Educational Compiler Project — COMPO

Session 1 — Grammar

Tanguy Berthoud Yohann Martin

October 16, 2022

Contents

Session goals You will discover the Markdown language and Compo specifications, and produce a first version of your grammar.

1 Introduction to Markdown

Markdown is a lightweight markup language for creating formatted text using a plain-text editor. It has been created in 2004 by John Gruber and Aaron Swartz.

Markdown input

```
# Hello, world!

## Subtitle

**Lorem ipsum** dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in *voluptate* velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.
```

Rendered document

Hello, world!

Subtitle

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in *voluptate* velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

In our project, we will be using some features added by GitHub-flavoured Markdown.

2 Compilation specifications

2.1 Markdown to HTML

Markdown	HTML
Paragraph	`<p>` tag
Bold text (`**...**`)	`<b>` tag
Italic text (`*...*`)	`<i>` tag
Underlined text (`__...__`)	`<u>` tag
Strikethrough text (`~~...~~`)	`<s>` tag
Titles (`#` to `######`)	`<h1>` to `<h6>` tags
Links (`[...](...)`)	`<a>` tag
Images (`![...](...)`)	`<img>` tag
Quotes (`> ...`)	`<blockquote>` tag
Inline code (`` `...` ``)	`<code>` tag
Block code (` ```...``` `)	`<pre>` tag
SVG extension (` ```xsvg ...``` `)	subsection 2.2
Horizontal rule (`---`)	`<hr>` tag

2.1.1 Paragraph

Paragraphs in Markdown are several continuous lines with text, separated by blank lines.

Markdown input

```
Lorem ipsum dolor sit amet,
consectetur adipiscing elit.
Pellentesque commodo accumsan
sollicitudin.
Aliquam non leo dapibus,
aliquet lectus non, interdum
sem.

Curabitur ullamcorper ligula
purus, et volutpat risus
suscipit ac.
Curabitur vel posuere metus.
```

HTML output

```
<p>
Lorem ipsum dolor sit amet,
   consectetur adipiscing elit.
Pellentesque commodo accumsan
   sollicitudin.
Aliquam non leo dapibus,
   aliquet lectus non,
   interdum sem.
</p>
<p>
Curabitur ullamcorper ligula
   purus, et volutpat risus
   suscipit ac.
Curabitur vel posuere metus.
</p>
```

Rendered in browser

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque commodo accumsan sollicitudin. Aliquam non leo dapibus, aliquet lectus non, interdum sem.

Curabitur ullamcorper ligula purus, et volutpat risus suscipit ac. Curabitur vel posuere metus.

2.1.2 Italic text

Italic text in Markdown are delimited with a simple asterisk.

Markdown input

```
This is *italic*!
```

HTML output

```
This is <i>italic</i>!
```

Rendered in browser

This is *italic*!

2.1.3 Bold text

Bold text in Markdown are delimited with a double asterisk.

Markdown input	HTML output	Rendered in browser
`This is **bold**!`	`This is <b>bold</b>!`	This is **bold**!

2.1.4 Underlined text

Underlined text in Markdown are delimited with a double underscore.

Markdown input	HTML output	Rendered in browser
`This is __underlined__!`	`This is <u>underlined</u>!`	This is underlined!

2.1.5 Strikethrough text

Strikethrough text in Markdown are delimited with a double tilde.

Markdown input	HTML output	Rendered in browser
`This is ~~struck~~!`	`This is <s>struck</s>!`	This is ~~struck~~!

2.1.6 Titles

Titles can go from 1 to 6, in descending order. They are marked with a number of # characters at the beginning of the line.

Markdown input

```
# First title

## Sub title

### Sub sub title
```

HTML output

```
<h1>First title</h1>
<h2>Sub title</h2>
<h3>Sub sub title</h3>
```

Rendered in browser

First title

Sub title

Sub sub title

2.1.7 Links

Links are clickable texts which open a new page. They are marked by a text between square brackets followed by the link between parenthesis.

