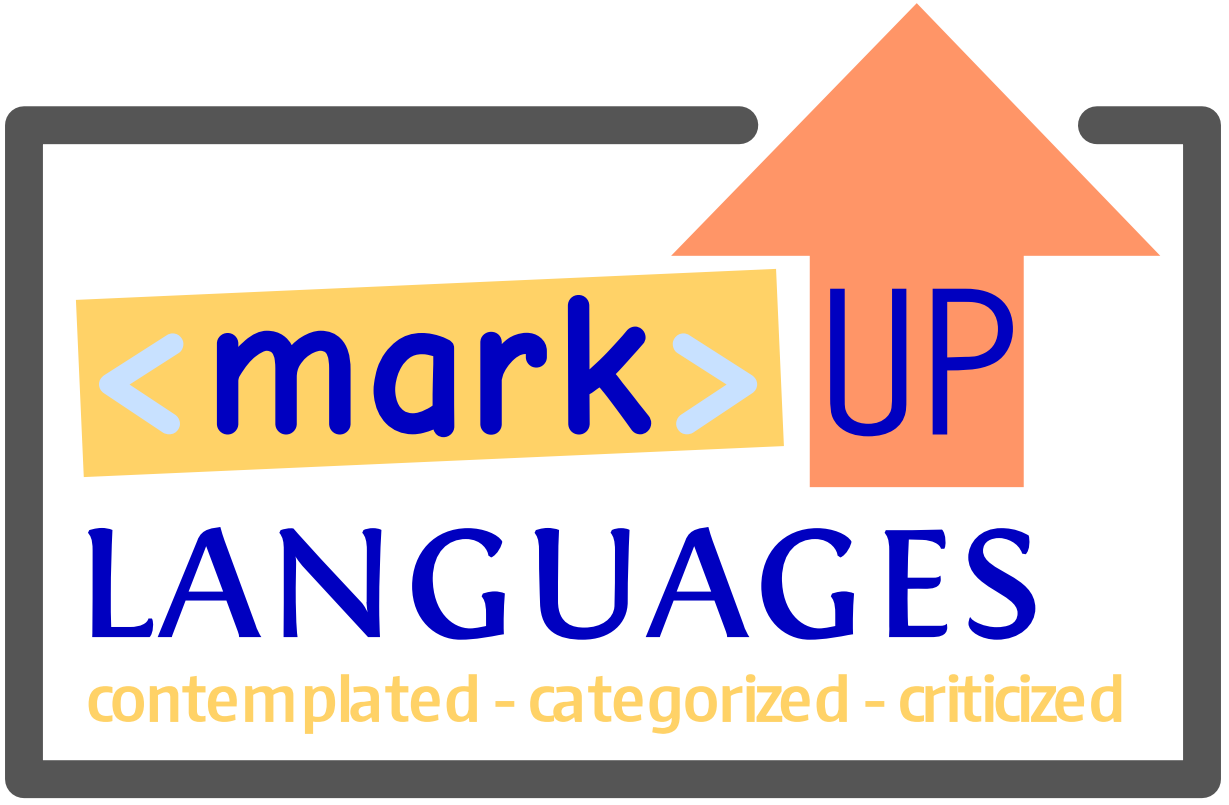
