="165.58" stretch="1" stretchorder="3" shrink="0" shrinkorder="0" subtype="0"/>
          </hlist>
          <glue id="554" wd="2" stretch="0" stretchorder="0" shrink="0" shrinkorder="0" subtype="0" origin="last lineskip"/>
        </vlist>
      </vlist>
      <penalty id="557" penalty="10000" width="0"/>
      <glue id="558" wd="1" stretch="1" stretchorder="1" shrink="0" shrinkorder="0" subtype="0"/>
      <glue id="562" wd="0" stretch="0" stretchorder="0" shrink="0" shrinkorder="0" subtype="0"/>
    </hlist>
    <glue id="565" wd="0" stretch="0" stretchorder="0" shrink="0" shrinkorder="0" subtype="0" origin="last lineskip"/>
  </vlist>
</vlist>

```

Next steps (from last year)

Next steps (from last year)

- ▣ Experiment with the algorithms

Next steps (from last year)

- ▣ Experiment with the algorithms
- ▣ Optimizations for page break and paragraph break

Next steps (from last year)

- ▣ Experiment with the algorithms
- ▣ Optimizations for page break and paragraph break
- ▣ Parallel tasks

Conclusion (1)

Conclusion (1)

- ▣ Porting T_EX algorithms and data structures is possible

Conclusion (1)

- ▣ Porting T_EX algorithms and data structures is possible
- ▣ Development takes much more time than estimated

Conclusion (1)

- ▣ Porting T_EX algorithms and data structures is possible
- ▣ Development takes much more time than estimated
- ▣ Seeing the results keeps my motivation high

Conclusion (2)

“Roses are red
Violets are blue
I create my PDF
With boxes and glue

Homepage <https://boxesandglue.dev>

GitHub <https://github.com/speedata/boxesandglue>

Mastodon [@boxesandglue@typo.social](https://typo.social/@boxesandglue)