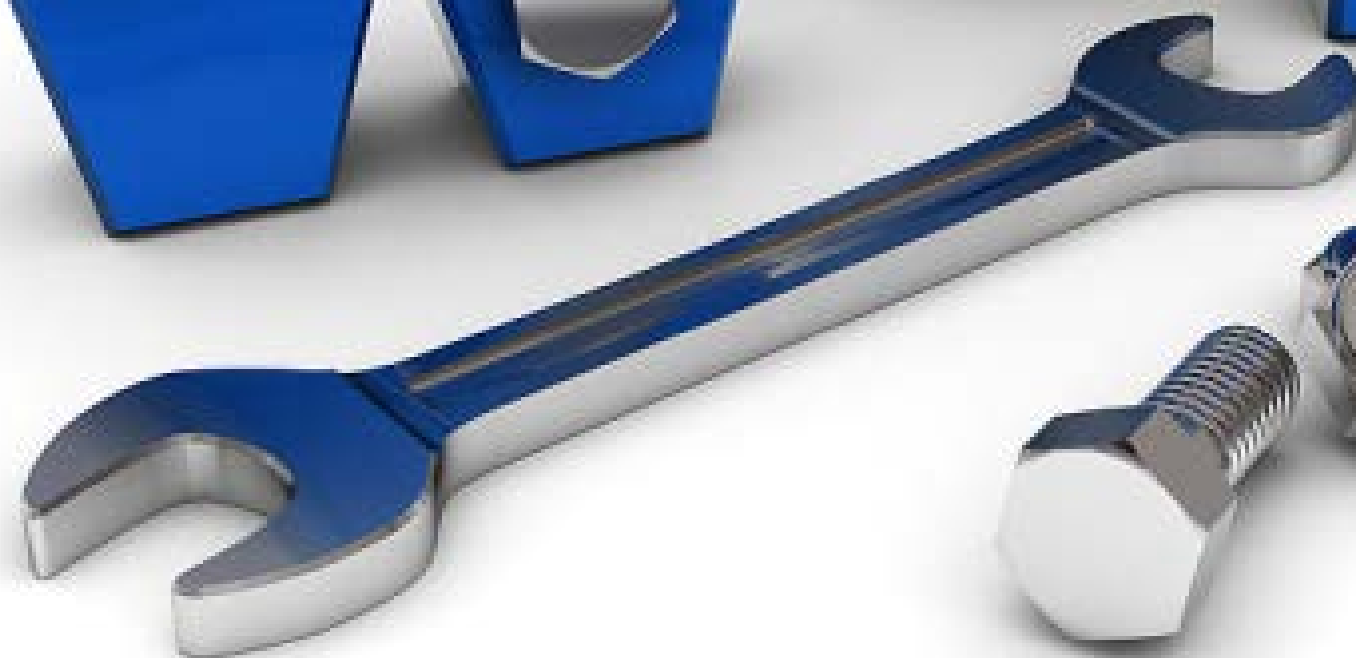


WEB



Browsers



Web-Big Picture



The Client

- Any software that is capable of issuing HTTP requests
- A general purpose software application for requesting HTTP resources and displaying responses to the user is a web browser.

Web Resources

- Page/Document
 - Typically written in Hyper Text Markup Language (HTML) or Extensible Markup Language (XML)
- File/Data
 - Images
 - Music
 - Executable Objects
 - ...

Client Side Technologies

- Document structure / content
 - HTML, XML
- Document styling
 - CSS
- Dynamics
 - Javascript, Java, Flash, Silverlight

HTML

- Markup to support structured documents
- Semantics for . . .
 - Text (headings, paragraphs, lists, tables, etc)
 - Multimedia (images, music, video, etc)
 - Links to other resources
 - Forms
 - Styling
 - Scripting

HTML

Example: (Hello_world.html)

```
<!DOCTYPE HTML>
```

```
<html>
```

```
  <head>
```

```
    <meta http-equiv="Content-type"  
    content="text/html; charset=UTF-8">
```

```
    <title>Computer Science </title>
```

```
  </head>
```

```
  <body>
```

```
    <p>hello</p>
```

```
    <p>world</p>
```

```
  </body>
```

```
</html>
```