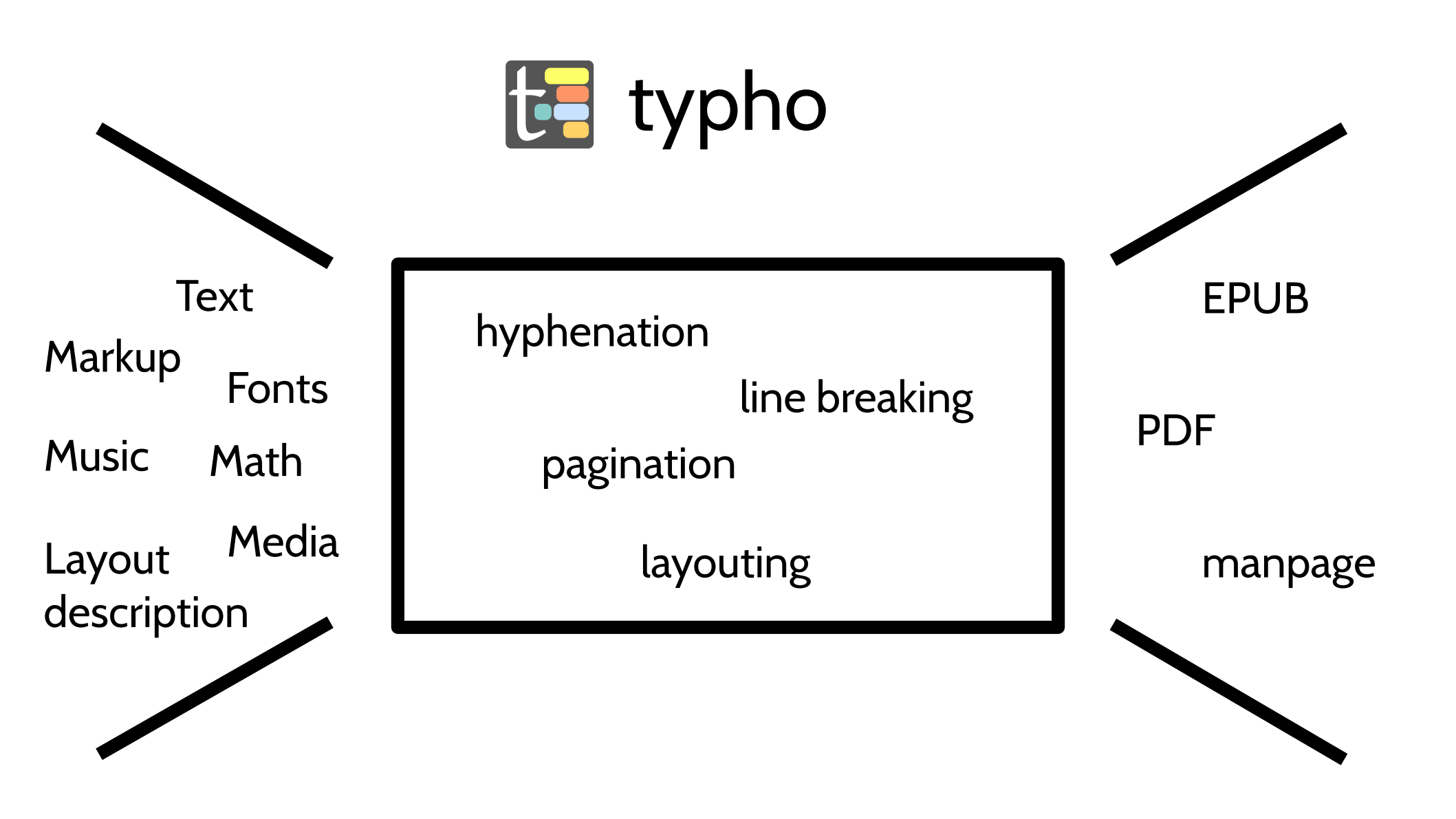


