What would Doxygen look like if it were written in Miking?

Martin Zivojinovic

What is Doxygen ?

A documentation generator !

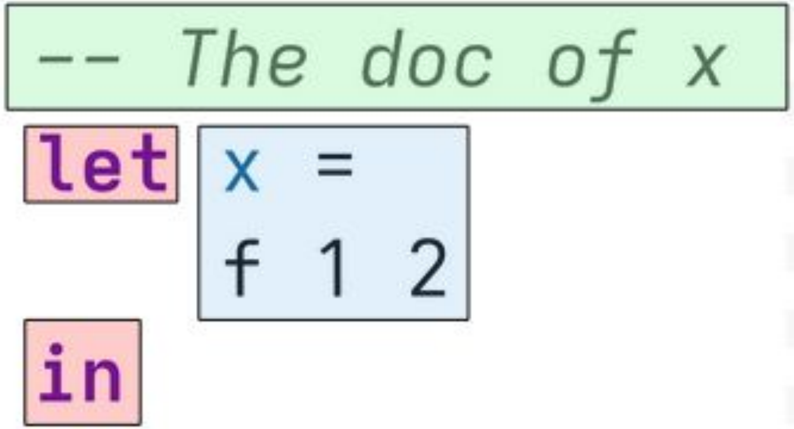
But what is a documentation generator ?

“A documentation generator is a tool that automatically creates documentation from a project’s source files.

Instead of writing all documentation manually, developers include structured comments, annotations, or metadata in the code. The generator then processes these inputs and produces readable documents.”

Writing a Parser

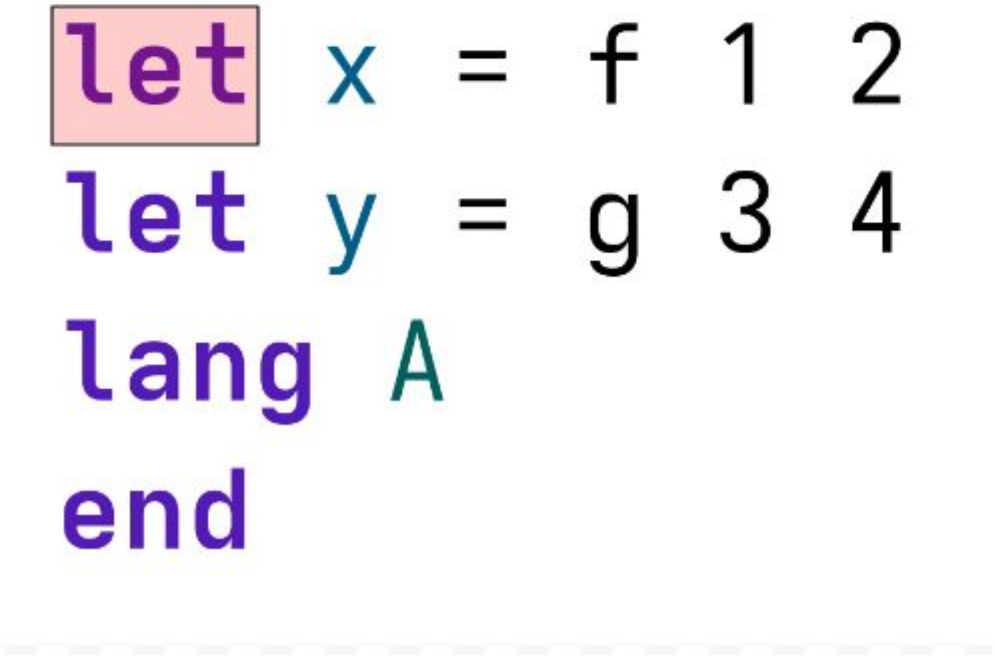
```
-- The doc of x
let x =
    f 1 2
in
```



```
-- The doc of x
let x =
    f 1 2
in
```

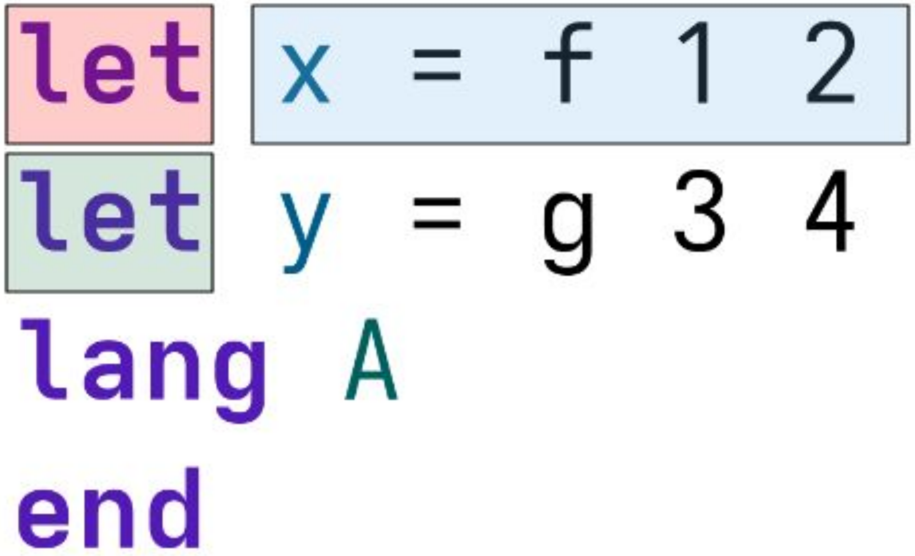
Opener	Closer
lang	end
let	in
syn	syn, sem, end
type	in

```
let x = f 1 2
let y = g 3 4
lang A
end
```

