

---

# **Human Bingo Documentation**

***Release 3.0.2***

**Matthew Leingang**

**Jun 07, 2019**



---

## Contents

---

<b>1</b>	<b>Human Bingo</b>	<b>3</b>
1.1	Synopsis . . . . .	3
1.2	Features . . . . .	3
1.3	Usage . . . . .	3
1.4	Documentation . . . . .	4
1.5	License . . . . .	4
1.6	Credits . . . . .	4
<b>2</b>	<b>Installation</b>	<b>5</b>
2.1	Environment . . . . .	5
2.2	Stable release . . . . .	6
2.3	From sources . . . . .	6
<b>3</b>	<b>Usage</b>	<b>7</b>
3.1	Command-line application . . . . .	7
3.2	As a python package . . . . .	8
<b>4</b>	<b>Contributing</b>	<b>9</b>
4.1	Types of Contributions . . . . .	9
4.2	Get Started! . . . . .	10
4.3	Pull Request Guidelines . . . . .	11
4.4	Tips . . . . .	11
<b>5</b>	<b>History</b>	<b>13</b>
5.1	Version 3.0.2 (June 7, 2019) . . . . .	13
5.2	Version 3.0.1 (August 24, 2017) . . . . .	13
5.3	Version 3.0.0 (August 24, 2017) . . . . .	13
5.4	Version 2.1.0 (August 4, 2017) . . . . .	13
5.5	Version 2.0.2 (August 3, 2017) . . . . .	14
5.6	Version 2.0.1 (July 28, 2017) . . . . .	14
5.7	Version 2.0.0 (July 28, 2017) . . . . .	14
5.8	Version 1.1.0 (July 26, 2017) . . . . .	14
5.9	Version 1.0.0 (January 27, 2013) . . . . .	14
5.10	2007 . . . . .	14
5.11	2006 . . . . .	14
<b>6</b>	<b>Indices and tables</b>	<b>15</b>



Contents:



# CHAPTER 1

---

## Human Bingo

---

Create cards for the “Human Bingo” icebreaker game, with customizable properties and templating.

### 1.1 Synopsis

This is an icebreaker game to play on the first day of class. See [About.com](#) or [WikiHow](#) for references.

### 1.2 Features

- Read cardset specifications as XML and YAML files. The YAML specification is newer, more flexible, and easier to write. The XML specification is older, and seemed like a good idea at the time.
- Write HTML and PDF files. Since it’s more likely the cards will be printed onto paper, the default is writing PDF. Originally, only HTML files were created, and PDF conversion was done by a third-party utility ([wkhtmltopdf](#)). In version 3 the PDF writer was baked in. Since the PDF writer internally creates HTML first, the HTML feature is still maintained.

### 1.3 Usage

Installing the package exposes a script `humanbingo`. Call it like this:

```
$ humanbingo [-n|--number NUM]
               [-v|--verbose]
               [-d|--debug]
```

(continues on next page)

(continued from previous page)

```
[ -o|--output ] OUTPUTFILE
[ -f|--input-format ] (yaml|xml)
[ -F|--output-format ] (pdf|html)
INPUTFILE
```

Read properties file as its sole argument. If neither `-n` or `--number` option is not specified, write one bingo card to stdout. If `-n NUMBER` or `--number=NUMBER` is specified, writes `NUMBER` cards to separate files.

**Warning:** At this time writing PDF to stdout is not supported. But you probably don't want to do that anyway.

To easily print the PDF cards, they can be concatenated with `pdftk`:

```
$ /path/to/pdftk card*.pdf cat output allcards.pdf
```

## 1.4 Documentation

See <https://human-bingo.readthedocs.io>.

## 1.5 License

Free software: MIT license

## 1.6 Credits

Human bingo relies on a lot of big python packages, heartily endorsed.

- [Jinja2](#) for templating
- [Click](#) for publishing command-line applications
- [WeasyPrint](#) for converting HTML to PDF

We also use [ruamel.yaml](#) over `PyYaml` since it supports a few more features (notably, the `!!omap` sequence declaration).

This package was created with [Cookiecutter](#) and the [audreyr/cookiecutter-pypackage](#) project template.



### 2.1 Environment

When installing software, you typically have two choices: install as root, or install as user in your own user space. You may only have one choice if you don't have root privileges on your machine.

If you want to install `humanbingo` in the system's executable domain, you will probably need to preface any command below with `sudo`. If you already know what that means, proceed with caution. If you don't, consider one of the alternatives, which are designed to prevent you from wrecking your system unintentionally.