https://lukas-prokop.at/talks/glt24-markup-languages/

Grazer LINUXTAGE

2024-04-07 part of project typho.org


typho
Text
Markup
Fonts
Music
Math
Layout description
Media
hyphenation
line breaking
pagination
layouting
EPUB
PDF
manpage

What is a markup language?

What is a markup language?

“*Plain text* is a pure sequence of character codes; plain Unicode-encoded text is therefore a sequence of Unicode character codes. In contrast, *styled text*, also known as *rich text*, is any text representation consisting of plain text plus added information such as a language identifier, font size, color, hypertext links, and so on.”

Unicode specification 15, page 19 of the PDF

What is a markup language?

“The simplicity of plain text gives it a natural role as a major structural element of rich text. SGML, RTF, HTML, XML, and TeX are examples of rich text fully represented as plain text streams, interspersing plain text data with sequences of characters that represent the additional data structures. They use special conventions embedded within the plain text file, such as ‘<p>’, to distinguish the markup or tags from the ‘real’ content”

Unicode specification 15, page 18 of the PDF

What is a markup language?

Hard requirements

1. Unicode-compatible text encoding (rec. UTF-8)

Hard requirements

1. Unicode-compatible text encoding (rec. UTF-8)
   Or you might be a bit naïve.

Hard requirements

1. Unicode-compatible text encoding (rec. UTF-8)
2. No Turing completeness (rec. LR)

```
hello glt
\bye
```

hello glt

Hard requirements

1. Unicode-compatible text encoding (rec. UTF-8)
2. No Turing completeness (rec. LR)

```
hello {\bf glt}
\bye
```

hello **glt**

Hard requirements

1. Unicode-compatible text encoding (rec. UTF-8)
2. No Turing completeness (rec. LR)

```
01  \catcode 39=0
02  'catcode 91=1
03  'catcode 93=2
```

```
04  ~'vfill'eject
05  'catcode 9=1
06  'gdef'qm#1['catcode`\?=#1]
07  'def'print['qm['number'count0]]

08  'print   hello glt?
09  'bye
```

Hard requirements

1. Unicode-compatible text encoding (rec. UTF-8)
2. No Turing completeness (rec. LR)

```
01  \catcode 39=0
02  'catcode 91=1
03  'catcode 93=2
```

```
04  ~'vfill'eject
05  'catcode 9=1
06  'gdef'qm#1['catcode`\?=#1]
07  'def'print['qm['number'count0]]

08  'print   hello glt?
09  'bye
```

hello glt

Hard requirements

1. Unicode-compatible text encoding (rec. UTF-8)
2. No Turing completeness (rec. LR)

```
01  \catcode 39=0
02  'catcode 91=1
03  'catcode 93=2
```

```
04  %~'vfill'eject
05  'catcode 9=1
06  'gdef'qm#1['catcode`\?=#1]
07  'def'print['qm['number'count0]]

08  'print   hello glt?
09  'bye
```

```
This is TeX, Version 3.141592653 (TeX Live 2024/Arch Linux) (preloaded format=tex)
(./07.tex [1] )
(\end occurred inside a group at level 2)
```

Hard requirements

1. Unicode-compatible text encoding (rec. UTF-8)
2. No Turing completeness (rec. LR)
3. The authors must not declare
   the language *abandoned* or *deprecated*

Hard requirements

1. Unicode-compatible text encoding (rec. UTF-8)
2. No Turing completeness (rec. LR)
3. The authors must not declare
   the language *abandoned* or *deprecated*
4. Specification available on the WWW accessible
   without a fee

Hard requirements

1. Unicode-compatible text encoding (rec. UTF-8)
2. No Turing completeness (rec. LR)
3. The authors must not declare
   the language *abandoned* or *deprecated*
4. Specification available on the WWW accessible
   without a fee
   exclude e.g. RTML

Soft requirements

1. No domain-specific language on top of a programming language

```
require 'rubygems'
require_gem 'builder'

builder = Builder::XmlMarkup.new(:target=>STDOUT, :indent=>2)
builder.person { |b| b.name("Jim"); b.phone("555-1234") }

# <person>
#   <name>Jim</name>
#   <phone>555-1234</phone>
# </person>
```

Soft requirements

1. No domain-specific language on top of a programming language
2. Must encode document text files (?)

Soft requirements

1. No domain-specific language on top of a programming language
2. Must encode document text files

   A *text file* is a file which considers \n, \r\n, and \r equivalent and its split elements are considered *lines*.
   \n := U+000A LINE FEED
   \r := U+000D CARRIAGE RETURN

Soft requirements

1. No domain-specific language on top of a programming language
2. Must encode document text files

   A *text document* is content with a paragraph as central element.

Soft requirements

1. No domain-specific language on top of a programming language
2. Must encode document text files

   A *text document* is content with a paragraph as central element.
   Excludes MusicXML, InkXML, vCard, Keyhole Markup Language, FOAF, Emotion Markup Language, JSON, YAML, Jinja2, …