Markdown input

```
Click on [this link](https://
youtu.be/dQw4w9WgXcQ)
```

HTML output

```
Click on <a href="https://
   youtu.be/dQw4w9WgXcQ">
  this link
</a>
```

Rendered in browser

Click on this link

2.1.8 Images

Markdown input

```
![White rhinos](https://free-
images.com/or/7bab/africa
_white_rhinos_rhino.jpg)
```

HTML output

```
<img src="https://free-images.
   com/or/7bab/
   africa_white_rhinos_rhino.
   jpg" alt="White rhinos"/>
```

Rendered in browser

2.1.9 Quotes

Markdown input

```
> This is a famous quote.
> This quote has two lines.
```

HTML output

```
<blockquote>
This is a famous quote.
This quote has two lines.
</blockquote>
```

Rendered in browser

> This is a famous quote.
> This quote has two lines.

2.1.10 Inline code

Markdown input

```
There is some `inline code`!
```

HTML output

```
There is some <code>inline
   code</code>!
```

Rendered in browser

There is some `inline code`!

2.1.11 Block code

Markdown input

```
```
This is a block of code
on multiple lines!
```
```

HTML output

```
<pre><code>
This is a block of code
on multiple lines!
</code></pre>
```

Rendered in browser

```
This is a block of code
on multiple lines!
```

2.1.12 SVG extension

Our SVG extension is a special block code (see subsubsection 2.1.11) annotated with some information: the minimum x-position, the minimum y-position, the width and the height of the viewport. See the `viewBox` attribute for more details. Inside the code block is expected a valid code from our SVG extension.

Markdown input

````
```xsvg:minx,miny,w,h
...
```
````

HTML output

```
<svg viewBox="minx miny w h">
    ...
</svg>
```

2.1.13 Horizontal rule

Markdown input

```
A block of text

---

A second block of text

---------------

A third block of text
```

HTML output

```
<p>A block of text</p>
<hr/>
<p>A second block of text</p>
<hr/>
<p>A third block of text</p>
```

Rendered in browser

A block of text

A second block of text

A third block of text

2.2 SVG extension

SVG extension	SVG
`line x1,y1 x2,y2 (color)`	`<line>` tag
`polyline x1,y1 ...(color)`	`<polyline>` tag
`polygon x1,y1 ...(color-fill) (color-stroke)`	`<polygon>` tag
`circle x,y r (color-fill) (color-stroke)`	`<circle>` tag
`ellipse x,y rx ry (color-fill) (color-stroke)`	`<ellipse>` tag
`rect x,y w h (color-fill) (color-stroke)`	`<rect>` tag
`text x,y "text" (anchor) (color-fill) (color-stroke)`	`<text>` tag

2.2.1 SVG line

```
line x1,y1 x2,y2 color
```

- `x1,y1 x2,y2` are the coordinates of the line
- `color` is the CSS color of the line (optional)

Markdown input

````
```xsvg:0,0,20,10
line 0,0 10,0 red
```
````

HTML output

```
<svg viewBox="0 0 20 10">
    <line x1="0" y1="0"
   x2="10" y2="0" stroke="red"
   />
</svg>
```

Rendered in browser

2.2.2 SVG polyline

```
polyline x1,y1 ... color
```

- `x1,y1 ...` are the coordinates of each line
- `color` is the CSS color of the lines (optional)

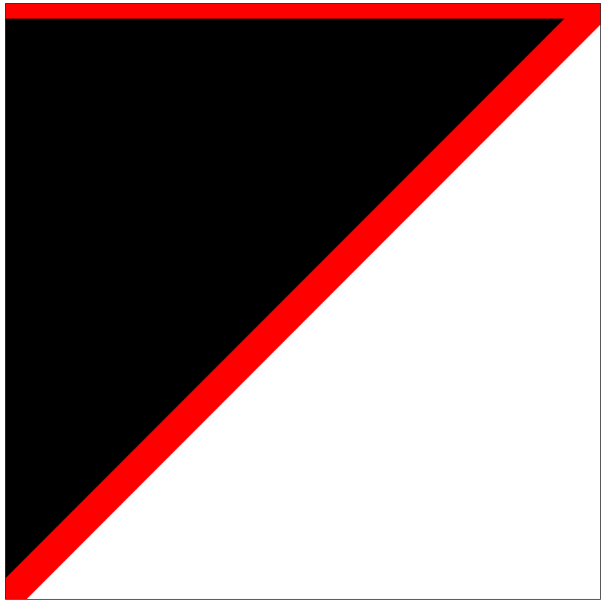
Markdown input

````
```xsvg:0,0,20,20
polyline 0,0 5,5 10,10 15,15
   red
```
````

