

Wikipedia

to

L<sub>A</sub>T<sub>E</sub>X

by

Dirk Hünninger

Physicist

# What do we need?

- Parsec
- Some very generic brackets
- Parse-Tree
- Recursion on it to make LaTeX

# Programming Languages?

## Python 3

- GUI
- I/O
- “Glue”

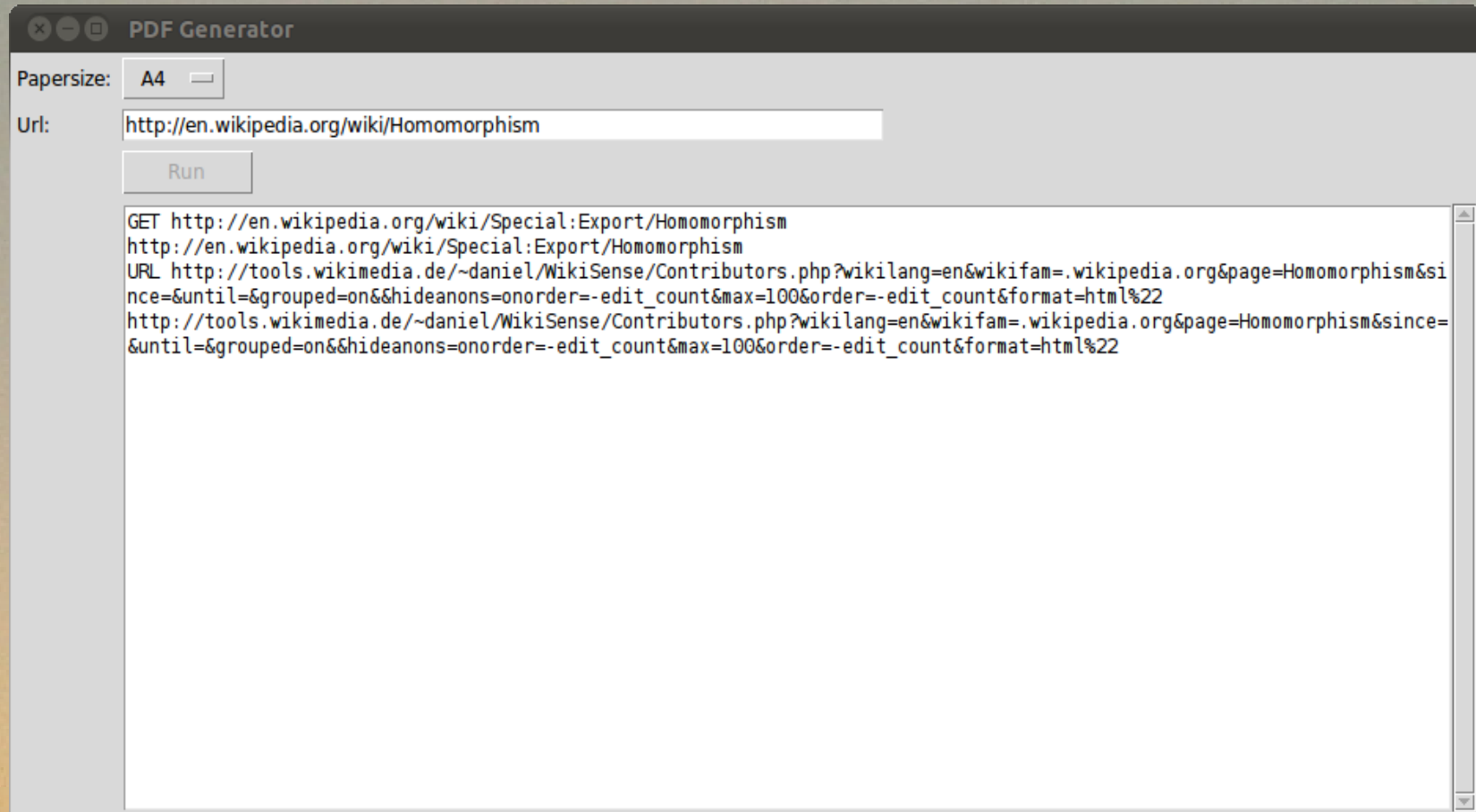
## Haskell

- Compiler

# Other Software included?

- RSVG
- Image Magick
- LaTeX distribution

# The GUI





# Parse-Tree

```
'\n',Environment Wikiheading (Str "==")
[C 'S',C 'e',C 'e',C ' ',C 'a',C 'l',C
's',C 'o'],Environment Itemgroup (Str
"*") [Environment ItemEnv (Str "*") [Item
'*',Environment Wikilink (Str "") [C
'c',C 'u',C 'n',C 'c',C 't',C 'i',C 'o',C
'n'],Item '*',Environment Wikilink (Str
"") [C 'd',C 'i',C 'f',C 'f',C 'e',C
'o',C 'm',C 'o',C 'r',C 'p',C 'h',C 'i',C
's',C 'm'],Item '*',Environment Wikilink
(Str "") [C 'C
```

# Generic Brackets

```
wikilinkp = baseParser {  
    start = string "[[">> return (Str ""),  
    end = \x-> string "]">> return (),  
    allowed = everywhere++wikilinkwhere,  
    self = Wikilink  
}
```



# More on Generic Brackets

```
data MyParser tok= MyParser {
  bad :: GenParser tok () (),
  start :: GenParser tok () StartData,
  end :: StartData->GenParser tok () (),
  allowed :: [EnvType],
  self :: EnvType,
  modify :: StartData->[Anything tok]->
[Anything tok]
}
```

Questions?