

# Elphie

Stanislav Böhm  
<http://github.com/spirali/elphie>

Elphie is a presentation framework  
based on *Python* and *Inkscape*

The following code ...

```
import elphie

slides = elphie.Slides()
slide = slides.new_slide("The first slide")
slide.text("Hello world!")
slides.render()
```

will produce slide.pdf

You can see the result at next slide

Hello world!

Let us continue with the demonstration ...

Let us continue with the demonstration ...

Elphie ...

Let us continue with the demonstration ...

Elphie ...

... supports ...

Let us continue with the demonstration ...

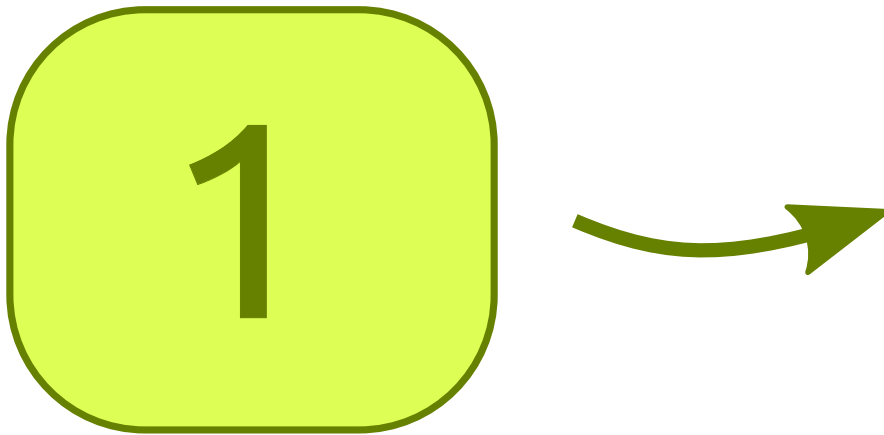
Elphie ...

... supports ...

... fragments.











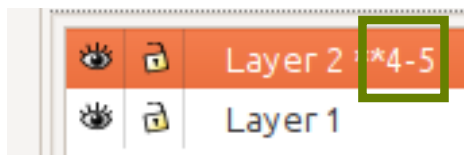
Fragments in svg  
by labelling elements or layers



Fragments in svg  
by labelling elements or layers



Show in the fourth step and later



Show in the fourth step and fifth step

## Example of a list:

- Apple
- Orange
- Banana

```
#include <stdio.h>

/* Hello world program */

int main() {
    printf("Hello world!\n");
    return 0;
}
```

## Columns demonstration

- First item
- Second item
- Third item

```
#include <stdio.h>

int main() {
    printf("Hello world!\n");
    return 0;
}
```



## Title of the frame

This text is in the frame!

- First bullet
- Second bullet

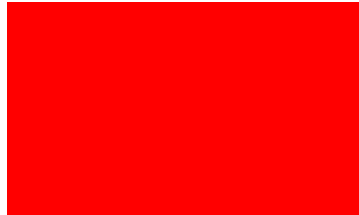
# Header 1

## Header 2

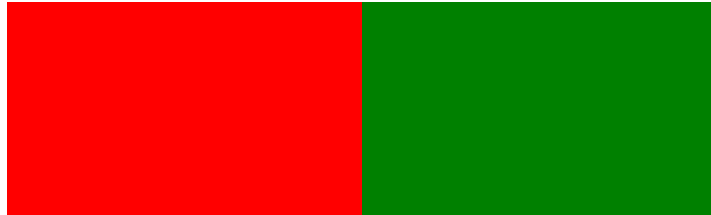
### Header 3

Normal text | Type writer | *emphasis* | **alert**

```
class MyElement(elphie.Element):  
  
    def get_size_request(self, ctx):  
        return elphie.SizeRequest(500, 100)  
  
    def render_body(self, ctx, rect):  
        colors = ("red", "green", "blue")  
        for i, color in enumerate(colors)[:ctx.step]:  
            w = rect.width / 3.0  
            r = elphie.Rect(rect.x + i * w, rect.y,  
                           w, rect.height)  
            ctx.renderer.draw_rect(r, color)
```



```
class MyElement(elphie.Element):  
  
    def get_size_request(self, ctx):  
        return elphie.SizeRequest(500, 100)  
  
    def render_body(self, ctx, rect):  
        colors = ("red", "green", "blue")  
        for i, color in enumerate(colors)[:ctx.step]:  
            w = rect.width / 3.0  
            r = elphie.Rect(rect.x + i * w, rect.y,  
                           w, rect.height)  
            ctx.renderer.draw_rect(r, color)
```



```
class MyElement(elphie.Element):  
  
    def get_size_request(self, ctx):  
        return elphie.SizeRequest(500, 100)  
  
    def render_body(self, ctx, rect):  
        colors = ("red", "green", "blue")  
        for i, color in enumerate(colors)[:ctx.step]:  
            w = rect.width / 3.0  
            r = elphie.Rect(rect.x + i * w, rect.y,  
                            w, rect.height)  
            ctx.renderer.draw_rect(r, color)
```



This slide is rendered by a different theme

Have a nice day!