HTML output

```
<svg viewBox="0 0 20 20">
    <polyline points="0,0
   20,0 0,20" stroke="red" />
</svg>
```

Rendered in browser



2.2.3 SVG polygon

```
polygon x1,y1 ... color-fill color-stroke
```

- `x1,y1 ...` are the coordinates of each line
- `color-fill` is the CSS color inside the shape (optional)
- `color-stroke` is the CSS color of the lines (optional)

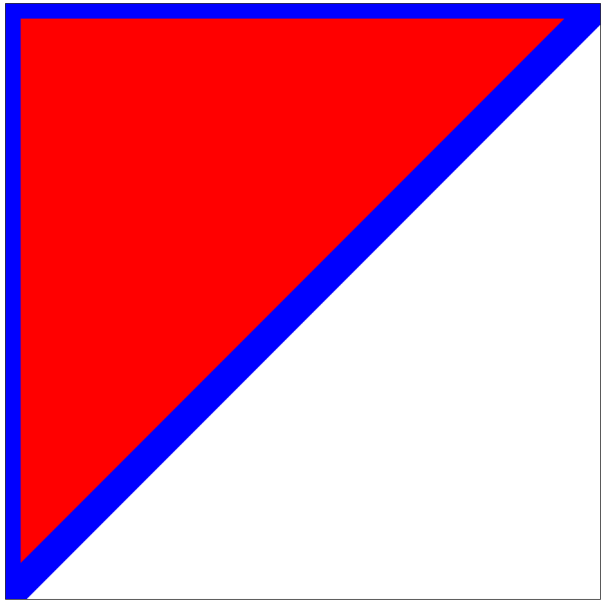
Markdown input

````
```svg:0,0,20,20
polygon 0,0 20,0 0,20 blue red
```
````

HTML output

```
<svg viewBox="0 0 20 20">
    <polyline points="0,0
  20,0 0,20" fill="red"
  stroke="blue" />
</svg>
```

Rendered in browser



2.2.4 SVG circle

```
circle x,y r color-fill color-stroke
```

- `x,y` are the coordinates of the center of the circle
- `r` is the radius of the circle
- `color-fill` is the CSS color inside the shape (optional)
- `color-stroke` is the CSS color of the outer circle (optional)

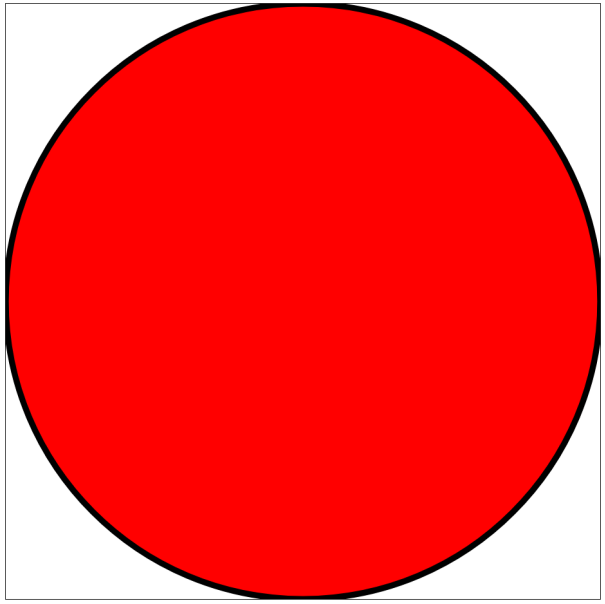
Markdown input

````
```xsvg:0,0,100,100
circle 50,50 50 red black
```
````

HTML output

```
<svg viewBox="0 0 100 100">
    <circle cx="50" cy="50"
   r="50" fill="red"
   stroke="black"/>
</svg>
```