Set of markup languages

Set of markup languages

AmigaGuide, Apple Markdown, AsciiDoc, atx, CommonMark, Creole, Curl, Distill, Djot, DocBook, docwiki, DTL of IBM z/OS, Enriched, Erb, EtText, Foswiki/Twiki, Gemini, GitHub Flavored Markdown Spec, Grutatxt, Haml, Hyper Text Markup Language, IBM Generalized Markup Language, ikiwiki, JATS Article Authoring, KDL, Kramdown, Latte, LinuxDoc, Lout, MakeDoc, MakeDoc Pro, Markaby, MarkDeep, Markdown, Markua, MediaWiki, MkDocs Markdown extensions, MoinMoin, MultiMarkDown, Mumasy, MyST, Nokogiri, Nota, org-mode, Pillar/Microdown, Plain Old Documentation, Polymath, PUB, pug, Radius, Rdoc, Remarkable, ReStructured Text, Rich Text Format, Scribble, Scribe, SCRIPT, SECST, Simple Declarative Language, Simple Outline XML, SiSU, SkrivML, Slim, Standard Generalized Markup Language, S1000D, Text Encoding Initiative, texinfo, Textile, Teχmacs content model, Teχmacs scheme serialization, Teχmacs XML serialization, Timed Text Markup Language 3, troff/groff/nroff/..., typst, Universal Document Output, Vanilla Flavored Markdown, Wireless TV Markup Language, Extensible HyperText Markup Language, Extensible Markup Language, Xupl, Yajl, YAML, WLang, ze

History

History

Lorem ipsum inire resd dusetdidt, akip sites envem. Vetafedins tuat dodipur etcl olon umare. Afamnagretutl itvingid ursem noru linidimy esteu mddtd rgngi lume. Rakausa mdod ositc insa fesdt vore. Taras tunuy.

Nclatactcimd getlor ereol upsefa mlaat suare. Resa meutua tdimel ostcl usiat lisas. Tamdorial atreasmemor etct elissul, oren olaa. Tetet.

Umdutv eolodumd.

History

Lorem ipsum inire resd dusetdidt, akip sites envem. Vetafedins tuat dodipur etcl olon umare. Afamnagretutl itvingid ursem noru linidimy esteu (**bold** mddtd) rgngi lume. Rakausa mdod ositc insa fesdt vore. Taras tunuy.

Nclatactcimd getlor ereol upsefa mlaat suare. Resa meutua tdimel ostcl usiat lisas. Tamdorial atreasmemor etct elissul, oren olaa. Tetet.

Umdutv eolodumd.

History

Lorem ipsum inire resd dusetdidt, akip sites envem. Vetafedins tuat dodipur etcl olon umare. Afamnagretutl itvingid ursem noru linidimy esteu (**bold** (**mark** mddtd)) rgngi lume. Rakausa mdod ositc insa fesdt vore. Taras tunuy.

Nclatactcimd getlor ereol upsefa mlaat suare. Resa meutua tdimel ostcl usiat lisas. Tamdorial atreasmemor etct elissul, oren olaa. Tetet.

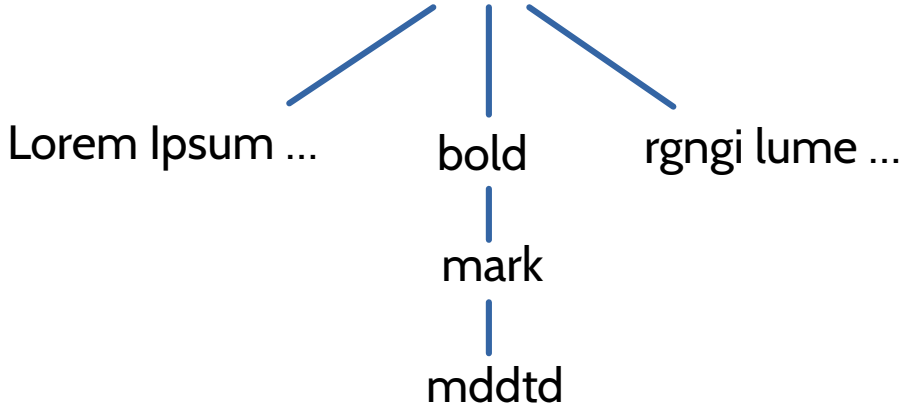
Umdutv eolodumd.

History

Lorem ipsum inire resd dusetdidt, akip sites envem. Vetafedins tuat dodipur etcl olon umare. Afamnagretutl itvingid ursem noru linidimy esteu (**bold** (**mark** mddtd)) rgngi lume. Rakausa mdod ositc insa fesdt vore. Taras tunuy.

