

---

# that Documentation

*Release 1.0.7*

**Daniel Greenfeld**

December 19, 2016



<b>1</b>	<b>Installation</b>	<b>3</b>
<b>2</b>	<b>Usage</b>	<b>5</b>
<b>3</b>	<b>Contributing</b>	<b>7</b>
<b>4</b>	<b>Table of Contents</b>	<b>9</b>
<b>5</b>	<b>Indices and tables</b>	<b>11</b>



Warning: This is a joke. Don't use these bits unless you want to be hated.

Contents:

```
>>> import that
```

```
The Anti-Zen of Python, by Daniel Greenfeld
```

```
Ugly is better than beautiful.
```

```
Implicit is better than explicit.
```

```
Complicated is better than complex.
```

```
Complex is better than simple.
```

```
Nested is better than flat.
```

```
Dense is better than sparse.
```

```
Line code counts.
```

```
Special cases are special enough to break the rules.
```

```
Although purity beats practicality.
```

```
Errors should always pass silently.
```

```
Spelchek iz fur losers.
```

```
In the face of ambiguity, one guess is as good as another.
```

```
There should be many ways to do it.
```

```
Because only a tiny minority of us are Dutch.
```

```
Get things running, then fix them later.
```

```
If the implementation is hard to explain, it's enterprisey.
```

```
If the implementation is easy to explain, it won't take enough time to do.
```

```
Namespaces are too hard, just use "from module import *"!
```



---

## Installation

---

Installing from PyPI:

```
pip install that
```

Or if you want to try and play with the awesome source code:

```
git clone git@github.com:pydanny/that.git
```





---

**Usage**

---

From the Python shell:

```
import that
```

Enjoy the wisdom!



---

**Contributing**

---

Clearly, the code that generates *THAT* is too straightforward. If you can make it better, then do so. Just don't do anything like add tests or document the code.



---

## Table of Contents

---

### 4.1 Inspiration

These are the people who have made this project possible (in order of appearance):

#### 4.1.1 Chris Shenton

For re-introducing me to Python in the summer of 2005.

#### 4.1.2 Zed Shaw

For helping me to remember never to take myself or this community that seriously.

#### 4.1.3 Jacob Kaplan-Moss

For calling me up on beach vacation in the August of 2010 and telling me to fix my code.

#### 4.1.4 Audrey Roy

For showing me daily how smart people do it.

#### 4.1.5 Raymond Hettinger

For demonstrating to me that after 5 years of doing Python professionally I barely qualify for the basic class.

### 4.2 Contributors

These are the people who make this all possible

#### 4.2.1 Project Lead

- Daniel Greenfeld

## 4.2.2 Contributors

- Steve Holden

---

## Indices and tables

---

- `genindex`
- `modindex`
- `search`