

# Web Services, Python, and flask

The background of the slide features a photograph of the Almathea statue at the University of Illinois, which depicts a woman in classical attire with her arms outstretched. The entire image is overlaid with a semi-transparent orange color.

**CS 240 - The University of Illinois**

Wade Fagen-Ulmschneider

October 5, 2021

# Web Services

A photograph of a statue of Alma Mater, a personification of a university, surrounded by a crowd of people. The entire image is overlaid with a solid red color. The statue is the central focus, standing on a pedestal. The people are in the foreground, some looking towards the statue. The background shows trees and a building.

[1]:

```
1 GET /cs240/fa2021/ HTTP/1.1\r\n
2 Host: courses.grainger.illinois.edu\r\n
... ..
```

[2]:

# [2]: RESTful Web Services

National Weather Service API:

```
https://api.weather.gov/points/{latitude},{longitude}
```

Location of 2215 CIF:

```
1 GET /points/ HTTP/1.1\r\n
2 Host: api.weather.gov\r\n
... ..
```

**[3]:**



```
1 POST /q/api/queues/788/staff/1 HTTP/1.1\r\n
2 Host: queue.illinois.edu\r\n
3 Content-Length: 477\r\n
...
{ "id": 30540, "startTime": "2021-10-05T15:48:57.000Z", "endTime": null,
  "createdAt": "2021-10-05T15:48:57.000Z", "updatedAt":
  "2021-10-05T15:48:57.000Z", "userId": 1, "queueId": 788, "user": { "name":
  "Wade A Fagen-Ulmschneider", "id": 1, "netid": "waf", "universityName": "Wade
  A Fagen-Ulmschneider", "preferredName": null, "isAdmin": true, "createdAt":
  "2018-02-14T05:27:54.000Z", "updatedAt": "2018-02-21T05:34:33.000Z" } }
```

# HTTP Action Verbs

The image features a crowd of people gathered around a statue of Alma Mater, which is a common symbol for universities. The entire scene is overlaid with a semi-transparent orange filter. The text 'HTTP Action Verbs' is prominently displayed in the center in a white, bold, sans-serif font. The background shows the intricate details of the statue and the silhouettes of the people in the crowd.

Method	Description	Sec.
GET	Transfer a current representation of the target resource.	4.3.1
HEAD	Same as GET, but only transfer the status line and header section.	4.3.2
POST	Perform resource-specific processing on the request payload.	4.3.3
PUT	Replace all current representations of the target resource with the request payload.	4.3.4
DELETE	Remove all current representations of the target resource.	4.3.5
CONNECT	Establish a tunnel to the server identified by the target resource.	4.3.6
OPTIONS	Describe the communication options for the target resource.	4.3.7
TRACE	Perform a message loop-back test along the path to the target resource.	4.3.8

All general-purpose servers MUST support the methods GET and HEAD.  
All other methods are OPTIONAL.

## [RFC 7231, Section 4](#)

- **GET:** Requests a representation of the specified resource. Requests using GET should only retrieve data.
- **POST:** Submits an entity to the specified resource, often causing a change in state or side effects on the server.

- **HEAD:** Asks for a response identical to a GET request, but without the response body.
- **PUT:** Replaces all current representations of the target resource with the request payload.
- **DELETE:** Deletes the specified resource.
- **PATCH:** Applies partial modifications to a resource.

- **CONNECT:** Establishes a tunnel to the server identified by the target resource.
- **OPTIONS:** Describes the communication options for the target resource.
- **TRACE:** Performs a message loop-back test along the path to the target resource.

# Python Programming

A photograph of a crowd of people gathered around a statue of a woman in a long dress, set against a background of trees. The entire image is overlaid with a semi-transparent orange filter. The text "Python Programming" is centered in white.

# Python

Python is an “**interpreted**” programming language:



# Python

Python is a “**dynamically typed**” programming language:

# Python

Python's control-flow is **whitespace delimited**:

# Python

Python places heavy emphasis on **code readability**:

# 12/hello.py

```
1 s1 = "Hello"  
2 s2 = " World"  
3  
4 for i in range(10):  
5     if i < 5:  
6         print(s1)  
7     elif i < 8:  
8         print(s1 + s2)  
9     else:  
10        print(s2)
```

# 12/app.py

```
1 from flask import Flask
2 app = Flask(__name__)
3
4 # Route for "/" for a web-based interface to this
  micro-service:
5 @app.route('/')
6 def index():
7     from flask import render_template
8     return render_template("index.html")
9
10 # Extract a hidden "uiuc" GIF from a PNG image:
11 @app.route('/extract', methods=["POST"])
12 def extract_hidden_gif():
13     # ...
```

```
1 from flask import Flask
2 app = Flask(__name__)
3
4 # Route for "/" for a web-based interface to this
  micro-service:
5 @app.route('/')
6 def index():
7     from flask import render_template
8     return render_template("index.html")
9
10 # Extract a hidden "uiuc" GIF from a PNG image:
11 @app.route('/extract', methods=["POST"])
12 def extract_hidden_gif():
13     # ...
```

```
1 from flask import Flask
2 app = Flask(__name__)
3
4 # Route for "/" for a web-based interface to this
  micro-service:
5 @app.route('/')
6 def index():
7     from flask import render_template
8     return render_template("index.html")
9
10 # Extract a hidden "uiuc" GIF from a PNG image:
11 @app.route('/extract', methods=["POST"])
12 def extract_hidden_gif():
13     # ...
```

```
1 from flask import Flask
2 app = Flask(__name__)
3
4 # Route for "/" for a web-based interface to this
  micro-service:
5 @app.route('/')
6 def index():
7     from flask import render_template
8     return render_template("index.html")
9
10 # Extract a hidden "uiuc" GIF from a PNG image:
11 @app.route('/extract', methods=["POST"])
12 def extract_hidden_gif():
13     # ...
```



```
1 from flask import Flask
2 app = Flask(__name__)
3
4 # Route for "/" for a web-based interface to this
  micro-service:
5 @app.route('/')
6 def index():
7     from flask import render_template
8     return render_template("index.html")
9
10 # Extract a hidden "uiuc" GIF from a PNG image:
11 @app.route('/extract', methods=["POST"])
12 def extract_hidden_gif():
13     # ...
```