WEB DEVELOPMENT SHOULD BE

Massimo Di Pierro

School of Computing and Digital Media

DePaul University

Chicago, IL

FIRST OF ALL....

Congratulations to Mariano
new member of the PSF

SHOULD EVERYBODY LEARN TO PROGRAM?

SHOULD EVERY CHILD LEARN TO PROGRAM?

Yes!

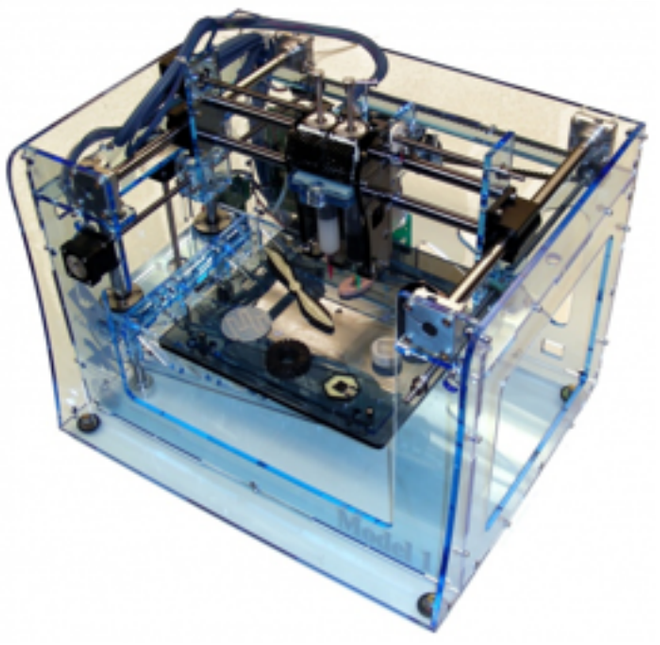
WHY?

- It improves cognitive skills
- In the next 10 years software development jobs are projected to increase 2x as other jobs (in average)
- We want kids to understand technology, not be simply consumers of technology
- Understanding science and technology is essential for the progress of society
- Computing technologies have given us unprecedented means to communicate and build social relations but can also be used to take away some of our rights (who owns you digital content?)

ANYTHING ELSE?

- It breaks the old capitalistic model which makes a distinction between the investors who buy the means of production, and the workers who build products
- Anyone can build software with modest "means of a production": a laptop, a good education, and time
- All software developers have the means to build their own companies.
- They make a product that has economic value, often social value, can be replicated at no cost, and can be discarded without polluting the environment
- Production of digital content (especially software) will continue to be an increasing part of the world' economy.

WHICH SOCIETY DO WE WANT?



DO WE TEACH PROGRAMMING WELL?

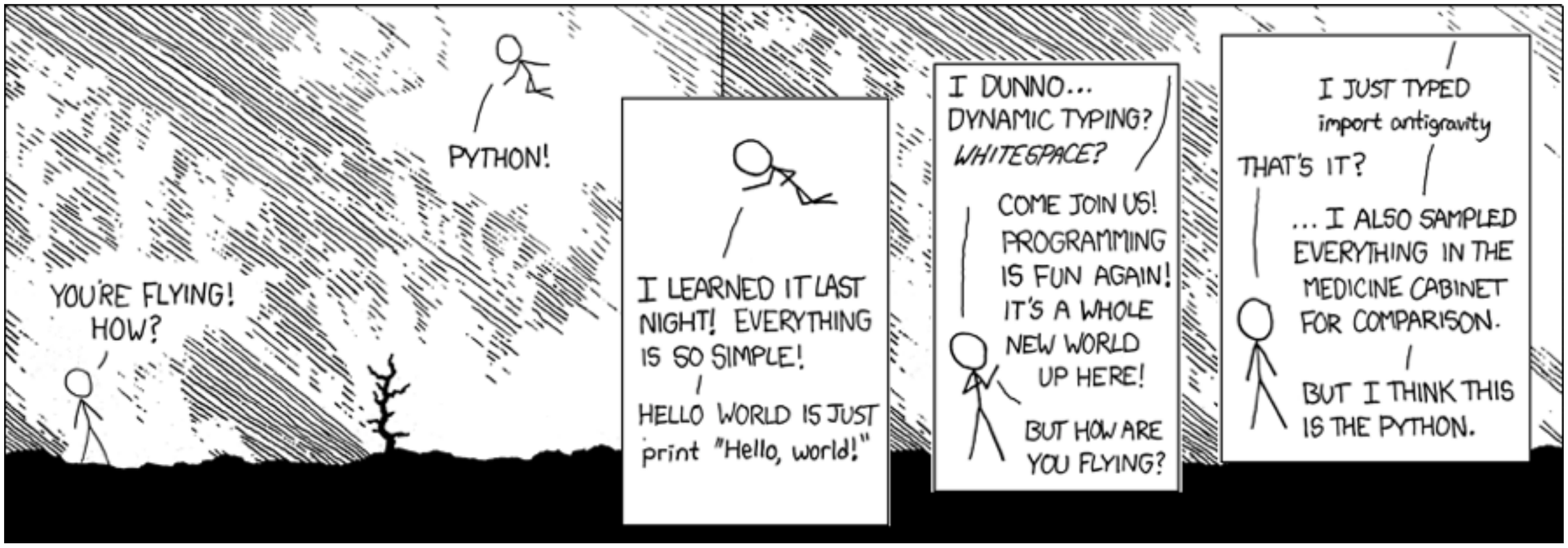

```
public class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello, World");
    }
}
```