Rendered in browser



2.2.5 SVG ellipse

```
ellipse x,y rx ry color-fill color-stroke
```

- `x,y` are the coordinates of the center of the ellipse
- `rx` is the x-axis radius of the ellipse
- `ry` is the y-axis radius of the ellipse
- `color-fill` is the CSS color inside the shape (optional)
- `color-stroke` is the CSS color of the outer ellipse (optional)

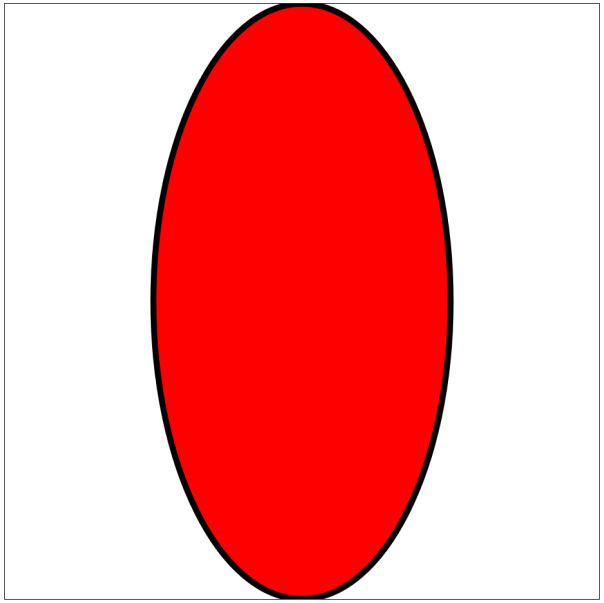
Markdown input

````
```xsvg:0,0,100,100
ellipse 50,50 25 50 red black
```
````

HTML output

```
<svg viewBox="0 0 100 100">
    <ellipse cx="50" cy="50"
   rx="25" ry="50" fill="red"
   stroke="black"/>
</svg>
```

Rendered in browser



2.2.6 SVG rectangle

```
rect x,y w h color-fill color-stroke
```

- `x,y` are the coordinates of the top left corner of the rectangle
- `w` is the width of the rectangle
- `h` is the height of the rectangle
- `color-fill` is the CSS color inside the shape (optional)
- `color-stroke` is the CSS color of the outer rectangle (optional)

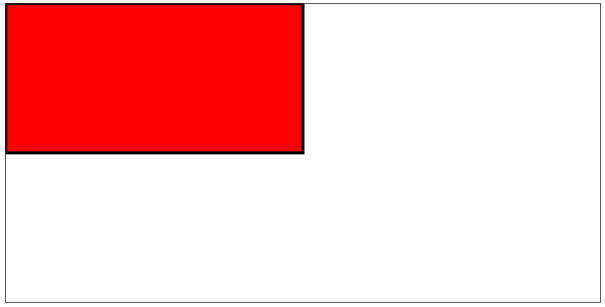
Markdown input

````
```xsvg:0,0,200,100
rect 0,0 100 50 red black
```
````

HTML output

```
<svg viewBox="0 0 200 100">
    <rect x="0" y="0"
   width="100" height="50"
   fill="red" stroke="black"/>
</svg>
```

Rendered in browser



2.2.7 SVG text

```
text x,y "text" anchor color-fill color-stroke
```

- `x,y` are the coordinates of the text
- `text` is the text to display
- `anchor` is the `text-anchor` attribute, which can take the following values: `start`, `middle`, `end`
- `color-fill` is the CSS color of the text (optional)
- `color-stroke` is the CSS color of the text border (optional)

Markdown input

````
```xsvg:0,0,200,100
text 0,50 "My text" middle red
```
````

HTML output

```
<svg viewBox="0 0 200 100">
    <text x="0" y="50"
   fill="red">My text</text>
</svg>
```

Rendered in browser

3 Deliverables

You are to develop a grammar that will recognize Markdown documents as described in section 2, taking into consideration that this grammar will be used later on to produce equivalent HTML code. Write a report containing the complete grammar, with additional comments if you think that more explanations are required.

COMPO

Report 1: Grammar

Student 1 Student 2 Student 3

Date

1 Grammar

Here input your complete grammar.

2 Comments

Here write some additional info about your choices, and the difficulties you have encountered.

1