HTML Elements

- HTML is a hierarchal bag of elements
 - Start/end tag
 - `<element></element>`
 - Attribute/value pairs
 - `<element key1="value1" key2="value2" .../>`
 - Content
 - `<element key="value">content</element>`

HTML Document Structure

- head
 - title – shown in the browser
 - meta – used for automated processing
 - styling, scripting, etc
- body
 - p – paragraph
 - img – image
 - ul, ol – unordered/ordered list
 - a – "anchor" (link to other resources)
 - forms, tables, etc.

HTML Form

- Define method and destination resource
`<form method = "get/post" action = "url">`
- Provide input form elements
 - Single/multiline input
 - Single/multiple element selection lists
 - Check /radio boxes
 - File upload
 - Buttons

Novell Services Login

Username:

Password:

City of Employment:

Web server:

Please specify your role:

☐ Admin

☐ Engineer

☐ Manager

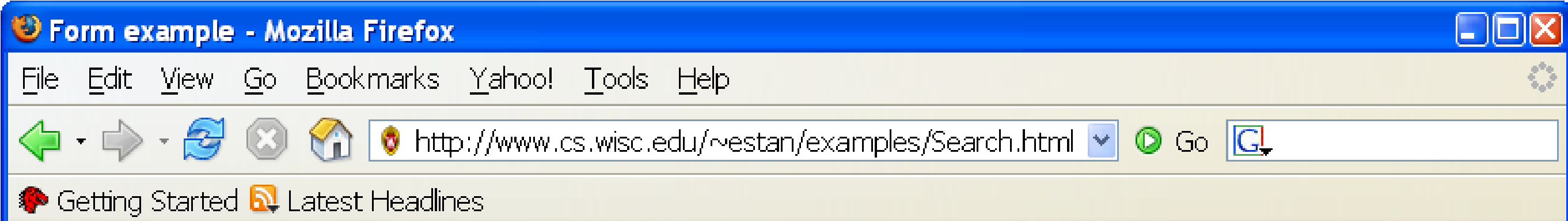
☐ Guest

Single Sign-on to the following:

☐ Mail

☐ Payroll

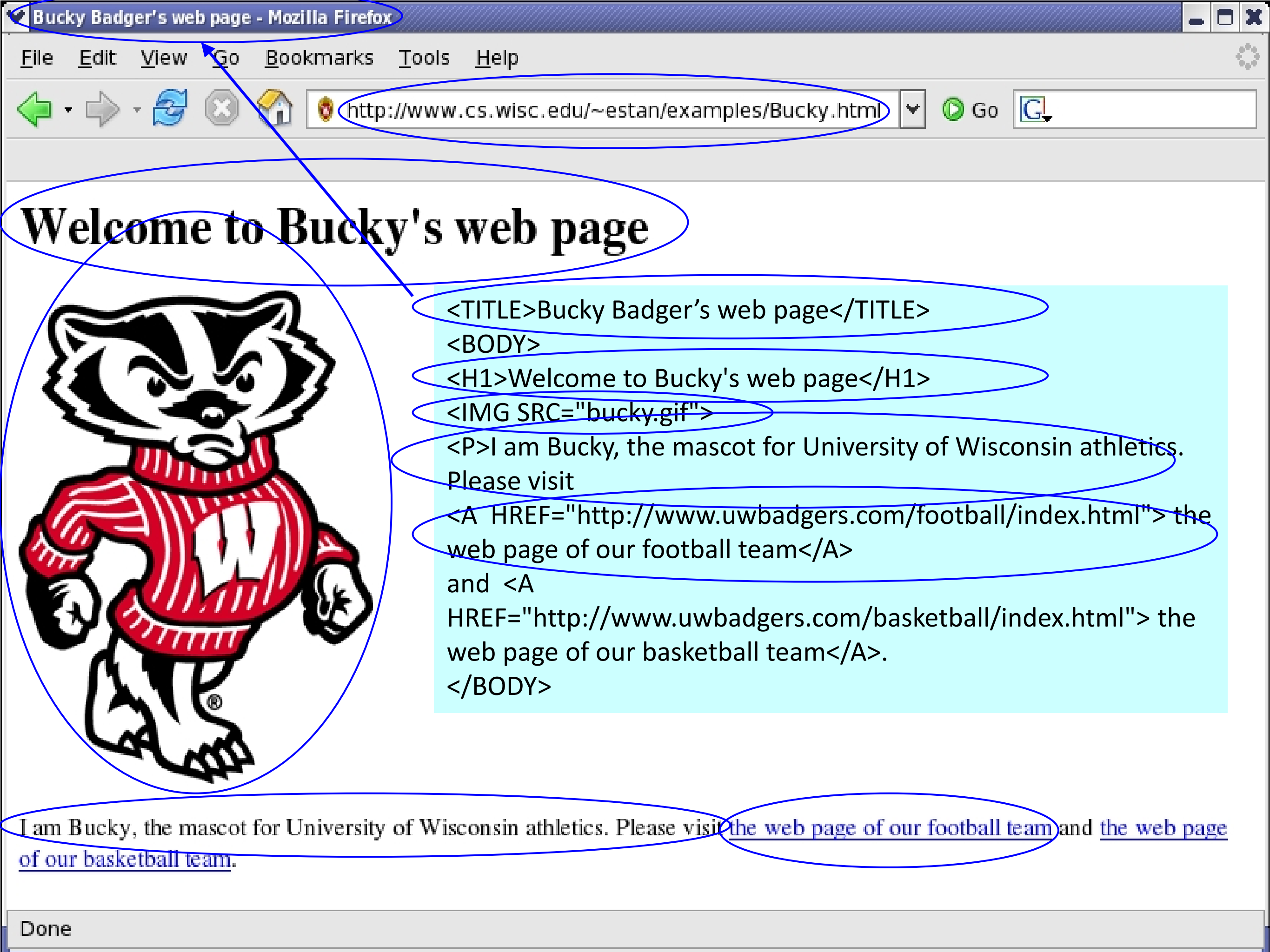
☐ Self-service



Search form

What are you looking for? Your search terms... Search UW-Madison

```
<h3>Search form</h3>
<form method="get"
  action="http://www.google syndicated search.com/u/univwisc">
  <p>What are you looking for?
  <input type="text" name="q" id="searchText" value="Your search terms..." />
  <input type="hidden" name="hl" value="en" />
  <input type="hidden" name="ie" value="ISO-8859-1" />
  <input type="submit" id="searchButton" value="Search UW-Madison" />
</p></form>
```



Welcome to Bucky's web page



```
<TITLE>Bucky Badger's web page</TITLE>
```

```
<BODY>
```

```
<H1>Welcome to Bucky's web page</H1>
```

```
<IMG SRC="bucky.gif">
```

```
<P>I am Bucky, the mascot for University of Wisconsin athletics.  
Please visit
```

```
<A HREF="http://www.uwbadgers.com/football/index.html"> the  
web page of our football team</A>
```

```
and <A
```

```
HREF="http://www.uwbadgers.com/basketball/index.html"> the  
web page of our basketball team</A>.
```