NO!

- We put obstacles in the way of young programmer (shell, IDE, ...)
- We use languages that focus on syntax not semantic
- We use examples that do not leverage on students' knowledge
- We do not provide motivation (intro classes far from real life)
- Bottom-up approach instead of top-down approach

HOW CAN WE DO IT BETTER?

```
print “hello world”
```



HOW CAN WE DO IT EVEN BETTER?

```
def index():
    return “hello world”
```

Put in context: www!

- The web provides motivation for learning to program
- The web gives kids means to communicate
- Kids already associate the computer with the web

HOW CAN WE DO IT EVEN BETTER?

```
def index():
    return “hello world”
```

Put in context: www!

- The web provides motivation for learning to program
- The web gives kids means to communicate
- Kids already associate the computer with the web

- Who here has a kid who is not addicted to youtube?

my job is to make web development

easy
more accessible
less error prone
more secure
modular

my job is to make web development

easy
more accessible
less error prone
more secure
modular

Disclaimer: I do not claim any success. I am just trying....

WE2PY: BATTERIES INCLUDED

web server
ssl enabled

DAL + database
auto-migrations SQLite

web IDE
design, deploy, manage

html, xml, json, rss, ics, pdf, rtf,
xmlrpc, jsonrpc, soap,
ldap, pam, janrain, dropbox, google,
CAS, OpenID, oauth 1&2, x509
marmin, markdown,
google wallet, authorize.net, stripe.com
memcache, redis
twitter bootstrap

.zip

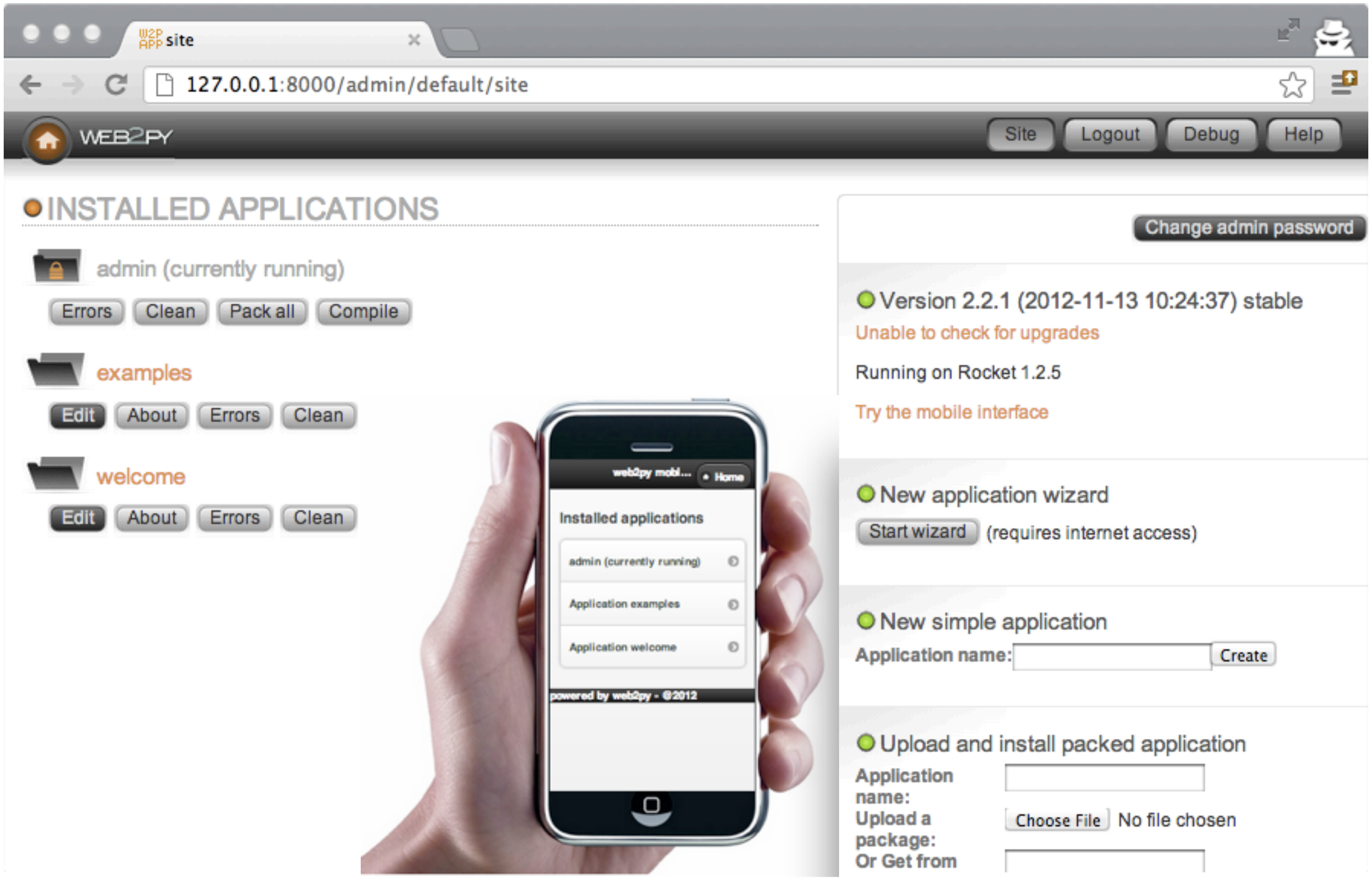
No installation. No configuration. Just Unzip and Click!

