LECTURE 19: WEB BASED APPLICATIONS: CONTINUED

FLASK

- Flask is a micro web framework
- Supports cookies
- Extensions available
- Lots of 3rd party libraries
- Install: pip(or pip3) install flask
- More details: http://flask.pocoo.org/docs/1.0/installation/

FLASK HELLO WORLD

```
from flask import Flask
app=Flask(__name__)
@app.route("/")
def hello():
   return"Hello World!”
if __name__=="__main__":
   app.run()
```

Source: http://flask.pocoo.org/docs/1.0/quickstart/

URL BUILDING & ELIZA EXAMPLE

See lecture18/flask_url_building.py

See lecture18/flask_form.py

Source: http://flask.pocoo.org/docs/1.0/quickstart/

TEMPLATES

- Need to decouple design from the backend code
- Useful for creating dynamic web pages
- Flask uses Jinja2 (http://jinja.pocoo.org/docs/2.10/)
- See template_example.py

ADDING VARIABLE URLS

- What if we wanted to display info for different students?
- See template_dynamic_url.py

WTFORMS SAMPLE FIELDS

- StringField
- TextAreaField
- SubmitField
- BooleanField
- RadioField
- …

BUILDING FORMS THE FLASK WAY

- WTForms for flask
- pip3 install flask-wtf

FORM VALIDATION

- What if an empty form is submitted?
- What if we have a max size limit?
- Solution: use form validation
- form.validate_on_submit()
- https://wtforms.readthedocs.io/en/stable/validators.html

FIN!