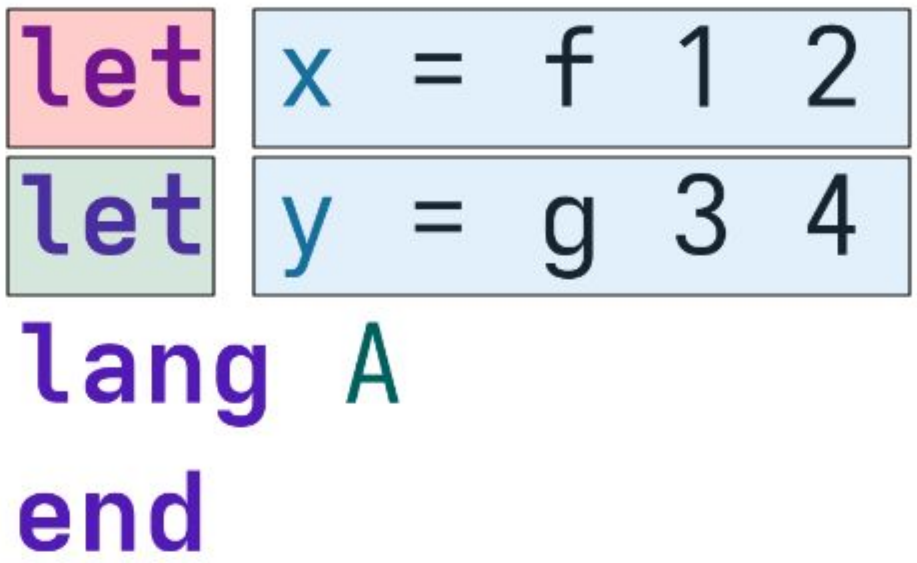


```
let x = f 1 2
let y = g 3 4
lang A
end
```

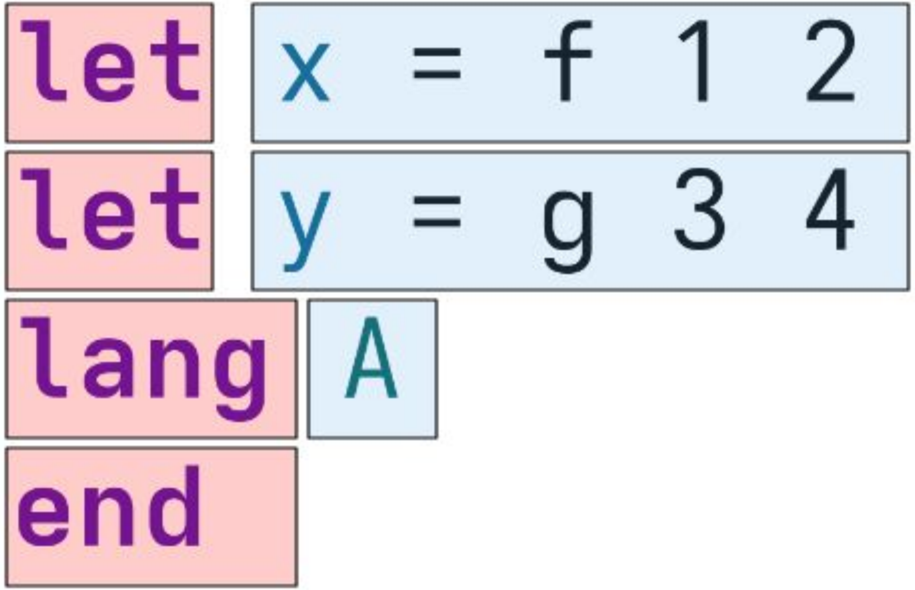
```
let x = f 1 2
let y = g 3 4
lang A
end
```



```
let x = f 1 2
let y = g 3 4
lang A
end
```



```
let x = f 1 2
let y = g 3 4
lang A
end
```



```
let x = f 1 2
let y = g 3 4
lang A
end
```

The full docgen pipeline

- Scanning - scan the structure of the project
- Parsing - parse the input code
- Extracting - generate IR from parsing output
- Labeling - fetch the types using the Miking parser
- Naming - associate names with url
- Rendering - generate the output files

Labeling

- Resolve the type of each item
- Map the Docgen custom AST to the actual Miking AST
- Treat the Miking AST as a stream that provides item types sequentially

Naming

- Associate names with their corresponding URLs
- Enable clickable type references from anywhere in the code
- Correctly handle scoping rules and shadowing

Rendering

- Generate all documentation pages
- Support multiple backends: HTML, Markdown, MDX, and soon LaTeX
- Leverage Miking’s assembling feature to eliminate code duplication smoothly

Thank you for your attention.

Questions?