BEW 1.1 Quiz 2: *Flask & Templating*

MAKE SCHOOL

Name: ______________________________ **Section:** ________ **Score:** ______ **/ 30**

1. (*10 Points*) You decide to contribute to a friend's project called "PawPals", which is a social network for dogs. In the template for the search page, you see the following HTML form:

```
<form action="/search">
  <label for="breed">Breed:</label>
  <input type="text" name="breed">Labrador</input>

  <label for="distance">Distance away (miles):</label>
  <input type="number" name="distance" min="1" max="50">15</input>
  <button type="submit">Search!</button>
</form>
```

   a. (*5 Points*) If the user submits the form with the default values, what URL will they be sent to? Include the query string. (Note: Multiple query parameters can be separated by a '&'.)

```
http://mypawpals.com/___________?_______________________________________
```

   b. (*5 points*) Your friend wrote the PawPals site using Flask, and has a route for the search page. Using your answer from part a), how would we extract the query parameters in Python code? Fill in all blanks to complete the function.

```
@app.route('____________')
def search_page():
    breed = ___________________________________________
    distance = ___________________________________________

    pals_list = ... # code to get data here
    return render_template('search_page.html', pals_list=pals_list)
```

2. (*10 Points*) We now want to display our PawPals data on the search page! You discover that the data for showing PawPals is in the form of a list of dictionaries. The sample data looks like this:

```
pals_list = [
    {
        "name": "Ducky",
        "owner": "Jordan",
        "img_url": "http://mypawpals.com/images/7A2B6C",
        "is_cat_friendly": True
    },
    {
        "name": "Moxie",
        "owner": "Dani",
        "img_url": "http://mypawpals.com/images/F4903D",
        "is_cat_friendly": False
    },
    ... # more dog friends here!
]
```

Write the code to display the search results using Jinja template control flow structures. This code should use a *for* loop to display the Pals. Each Pal should be displayed inside of a <div> and should contain the Pal's image, name, and owner's name. *If* the Pal is cat-friendly, the page should display a sentence stating so.

(*Hint:* To get a dictionary field named 'breed', use `name_of_dictionary['breed']`)

```
<!-- templates/search_page.html -->
<div id="pals_container">
   _______________________________________
       <div class="pal">
           <img src="http://example.com" alt="Picture of Pal" />
           Name: _______________________________________
           <br/>
           Owner: _______________________________________
           _______________________________________
               This dog is cat-friendly!
           _______________________________________
       </div>
   _______________________________________
</div>
```

Name: ____ **Section:** ________

3. (*10 Points*) In your own words, **identify 2 reasons why a developer would use HTML Jinja templates**, as opposed to including HTML inline in a Python file:

   - **Reason #1**:

     ____

     ____

     ____

     ____

     ____

     ____

   - **Reason #2**:

     ____

     ____

     ____

     ____

     ____

     ____