

cs350 - exercise 1: c9.io and Python Flask

Objective:

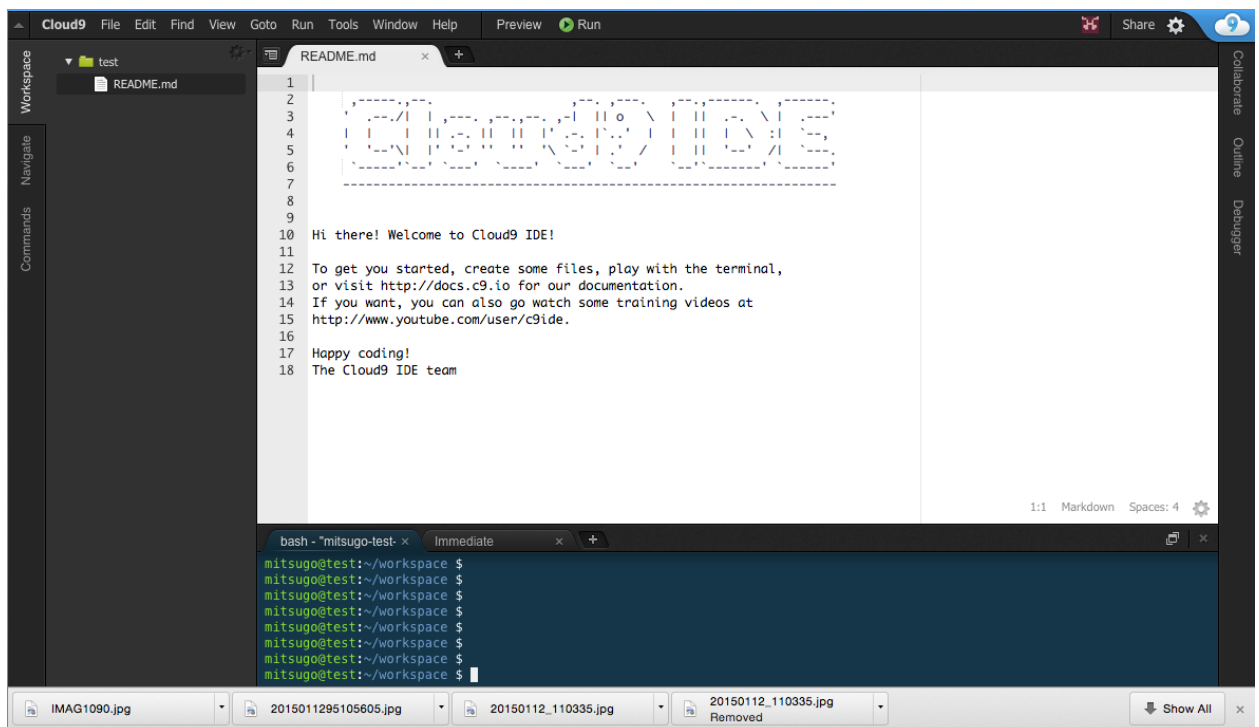
To get familiar with cloud9 (c9.io)

To learn how to build a simple website with flask.

Part 1 - getting started with c9 and flask

The checklist

1. ☐ create a c9.io account
2. ☐ create a new workspace (a hosted custom workspace)
3. ☐ select the 'start editing' link to get into your workspace.



4. ☐ the result will look similar to the above image. The white part is the editor. Below that (with the `mitsugo@test:~/workspace` prompt in the image above) is the terminal window. To the left is the file browser. Verify that Python is installed by doing something like adding 2 and 2:

```
bash - "mitsugo-test- x Immediate x +
mitsugo@test:~/workspace $ python
Python 2.7.6 (default, Mar 22 2014, 22:59:56)
[GCC 4.8.2] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> 2 + 2
4
>>> exit()
mitsugo@test:~/workspace $
```

5. ☐ Now install Flask. In the terminal window type: `sudo easy_install flask` markdown

6. ☐ Create a new file with the following code (I named mine `server.py`):

```
1 import os
2 from flask import Flask
3 app = Flask(__name__)
4
5 @app.route('/')
6 def hello_world():
7     return 'Hello World!'
8
9 if __name__ == '__main__':
10     app.run(host=os.getenv('IP', '0.0.0.0'), port=int(os.getenv('PORT', 8080)), debug=True)
```

```
from flask import Flask, render_template
import os
```

```
app = Flask(__name__)
```

```
@app.route('/')
def mainIndex():
    return "<h1 style = \"color:red\" >Hello there!</h1>"
```

```
if __name__ == '__main__':
    app.run(host=os.getenv('IP',
'0.0.0.0'),port=int(os.getenv('PORT', 8080)), debug=True)
```