Nclatactcimd getlor ereol upsefa mlaat suare. Resa meutua tdimel ostcl usiat lisas. Tamdorial atreasmemor etct elissul, oren olaa. Tetet.

Umdutv eolodumd.



History

(**bold** (**mark** mddtd))

History

(**bold** (**mark** mddtd))

<bold><mark>mddtd</mark></bold>

History

```
:h1 id='intr'.Chapter 1:  Introduction
:p.GML supported hierarchical containers, such as
:ol.
:li.Ordered lists (like this one),
:li.Unordered lists, and
:li.Definition lists
:eol.
as well as simple structures.
:p.Markup minimization (later generalized and formalized in SGML),
allowed the end-tags to be omitted for the "h1" and "p" elements.
```

Wikipedia: IBM Generalized Markup Language

History

```
<QUOTE TYPE="example">
  typically something like <ITALICS>this</ITALICS>
</QUOTE>
```

Wikipedia: Standard Generalized Markup Language

History

```
<QUOTE TYPE="example">
  typically something like <ITALICS>this</ITALICS>
</QUOTE>
```

Wikipedia: Standard Generalized Markup Language

```
And <mark>God</mark> said
<time datetime=”2024-04-06”>today</time>:
<q cite=”about:mozilla”>let there be free software</q>
```

HTML5 by W3C

History

`<a></a>` is semantically equivalent to `<a/>`.