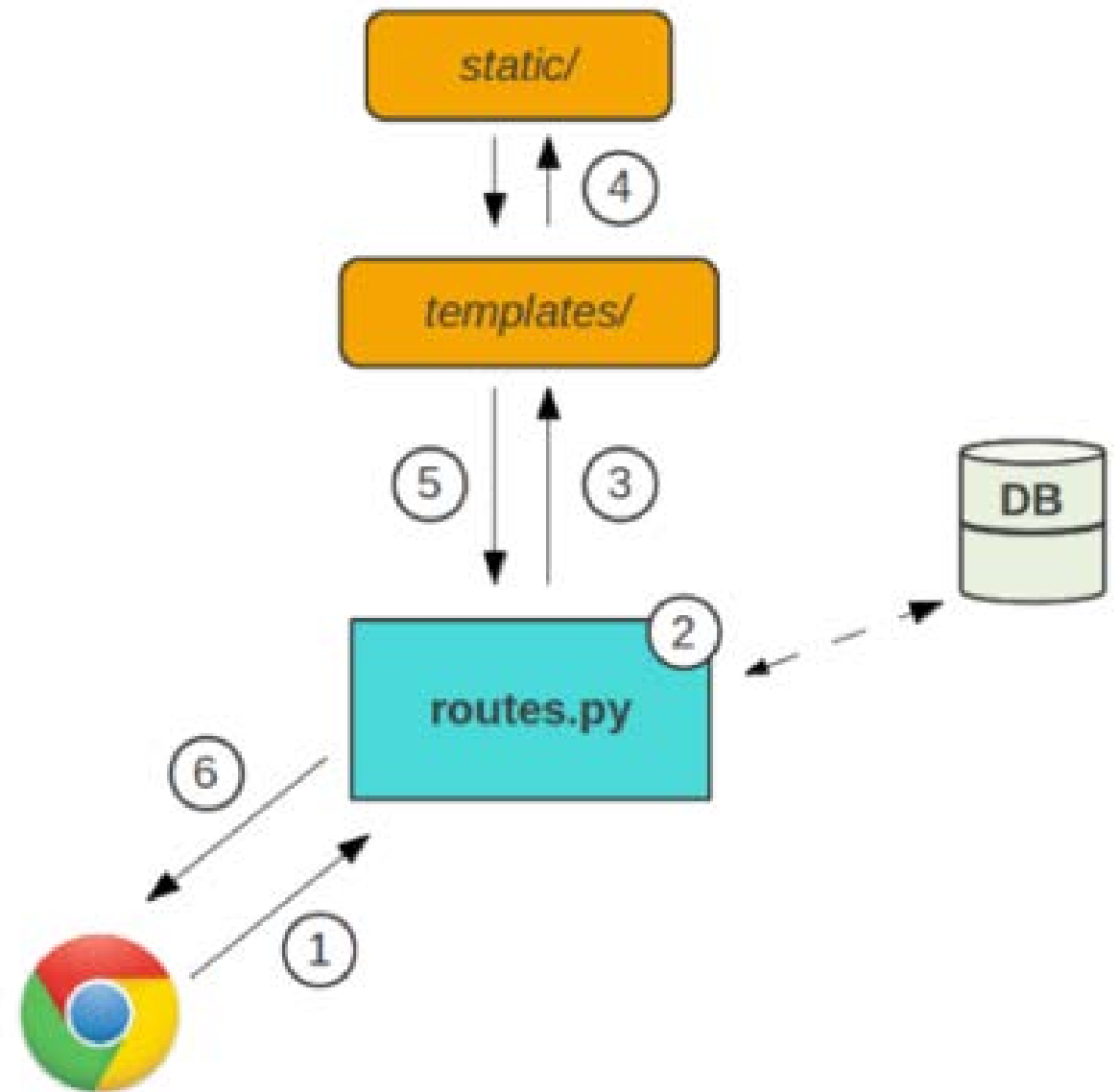
```
</BODY>
```

I am Bucky, the mascot for University of Wisconsin athletics. Please visit [the web page of our football team](http://www.uwbadgers.com/football/index.html) and [the web page of our basketball team](http://www.uwbadgers.com/basketball/index.html).



Flask

1. A user issues a request for a domain's root URL / to go to its home page.
2. routes.py maps the URL / to a Python function.
3. The Python function finds a web template living in the *templates/* folder.
4. A web template will look in the *static/* folder for any images, CSS, or JavaScript files it needs as it renders to HTML
5. Rendered HTML is sent back to routes.py
6. routes.py sends the HTML back to the browser



Flask

Example: (hello.py)

```
from flask import Flask
app = Flask(__name__)

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

On my c:\cygwin\tmp directory, there was a flask.py lying around!

to be continued...