WEB2PY CONTRIBUTORS

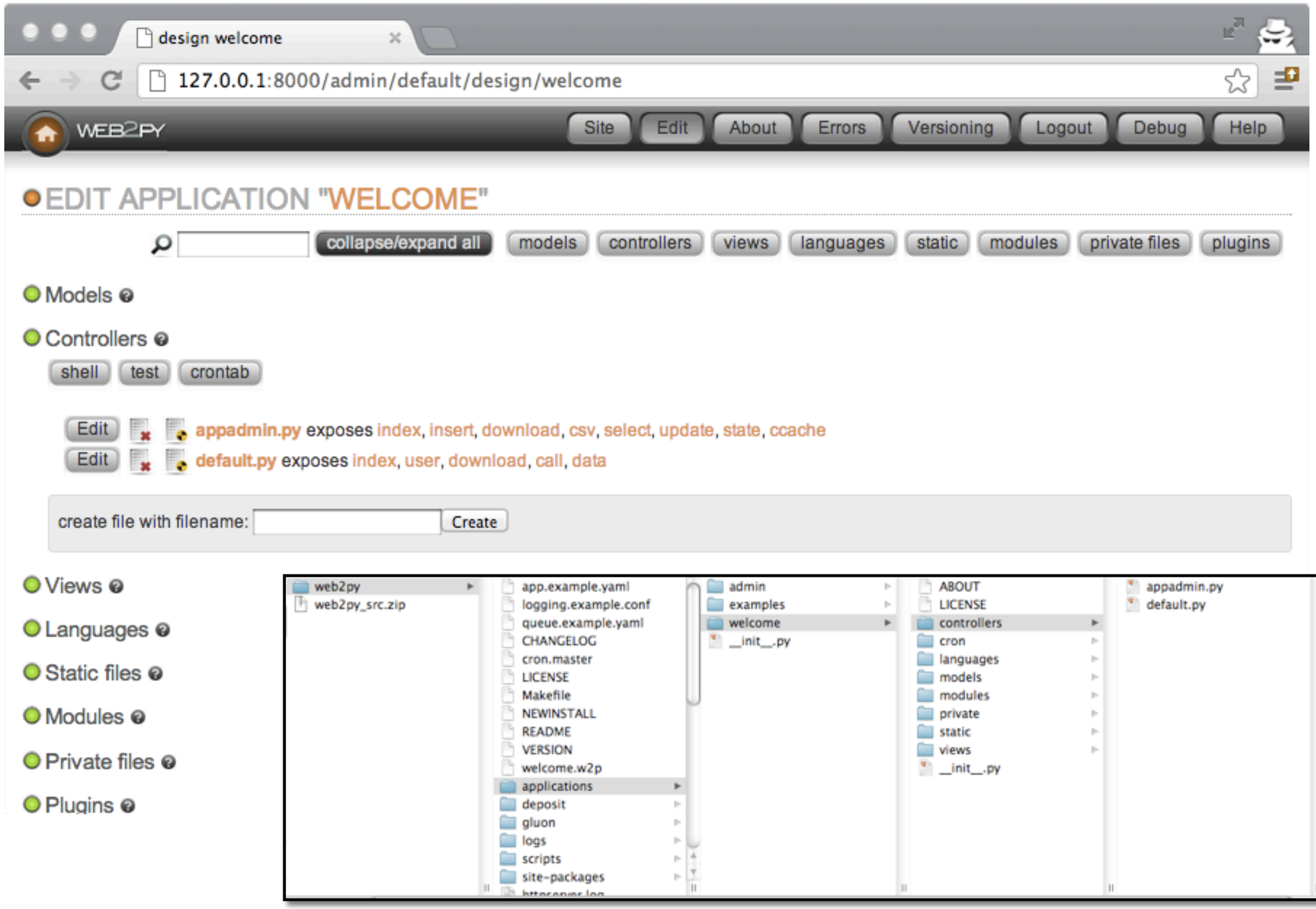
2011

2012

Web based IDE "admin" with hot plug and