To install in user space, you will need to add `--user` to the installation commands. The install scripts will put the files in the places within your user directory that will automatically be searched.

But `humanbingo` relies on other python packages to run correctly. Installing `humanbingo` will automatically download and install those packages as well. This can be trouble if you have other versions of those packages already installed, and different modules require different versions.

So as an alternative, python has *virtual environments*. This is a way to set up a complete python sandbox with packages and executables. Here is how to do that.

You should have a `virtualenv` executable on your system. Run it and give it the name of a directory, in which the environment will reside:

```
$ virtualenv ~/Library/virtualenv/humanbingo
```

or, from another unix:

```
$ virtualenv ~/lib/virtualenv/humanbingo
```

Not sure of the best place for Windows. This will give you a shiny new python installation that is independent from any others on your machine. To activate it, change into the directory specified above and run:

```
$ source bin/activate
(humanbingo)$ which python
...
```

Notice how the prompt changes to indicate the active virtual environment. Your `which` command should return the directory of your virtual environment, with `bin/python` attached.

You can learn more about this from the [Virtualenv user guide](#). Trust me, it's worth it.

## 2.2 Stable release

To install Human Bingo, run this command in your terminal:

```
$ pip install humanbingo
```

Along with using a virtual environment as above, is the preferred method to install Human Bingo, as it will always install the most recent stable release.

If you don't have `pip` installed, this [Python installation guide](#) can guide you through the process.

## 2.3 From sources

The sources for Human Bingo can be downloaded from the [Github repo](#).

You can either clone the public repository:

```
$ git clone git://github.com/leingang/humanbingo
```

Or download the [tarball](#):

```
$ curl -OL https://github.com/leingang/humanbingo/tarball/master
```

Once you have a copy of the source, you can install it with:

```
$ python setup.py install
```

## 3.1 Command-line application

First, create the cardset specification file. The recommended format is YAML.

Here is an example such file:

```
name: Human Bingo
instructions: |
    Find people in the class (other than yourself) who match the descriptions
    in each box; write their name in the box.

    Try to fill as many boxes as possible (one box per person). Five in a row wins!
properties: !!omap
# First-level item keys are the category names
# Second-level items are the values for those categories
- Origin:
    - New York City
    - New Jersey
    - US East Coast
    - US West Coast
    - US Midwest
    - US South
    # ...
- Housing:
    # First Year Halls
    - Brittany Hall
    - Founders Hall
    - Goddard Hall
    # ...
- Math:
    - thinks that math is cool
    - thinks that math is for losers
    - is required to take this class
```

(continues on next page)

(continued from previous page)

```
- is math anxious
# ...
- Interests:
# ...
- Potpourri:
  - has a blog
  - has a Windows Phone
  - has a fidget spinner
# ...
free_space_value: Wants to get an A in this class
```

The package was created to create icebreaker activities in an undergraduate math class. That is why the categories are related to math, or college, or things college students may like to do.

You must have five categories; more or fewer is not supported at this time. For each category, you must have at least five values. A regular bingo card has fifteen possible values (numbers) in each category (letter). Feel free to have many more values in a category (especially if they form a complete list), but don't be so obscure that a particular card becomes impossible to fill.

Save this into a file called, for instance, `spec.yml`.

To make a single card file as PDF, use

```
$ humanbingo -n 1 spec.yml
```

This will result in a file `card01.pdf` in the same directory.

It's more fun if different people have different cards, though. So if you have 20-30 people in the room, use

```
$ humanbingo -n 30 spec.yml
```

and you will get 30 PDFs named `card01.pdf` through `card30.pdf`.

You can print those files individually, or, if you have the `pdftk` utility installed, you can concatenate them all into a single file:

```
$ /path/to/pdftk card*.pdf cat output allcards.pdf
```

Then print the one file.

## 3.2 As a python package

To use Human Bingo in a project:

```
import humanbingo
```

See the package modules documentation for the full API.

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given. You can contribute in many ways:

## 4.1 Types of Contributions

### 4.1.1 Report Bugs

Report bugs at <https://github.com/leingang/humanbingo/issues>.

If you are reporting a bug, please include:

- Your operating system name and version.
- Any details about your local setup that might be helpful in troubleshooting.
- Detailed steps to reproduce the bug.

### 4.1.2 Fix Bugs

Look through the GitHub issues for bugs. Anything tagged with “bug” and “help wanted” is open to whoever wants to implement it.

