**Back in the days** , to run a python application we used to use CGI and mod\_python. These options tends to be limited and had performance and security issues.

**Enter WSGI: (wiz-gee)**

Web server gateway interface

It’s a spec in software interface between a webserver and a python application.

The point of wsgi to allow any webserver to work with python applications.

This brings the concept of calling python function from webserver.

When the webserver receives a request from the client that should be processed by the python app function. The py functions runs and return the result to web server and the server passes it on to a client.

**Python app which implement the WSGI spec:**

def application(environ, start\_response):

start\_response(‘200 OK’, [(‘Content-Type’, ‘text/plain’)]

return [b’hello world’]

here,

environ – contain info about the request and various setting from wsgi server

start\_response: a function that must be called by the py app to start the HTTp response.

And return value of application function is the actual response body . ofcourse not all HTP have bodies.

If you'd like to actually see a WSGI app run, it's easy to do. First you'll need a WSGI compliant server. I recommend [uWSGI](https://uwsgi-docs.readthedocs.io/en/latest" \t "_blank), which has a simple command line interface. It can be installed by running

$pip install uwsgi

Save the python script I listed earlier into a file called app.py. Run the uWSGI server by typing this on the command line:

$uwsgi –http :8080 –wsgi-file app.py

and browsing to <http://localhost:8080>

Notice that the python app has no import statements. There are no external libraries or frameworks used. It's as plain a python app as they come. The magic is all in the implementation of the WSGI spec. It's important to note that if you don't name the function "application" it won't work. That's the default name that the uWSGI server expects, but it can be configured to work with a different name if you like.

***uWSGI***  
  
[uWSGI](http://uwsgi-docs.readthedocs.io/en/latest/index.html) is a popular web server that implements the WSGI standard. Don't get confused by the name. WSGI is a specification, uWSGI is a web server. That little "u" in the front makes a big difference. It's pretty common to pair Flask and uWSGI since they both talk WSGI. The uWSGI server is a full featured HTTP server that is quite capable of running production web apps. However, it's not as performant as nginx at serving static content, so it's pretty common to see nginx sitting in front a uWSGI server.

**uwsgi**  
  
Here's where some poor naming choices make things even more confusing. So we know WSGI is a software spec, uWSGI is a server, so what the hell is uwsgi? When it's spelled using all [lowercase letters](http://uwsgi-docs.readthedocs.io/en/latest/Protocol.html), it refers to a binary protocol for connecting the uWSGI server to other applications. As you no doubt recall from the above paragraph, it's common for uWSGI to be used in conjunction with nginx. Well, the uwsgi protocol is how the two web servers talk to each other.

