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

The first slide

Hello world!

Let us continue with the demonstration ...

Let us continue with the demonstration ...

Elphie ...

Let us continue with the demonstration ...

Elphie ...

... supports ...

Text fragments

Let us continue with the demonstration ...

Elphie ...

... supports ...

... fragments.


1

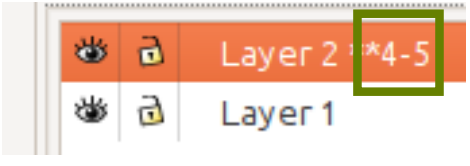

1


1
2

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

Code

```c
#include <stdio.h>

/* Hello world program */

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


```c
#include <stdio.h>

/* Hello world program */

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

Code

```c
#include <stdio.h>

/* Hello world program */

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


```c
#include <stdio.h>

/* Hello world program */

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

Code

```c
#include <stdio.h>

/* Hello world program */

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


```c
#include <stdio.h>

/* Hello world program */

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

Code

```c
#include <stdio.h>

/* Hello world program */

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


```c
#include <stdio.h>

/* Hello world program */

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

Code

```c
#include <stdio.h>

/* Hello world program */

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


```c
#include <stdio.h>

/* Hello world program */

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

Code

```c
#include <stdio.h>

/* Hello world program */

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


```c
#include <stdio.h>

/* Hello world program */

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

```
/path/to/elphie/example$ ls
python3 example.py

/path/to/elphie/example$ python3 example.py
Preprocessing.................. done
Building....................... done
Creating 'example.pdf'........ done
```

```
/path/to/elphie/example$ ls
python3 example.py

/path/to/elphie/example$ python3 example.py
Preprocessing.................. done
Building....................... done
Creating 'example.pdf'........ done
```

Preprocessing is cached

```
/path/to/elphie/example$ ls
python3 example.py

/path/to/elphie/example$ python3 example.py
Preprocessing..................... done
Building.......................... done
Creating 'example.pdf'........ done
```

Building is also cached

```
/path/to/elphie/example$ ls
python3 example.py

/path/to/elphie/example$ python3 example.py
Preprocessing.................. done
Building....................... done
Creating 'example.pdf'........ done
```

This line is a demonstration of *layers*

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

Separators

This is line 1

This is line 2

Under a thick separator

Column 1	Column 2	Column 3
Column 1	Column 2	Column 3
Column 1	Column 2	Column 3
Column 1	Column 2	Column 3

Header 1

Header 2

Header 3

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

Fixed size 15, ignore theme: Normal text | `Type writer` | *emphasis* | **alert**

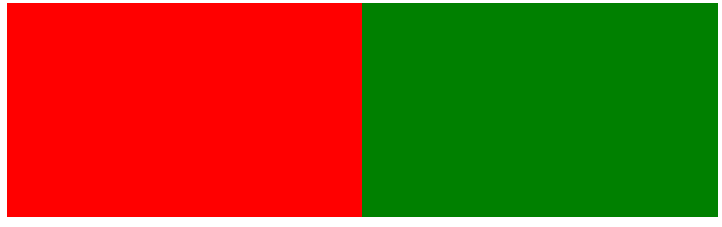
red green blue

Proceduraly generated elements

```
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)
```

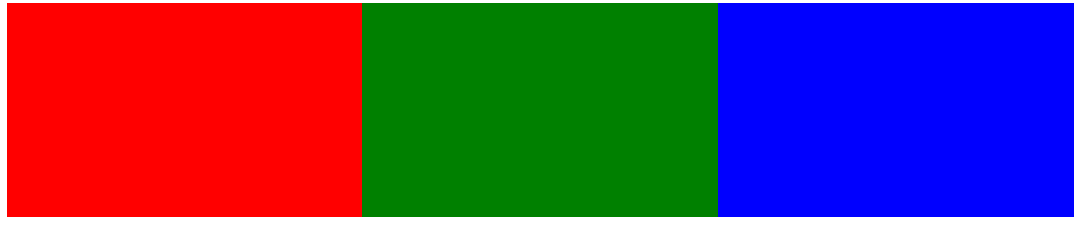
Proceduraly generated elements

```
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)
```



Proceduraly generated elements

```
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!