7. ☐ run your code (`python server.py`) and preview your app

8. ☐ The `@app.route('/')` provides Python code for when a user goes to the root level (for ex., <https://test-mitsugo.c9.io>). What happens if we want some text when the user goes to <https://test-mitsugo.c9.io/about> or <https://test-mitsugo.c9.io/news>? We add:

```
@app.route('/about')
def all_about_me(): #name is arbitrary
```

```
    blah blah blah
```

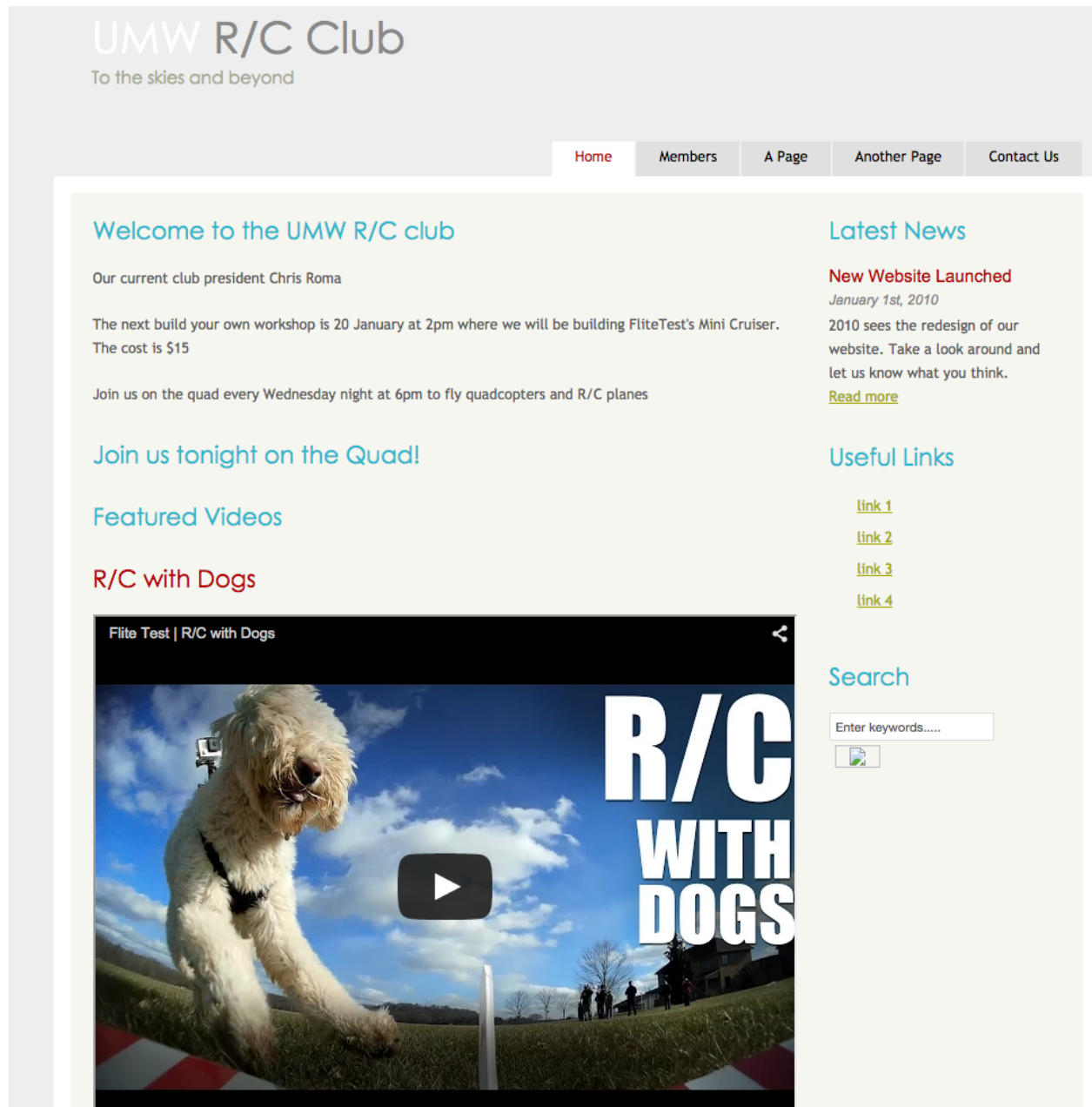
```
@app.route('/news')  
def newsy():  
    blah blah blah
```

Add code for 3 more pages. DEMO

XP 30 * percent of team having this task complete

Congratulations!!