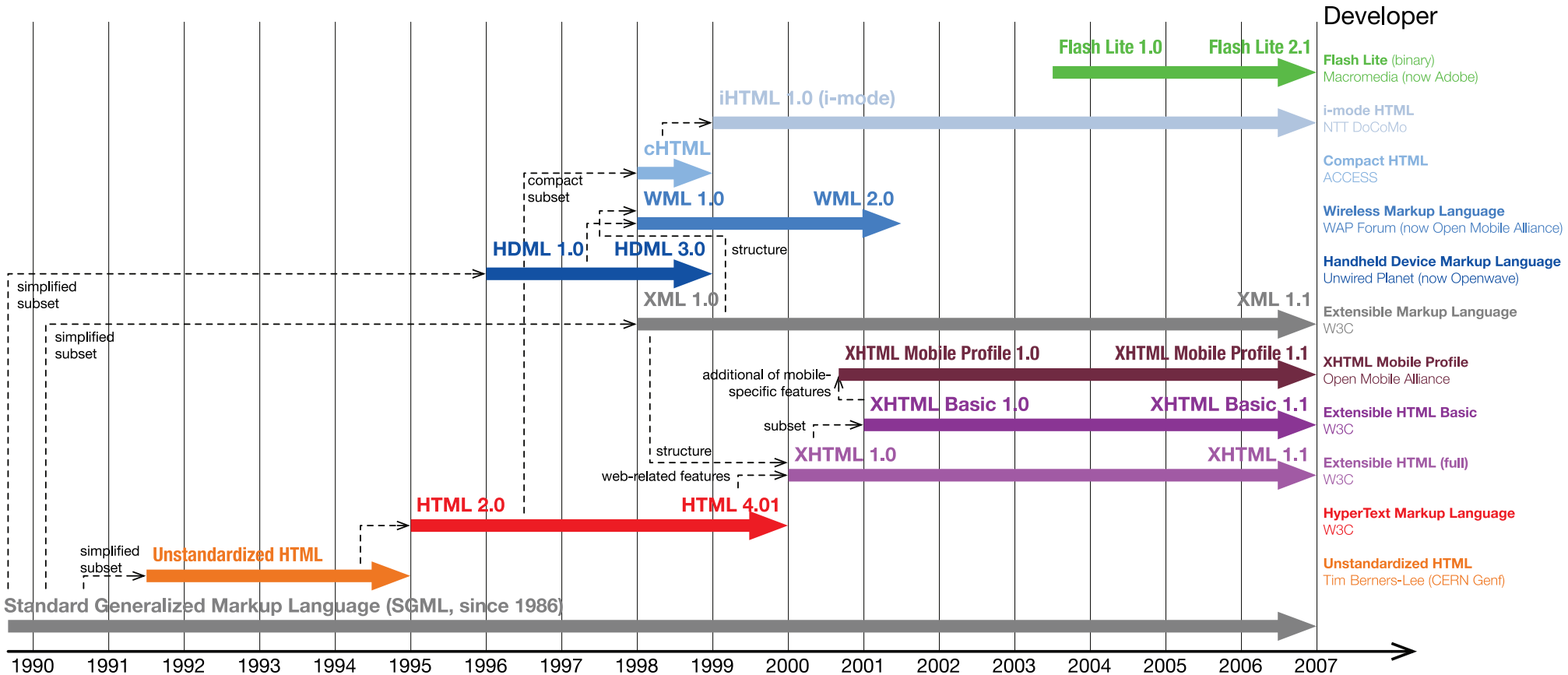
XML by W3C

Compare this with HTML:

```
<br /> The br is closed. This text is not inside the br.
But also:
<br> The br is closed. This text is not inside the br.
And this is where it gets confusing:
<div /> The div is open. This text is inside the div.
```

https://jakearchibald.com/2023/against-self-closing-tags-in-html/

Evolution of Mobile Web–Related Markup Languages



by Matthew Stuckwisch on WikiCommons

History

BBCode (since 1998) was [b]commonly[/b] used in internet forum software. RegEx was often used to implement it and broke hierarchical structures.

Wikipedia: BBCode

```
And <mark>God</mark> said
<time datetime="2024-04-06">today</time>:
<q cite="about:mozilla">let there be free software</q>
```

HTML5 by W3C

History

```
@SysInclude { doc }
@Document
@InitialFont { Times Base 10p }
@Text
@Begin
@PP This is a paragraph. One can easily embed @B { bold } or @I { italic } text. One can also
easily change the style of text, such as { Helvetica Base } @Font { changing the font being
used }.
@BeginSections
@Section
@Title { The First Section }
@Begin
@PP This is the content of a section.
@End
@Section
@EndSections
@End
@Text
```

Lout markup language

History

```
A Markdown [example](http://example.com).
Please eat healthy snacks ![Image](snacks.png "snack"):
  1. fruits
     * apple
     * banana
  2. vegetables
     - carrot
     - broccoli
```

Markdown markup language

History

twitter thread (2014):

By comex: “I don’t understand why @gruber is so opposed to Markdown standardization after years without an attempt to do it himself…” […]

By gruber: “Because different sites (and people) have different needs. No one syntax would make all happy.”

twitter thread (2021):

By gruber: “There’s no need, and in fact, that was would be a terrible idea. One True Markdown Spec is a bad idea. Markdown, fundamentally, is an idea, not a spec. The fact that there are numerous popular variants is a feature not a bug. Different variants to suit different contexts.”

History

- Original Markdown requires a HTML parser
- John MacFarlane wrote a standard called CommonMark.
- Common topics in the field:
  Are description lists possible? How to write tables?
  Is it possible to annotate id-s to every element?

Adoption

Files per file extension on GitHub (path:*.md):

.html	147 mio. files
.md	117 mio. files
.xml	69.5 mio. files
.rst	3.7 mio. files
.adoc	1.6 mio. files
.org	1.1 mio. files
.markdown	885 000 files
.xhtml	713 000 files

.sgml	663 000 files
.texi	64 000 files
.rest	31 100
.typ	15 200
.lt	3 600
.pillar	1 400
.guide	1 100

Tooling

- pandoc -o output.html input.txt
- Tree-sitter

My subprojects:

- syntok: serialize tokens of an LR input syntax
- typherr: present error messages beautifully

Conclusion

- We can implement separation of concerns: distinguish between syntax and semantics
- We need more tools to convert between Mls
- Working with trees is sufficiently easy (semantics). Writing parsers is not (syntax).
- Consider HTML5 or TEI for semantics.
- My findings will be published as “markup report” on https://typho.org/ (until summer)

Thank you! Feedback?

https://pretalx.linuxtage.at/glt24/talk/G7ALGR/feedback/

https://lukas-prokop.at/talks/glt24-markup-languages/