

# Shapes

— ett hyfsat funktionellt ritspråk

Henrik Tidefelt

LiTH

5 september 2007

# Mål

Med den här UppLYSningen hoppas jag

# Mål

Med den här UppLYSningen hoppas jag

- Att ni ska få ett hum om vad Shapes är.

# Mål

Med den här UppLYSningen hoppas jag

- Att ni ska få ett hum om vad Shapes är.
- Få höra era invändningar mot designen som den ser ut idag.

# Mål

Med den här UppLYSningen hoppas jag

- Att ni ska få ett hum om vad Shapes är.
- Få höra era invändningar mot designen som den ser ut idag.
- Lyckas hitta någon testpilot.

# Mål

Med den här UppLYSningen hoppas jag

- Att ni ska få ett hum om vad Shapes är.
- Få höra era invändningar mot designen som den ser ut idag.
- Lyckas hitta någon testpilot.
- Väcka intresse för utvecklingssamarbete.

# Plan

De stora inslagen idag är:

# Plan

De stora inslagen idag är:

- Beskriva hur språkets struktur ser ut idag.



# Plan

De stora inslagen idag är:

- Beskriva hur språkets struktur ser ut idag.
- Visa lite av de funktioner som kärnan erbjuder.

# Plan

De stora inslagen idag är:

- Beskriva hur språkets struktur ser ut idag.
- Visa lite av de funktioner som kärnan erbjuder.
- Diskutera intressanta utmaningar för framtiden.

# Introduktion

# Rötter

Shapes har sina rötter i många av de språk jag varit i kontakt med:

- MetaPost (en omarbetning av Knuths MetaFont) — Shapes kom till när jag ledsnade på MetaPost.

# Rötter

Shapes har sina rötter i många av de språk jag varit i kontakt med:

- MetaPost (en omarbetning av Knuths MetaFont) — Shapes kom till när jag ledsnade på MetaPost.
- Scheme — syntax och funktions-begreppet.

# Rötter

Shapes har sina rötter i många av de språk jag varit i kontakt med:

- MetaPost (en omarbetning av Knuths MetaFont) — Shapes kom till när jag ledsnade på MetaPost.
- Scheme — syntax och funktions-begreppet.
- Haskell — för sina rena ideal.

# Rötter

Shapes har sina rötter i många av de språk jag varit i kontakt med:

- MetaPost (en omarbetning av Knuths MetaFont) — Shapes kom till när jag ledsnade på MetaPost.
- Scheme — syntax och funktions-begreppet.
- Haskell — för sina rena ideal.
- C++ — utmatningssyntaxen.

# Alternativ

Några andra ritspråk som finns och/eller används idag:

- MetaPost



# Alternativ

Några andra ritspråk som finns och/eller används idag:

- MetaPost
- Asymptote

# Alternativ

Några andra ritspråk som finns och/eller används idag:

- MetaPost
- Asymptote
- PGF och TikZ

# Alternativ

Några andra ritspråk som finns och/eller används idag:

- MetaPost
- Asymptote
- PGF och TikZ
- Haskell PDF

# Varför Shapes?

Givet utbudet av alternativa rit-språk, varför utveckla ett till? Här är några skäl:

- Inte funktionellt orienterade (alla utom Haskell PDF).

# Varför Shapes?

Givet utbudet av alternativa rit-språk, varför utveckla ett till? Här är några skäl:

- Inte funktionellt orienterade (alla utom Haskell PDF).
- Dålig beräkningskapacitet (MetaPost och PDF/TikZ).

# Varför Shapes?

Givet utbudet av alternativa rit-språk, varför utveckla ett till? Här är några skäl:

- Inte funktionellt orienterade (alla utom Haskell PDF).
- Dålig beräkningskapacitet (MetaPost och PDF/TikZ).
- Saknar domän-specifik syntax (Haskell PDF).

# Varför Shapes?

Givet utbudet av alternativa rit-språk, varför utveckla ett till? Här är några skäl:

- Inte funktionellt orienterade (alla utom Haskell PDF).
- Dålig beräkningskapacitet (MetaPost och PDF/TikZ).
- Saknar domän-specifik syntax (Haskell PDF).
- Inte publicerade när Shapes påbörjades (Asymptote och Haskell PDF).

# Hello, shaper!

- `page << [stroke (0cm,0cm)--(1cm,1cm)]`



# Hello, shaper!

- `page << [stroke (0cm,0cm)--(1cm,1cm)]`
- `page << stroke [] ((0cm,0cm)--(1cm,1cm))`

# Hello, shaper!

```
•page << [stroke (0cm,0cm)--(1cm,1cm)]  
•page << stroke [] ((0cm,0cm)--(1cm,1cm))  
[(\ •dst pth .> { •dst << stroke [] pth } )  
  •page (0cm,0cm)--(1cm,1cm)]
```

# Språkets struktur

# Funktioner i kärnan

# Utmaningar för framtiden

# Sammanfattning

# Sammanfattning

- Shapes är...

Slut.