

Highlighted blocks of code in rst2beamer

The LaTeX source for the corresponding Beamer example can be produced:

```
rst2beamer codeblocks.rst codeblocks.tex
```

If Pygments is available, syntax highlighting can be used:

```
rst2beamer --codeblocks-use-pygments \  
codeblocks.rst codeblocks_hilite.tex
```

Simple codeblocks

The `code-block` (or `sourcecode`) directive can be used to format blocks of source code. Note that the language must be passed as an option. Normally this is represented as a literal block, but if Pygments is activated, the syntax will be highlighted:

```
def myfunc (arg1, arg2='foo'):  
    global baz  
    bar = unicode (quux)  
    return 25
```

Specifying language

Any 'name' recognised as by Pygments can be used as a codeblock language argument:

```
void main()
{
    // declare a variable
    int i;
    for (i= 0; i < 11; i++)
        cout << i << endl;
}
```

No language

A codeblock can be left without a language option, in which case the `codeblocks-default-language` argument is used, or Pygments will guess the language:

```
for ($count = 10; $count >= 1; $count--) {  
    print "$count ";  
}  
print "Blastoff.\n";
```

Tabs and indenting

By ReST translates leading (indenting) tabs as 8 spaces. The argument *codeblocks-replace-tabs* can be used to adjust the indent width. Set it to different values and see how the code below changes:

```
class MyClass (object):  
    def __init__ (self):  
        for i in ['foo', 'bar', 'baz']:  
            setattr (self, i, None)
```