Flask Web Application Micro-Framework

Flask is the pre-eminent Python back-end micro-framework, owing that success to community, documentation, and simplicity.

A **Web Framework** encapsulates the components necessary to host a web application: receiving requests, parsing request headers to build a request object, returning an HTTP response, connecting to a database to manage users and data, managing sessions, validating forms, etc, etc, etc.

A **Micro Framework** is a framework that chooses speed and ease of use over full-featured-ness.

Micro frameworks have come en vogue due to the prevalence of micro-services and single page apps, where most functionality is transferred to a front-end framework.

This workshop's code samples and exercises are available at http://github.com/codyhess/flask-workshop

Recommended for further reading:
"Flask Web Development" by Miguel Grinberg
"Python Web Frameworks" by Carlos de la Guardia (free)

Flask VS Express: Hello World

```
from flask import Flask

app = Flask(__name__)

@app.route(‘/’)
def index():
  return “Hello world.”

if __name__ == “__main__”:
  app.run()
```

```
var express = require(‘express’)
var app = express()

app.get(‘/’, (req, res) => {
  res.send(‘Hello world.’)
})

app.listen(‘3000’, () => {
  console.log(‘up and running…’)
});
```

Flask VS Express: Hello World JSON

```
from flask import Flask
from flask import json, request

app = Flask(__name__)

@app.route(‘/’)
def index():
  name = request.args.get(‘name’)
  return json.dumps({
    name: name,
  })

if __name__ == “__main__”:
  app.run()
```

```
var express = require(‘express’)
var app = express()

app.get(‘/’ (req, res) => {
  name = req.query.name
  if (!name) name = “Nobody”
  res.send({“name”: name})
})

app.listen(‘3000’, () => {
  console.log(‘up and JSON…’)
})
```

Circle and label the following components in each section of code:

- ROUTE
- REQUEST
- RESPONSE

List two reasons you might choose to use Flask for a project:

List two reasons you might choose to use Express for a project:

```
from flask import Flask, json, render_template

app = Flask(__name__)
app.debug = True

@app.route('/')
@app.route('/play')
def index():
    return json.dumps({
        'you': 'rock',
        'comp': 'rock',
        'win': 'WIN',
        })

@app.route('/v1')
def v1():
    return render_template('v1.html')

@app.route('/v2')
def v2():
    return render_template('v2.html')

@app.route('/v3')
def v3():
    return render_template('v3.html')

if __name__ == "__main__":
    app.run()
```

```
<head>
  <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.m
in.js">
  </script>
  <script>
var api = 'http://localhost:5000/play?choice=';
$(document).ready( () => {
  $('#choices').on('click', (event) => {
    $.get(api + event.target.id, (data) => {
      data = JSON.parse(data);
      console.log(data);
      $('#you').text('You picked: ' + data.you);
      $('#comp').text('Computer picked: ' + data.comp);
      $('#win').text('You ' + data.win + '!');
    });
  });
});
  </script>
</head>

<body>
  <h1>Rock, Paper, Scissors</h1>
  <ul id="choices">
    <li id="rock">Pick ROCK</li>
    <li id="paper">Pick PAPER</li>
    <li id="scissors">Pick SCISSORS</li>
  </ul>
  <hr>
  <ul id="results">
    <li id="you"></li>
    <li id="comp"></li>
    <li id="win"></li>
  </ul>
</body>
```

In the files v1.html, v2.html, v3.html, how might templates reduce the re-use of code?