### 4.1.3 Implement Features

Look through the GitHub issues for features. Anything tagged with “enhancement” and “help wanted” is open to whoever wants to implement it.

### 4.1.4 Write Documentation

Human Bingo could always use more documentation, whether as part of the official Human Bingo docs, in docstrings, or even on the web in blog posts, articles, and such.

### 4.1.5 Submit Feedback

The best way to send feedback is to file an issue at <https://github.com/leingang/humanbingo/issues>.

If you are proposing a feature:

- Explain in detail how it would work.
- Keep the scope as narrow as possible, to make it easier to implement.
- Remember that this is a volunteer-driven project, and that contributions are welcome :)

## 4.2 Get Started!

Ready to contribute? Here's how to set up *humanbingo* for local development.

1. Fork the *humanbingo* repo on GitHub.
2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/humanbingo.git
```

3. Install your local copy into a virtualenv. Assuming you have virtualenvwrapper installed, this is how you set up your fork for local development:

```
$ mkvirtualenv humanbingo
$ cd humanbingo/
$ python setup.py develop
```

4. Create a branch for local development:

```
$ git checkout -b name-of-your-bugfix-or-feature
```

Now you can make your changes locally.

5. When you're done making changes, check that your changes pass flake8 and the tests, including testing other Python versions with tox:

```
$ flake8 humanbingo tests
$ python setup.py test or py.test
$ tox
```

To get flake8 and tox, just pip install them into your virtualenv.

6. Commit your changes and push your branch to GitHub:

```
$ git add .
$ git commit -m "Your detailed description of your changes."
$ git push origin name-of-your-bugfix-or-feature
```

7. Submit a pull request through the GitHub website.

## 4.3 Pull Request Guidelines

Before you submit a pull request, check that it meets these guidelines:

1. The pull request should include tests.
2. If the pull request adds functionality, the docs should be updated. Put your new functionality into a function with a docstring, and add the feature to the list in README.rst.
3. The pull request should work for Python 2.6, 2.7, 3.3, 3.4 and 3.5, and for PyPy. Check [https://travis-ci.org/leingang/humanbingo/pull\\_requests](https://travis-ci.org/leingang/humanbingo/pull_requests) and make sure that the tests pass for all supported Python versions.

## 4.4 Tips

To run a subset of tests:

```
$ python -m unittest tests.test_humanbingo
```





### 5.1 Version 3.0.2 (June 7, 2019)

Fixed a bug where PDF output with no number of cards specified tried to write PDF content to stdout.

### 5.2 Version 3.0.1 (August 24, 2017)

Updated history file for version 3.0.0.

### 5.3 Version 3.0.0 (August 24, 2017)

Added the PDF writer. Pass `-F pdf` to explicitly write PDF files.

Backwards-incompatible changes: YAML input and PDF output are now default.

Improved refactoring and documentation.

### 5.4 Version 2.1.0 (August 4, 2017)

Added the ability to specify card sets with a YAML file instead of the old XML format. Pass `-f yaml` to `humanbingo`.

Lots of improvements to documentation.

## 5.5 Version 2.0.2 (August 3, 2017)

Pretty big refactoring into separate modules, in order to facilitate extensibility, practice dependency injections, etc. But no functional change yet.

## 5.6 Version 2.0.1 (July 28, 2017)

Flesh out history file.

## 5.7 Version 2.0.0 (July 28, 2017)

Name change: “Name Bingo” to “Human Bingo”. The latter sounds more interesting and seems to be the more popular term.

Morphing this script into a bona fide python package to upload to PyPI. Using the [pypackage](#) cookiecutter, add a `setup.py` script so that it can be installed with [pip](#).

Now `pip install` will install an executable file `humanbingo`, and the original `bingo.py` module/script is gone. So this is a backwards-incompatible change.

## 5.8 Version 1.1.0 (July 26, 2017)

- Port from Python 2.7 to 3.5
- Use the shell environment’s python instead of a hardcoded one

## 5.9 Version 1.0.0 (January 27, 2013)

- parse script arguments with [argparse](#)
- get closer to POSIX compliant interface (print to stdout when only one card is needed)
- add a README
- upload to github

## 5.10 2007

MPL converted the ColdFusion script to python as an exercise.

## 5.11 2006

Derek Bruff wrote the original “Name Bingo” script in ColdFusion.

## CHAPTER 6

---

### Indices and tables

---

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