

# Examples of Code Listings in L<sup>A</sup>T<sub>E</sub>X

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 T<sub>E</sub>X files with their output in a PDF. Users may then easily read, learn, or simply copy from the T<sub>E</sub>X 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

<b>1</b>	<b>Code Listing</b>	<b>1</b>
1.1	verbatim Environment . . . . .	1
1.2	listings Package . . . . .	2
1.3	Minted Package . . . . .	3

## List of listings

1	models.py from Django Tutorial . . . . .	2
1.1	models.py from Django Tutorial . . . . .	3

## 1 Code Listing

### 1.1 **verbatim** Environment

Here is an example that uses L<sup>A</sup>T<sub>E</sub>X'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 L<sup>A</sup>T<sub>E</sub>X's backtick-quote combination. This is possible because of the upquote package, included in the preamble of the document.

L<sup>A</sup>T<sub>E</sub>X further supplies the `\verb` command, allowing for inline verbatim formatting.

On a related note, 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](#). An example is demonstrated in the caption of listing 1.1 on page 3.

Finally, 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. We could use `\texttt` in captions as well, but that would defeat our purpose of demonstrating all the tools.

## 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 L<sup>A</sup>T<sub>E</sub>X syntax is placed in a 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)
```

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

```
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.

```
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

Simplified BSD License ©2013, Andrew Pinkham