

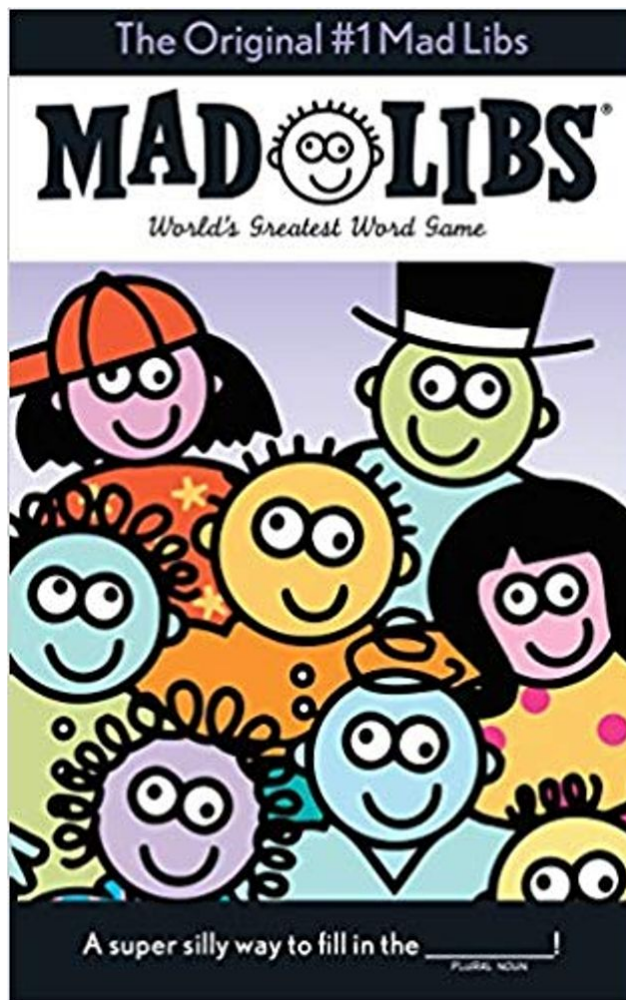


[Download our Starter Code](#)

## Flask Madlibs Exercise

In this exercise, you'll use Flask to make a Madlibs game.

### About Madlibs



In Madlibs, you're asked a series of questions, like this:

```
plural_noun:turnips verb:juggle
```

Those are then plugged into a story template, like this:

```
I love to {verb} {plural_noun}.
```

To create a story like:

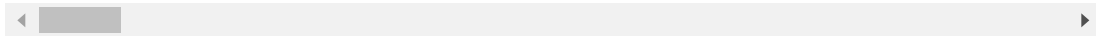
```
I love to juggle turnips.
```

## Code

We've given you some code to help with the core non-Flask-specific Madlibs part:

*stories.py*

```
"""Madlibs Stories.""" class Story: """Madlibs story. To make a story
```



This allows you to define Madlibs stories, and it can generate the resulting story from a set of answers. (It's also a nice example of a small but useful class!)

We've created a story, *story*, in that file.

**Warning:** Stop and explore here! Before starting to make a Flask app, make sure you understand how this *Story* class works — go into *ipython* and try out the *generate* method on our sample story to get a feel for the text-generating process for Madlibs.

## Challenge

Write a Flask app that imports the example story. Add a homepage for the application that shows a form prompting you for all the words in the story:

# Madlibs

place:

noun:

verb:

adjective:

plural\_noun:

Don't **hardcode this, though** — you want your form route to be able to ask for all of the questions required by the story, not for it to have a hard-coded form of asking these exact questions!

Add a route, `/story`, that shows the resulting story for those answers, like this:

## Your Story

Once upon a time in a long-ago condo, there lived a large frozen dragon. It loved to eat popsicles.

For now, don't worry about having template inheritance or a *base.html* — later, in further study, you can refactor this to use template inheritance.

## Further Study

## Use Template Inheritance

Make a *base.html* template of common parts of your templates (like the `<html>`, `<body>`, and other common things, and change your templates so they inherit from this base template.

## Allow User to Pick Story

Add a feature where there are several different story templates, rather than just one.

The homepage should change to a drop-down menu of the story templates. When the user picks a template, it should go to the page that prompts for the list of story questions. That should, as before, go to the page that shows the generated story.

## Add CSS

**Still want more?** Add some CSS to your madlibs, storing the CSS file in a *static/* directory and referencing it properly, so Flask will serve it up.

## Additional Further Study

**What? More time?** Add some JS to your madlibs – perhaps you can validate the form (make sure every question is answered, all answers are at least 3 characters long, all lowercase, etc) before you're allowed to submit the form.

**Even more, you say???** Try to add a page to your application where users can create their own madlibs, by providing a list of parts of speech, along with the text of the story. Submitting this form should create a new story instance that you could then select from the dropdown of stories.

**Note: This will be challenging given what you know now!** Consider it a super bonus. Also, because we don't yet know how to persist data, if you store your stories on the server in a list, that list will get reset every time you restart your server.