Examples of Code Listings in LaTeX

Andrew Pinkham

June 23, 2013

Abstract

This is a short file to test the various features and shortcuts built in our preamble file. The purpose is to provide TeX files with their output in a PDF. Users may then easily read, learn, or simply copy from the TeX files to jumstart their projects.

Note that all code is licensed under the Simplified BSD License, a copy of which is included with the project code, found on github.

Contents

1 Code Listing

1.1 **verbatim** Environment

Here is an example that uses LaTeX's `verbatim` environment.

```
from django.db import models

class Poll(models.Model):
    question = models.CharField(max_length=200)
    pub_date = models.DateTimeField('date published')

class Choice(models.Model):
    poll = models.ForeignKey(Poll)
```

```
    choice_text = models.CharField(max_length=200)
    votes = models.IntegerField(default=0)
```

Note that the quotes used are standard programming quotes, as opposed to LaTeX's backtick-quote combination. This is possible because of the `upquote` package, included in the preamble of the document.

LaTeX further supplies the `\verb` command, allowing for inline `verbatim` formatting.

Finally, note that the `\verb` command can only be used in captions if the entire command is preceded by the `\cprotect` command, included in the `cprotect` package, which you can read more about [on the documentation provided by CTAN](#). On a related note, text in chapter titles (or section, subsection, etc) are formatted with `\texttt` in this document, as the use of `\verb` in these areas appears to conflict with the `hyperref` package.

1.2 `listings` Package

The following uses the `listings` package to create an environment.

Listing 1: `models.py` from Django Tutorial
```
1  from django.db import models
2
3  class Poll(models.Model):
4      question = models.CharField(max_length=200)
5      pub_date = models.DateTimeField('date published')
6
7  class Choice(models.Model):
8      poll = models.ForeignKey(Poll)
9      choice_text = models.CharField(max_length=200)
10     votes = models.IntegerField(default=0)
```

On top of an environment, the `listings` package also provides a way to format inline text using `\lstinline{}`, replacing the `verb` command. Note that the settings for `\lstinline{}` must be set in the options of `\lstset{}` in the preamble. For more on the `listings` package, please see [the wikibook on the subject](#), or else [the documentation provided by CTAN.](#) Note that while not discussed here, it is possible to create a list of listings (a table of contents of all the listings).

1.3 `Minted` Package

The last environment I'll demonstrate is the `minted` package. It takes code in the environment, and runs it through Python's `pygments` library. The resulting colored LaTeX syntax is placed in a `verbatim` environment.

```python
from django.db import models

class Poll(models.Model):
    question = models.CharField(max_length=200)
    pub_date = models.DateTimeField('date published')

class Choice(models.Model):
    poll = models.ForeignKey(Poll)
    choice_text = models.CharField(max_length=200)
    votes = models.IntegerField(default=0)
```

While `minted` does not provide an inline formatting command, it does provide a shortcut command `mint` for short code snippets.

```python
fib = lambda n: n if n < 2 else fib(n-1) + fib(n-2)
```

The `minted` environment also provides the ability to be inserted into listings, with labels and captions.

```python
from django.db import models

class Poll(models.Model):
    question = models.CharField(max_length=200)
    pub_date = models.DateTimeField('date published')

class Choice(models.Model):
    poll = models.ForeignKey(Poll)
    choice_text = models.CharField(max_length=200)
    votes = models.IntegerField(default=0)
```

Listing 1.1: `models.py` from Django Tutorial

List of listings

Simplified BSD License ⓒ2013, Andrew Pinkham