Part 2: Flask templates



During the in-class tutorial you learned how to get Flask working with templates.

XP: 40xp times the percent of team demoing on their laptops. for example, if all the people in a team demo the project the team will get 50. If only half demo, the team will get 20.

WHAT TO DO

1. clone the code repository on your cloud9 box by doing the following in the terminal

git clone <https://github.com/zacharski/templateIntro.git>
and make sure it works.

2. This code is designed to work with flask running on my laptop, not on cloud9. Alter the app.run line to have it work in cloud9.
3. Without altering index.html change the president's name to someone else.
4. In fact, the club doesn't meet always on Wednesday's. I'd like you to delete that line and replace it with an <h3> line telling the next meeting's details, which are passed to the template by server.py. For example:

Join us Saturday Jan17th at 10am at the Athletic Field

On the Python side, the details **should be represented by a Python dictionary containing the date, time, and location of the next meeting.**

5. Underneath each youtube video is a short description. I have some sort of bug and the description is not appearing beneath the Argonay video. Fix it.
6. Add another feature to the Python dictionaries describing each video (for example, a postedBy feature or a location feature). Have this information displayed on the webpage.
7. There is a link on the top of the page to Members. I want you to add that page and populate it with a list of current members generated from a list on the python side.
8. DEMO

Part 3: INDIVIDUAL homework (to be done on your own).

XP possible for both technical and creative work.

1. OPTIONAL CREATIVE PART 20XP

Can you revise the site so it is about a completely different topic? The same features (passing variables, loops, conditionals) should be present. And the site should have 2

working pages.

2. OPTIONAL CREATIVE PART 25XP

The worksheet uses the simple_light template, which I got off the web at <http://www.html5webtemplates.co.uk/templates.html#.UtckmmRDsqQ>

I downloaded the template then flask-ified it. For example, I divided the code in the original index.html to a part that stays the same for all pages on the site (layout.html) and a part the is specific to index.html (the index.html file). I also added some flask specific code to these files. I moved the static resources like the css file and the images to the flask static directory. Can you find a template on the web and flask-ify it?

3. **REQUIRED TECHNICAL COMPONENT up to 30XP** Right now there are only 2 pages that work. Please create 2 additional pages. There are two menu items “A Page” and “Another Page”-- please change both the text and the link so they go to your new pages. The new pages should demonstrate your abilities to use Flask variables, conditionals and loops. In addition, if you add content to the site, fix the links in the right column, or any task that shows that you are learning something you will get additional XP.

DEMO: please demo the homework in one of the following ways:

1. Tuesday, from 10 minutes before class to 10 minutes after class starts.
2. Tuesday from 12:15 to 1:45
3. email a screencast demoing your site, and a zip file of your code to submit.o.bot@gmail.com by Tuesday at 11:59pm.

my rough notes

templates

often we want to separate how things look from the logic of our program. that way a designer can work on how things look and you can work on what you are good at, coding.

In flask we can create templates – they go in a templates directory

```
<html>
<head>
<title>{{title}} - microblog</title>
</head>
<body>
<h1>Hello, {{user.nickname}}!</h1>
</body>
</html>

@app.route('/hello/<title>')
def helloz(title):
    user = {'nickname': 'Lefty'}
    return render_template('hello2.html', title=title, user=user)
```

If statements

```
<html>
  <head>
    {% if title %}
    <title>{{title}} - microblog</title>
    {% else %}
    <title>Welcome to microblog</title>
    {% endif %}
  </head>
  <body>
    <h1>Hello, {{user.nickname}}!</h1>
  </body>
</html>
```

Loops

slogans.html

```
<html>
  <head>
    {% if title %}
    <title>{{title}} - microblog</title>
    {% else %}
    <title>microblog</title>
    {% endif %}
  </head>
  <body>
    <h1>Hi, {{user.nickname}}!</h1>
    {% for post in posts %}
    <p>{{post.author.nickname}} says: <b>{{post.body}}</b></p>
    {% endfor %}
  </body>
</html>
```

and the code

```
@app.route('/blog')
def index():
    user = { 'nickname': 'Mitsugo' } # fake user
    posts = [ # fake array of posts
        {
            'author': { 'nickname': 'John' },
            'body': 'Beautiful day in Portland!'
        },
        {
            'author': { 'nickname': 'Susan' },
            'body': 'The Avengers movie was so cool!'
        }
    ]
```

```
    return render_template("slogans.html",  
        title = 'Home',  
user = user,  
    posts = posts)
```