play of multiple apps



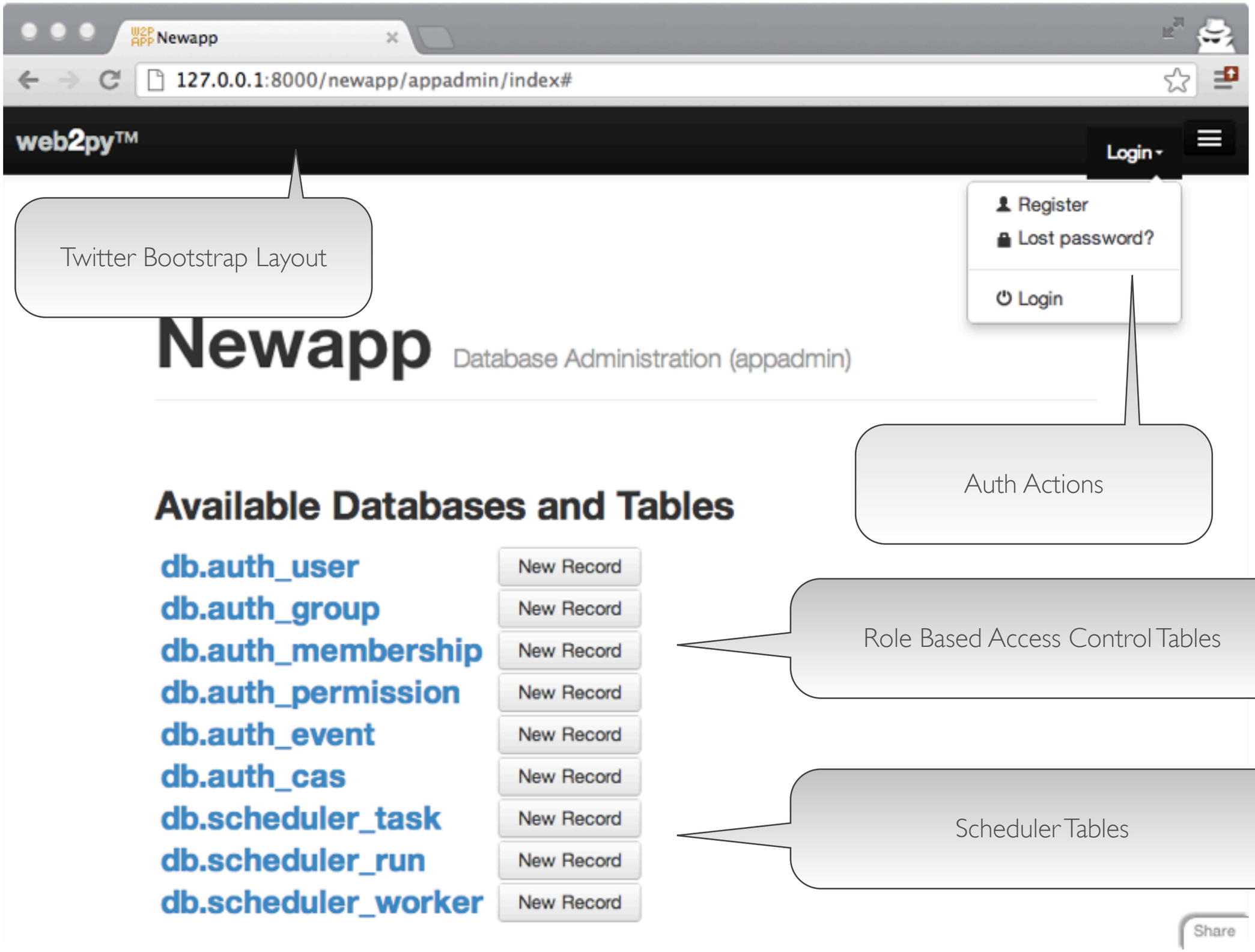
Thin-IDE: only shows file system, no metadata



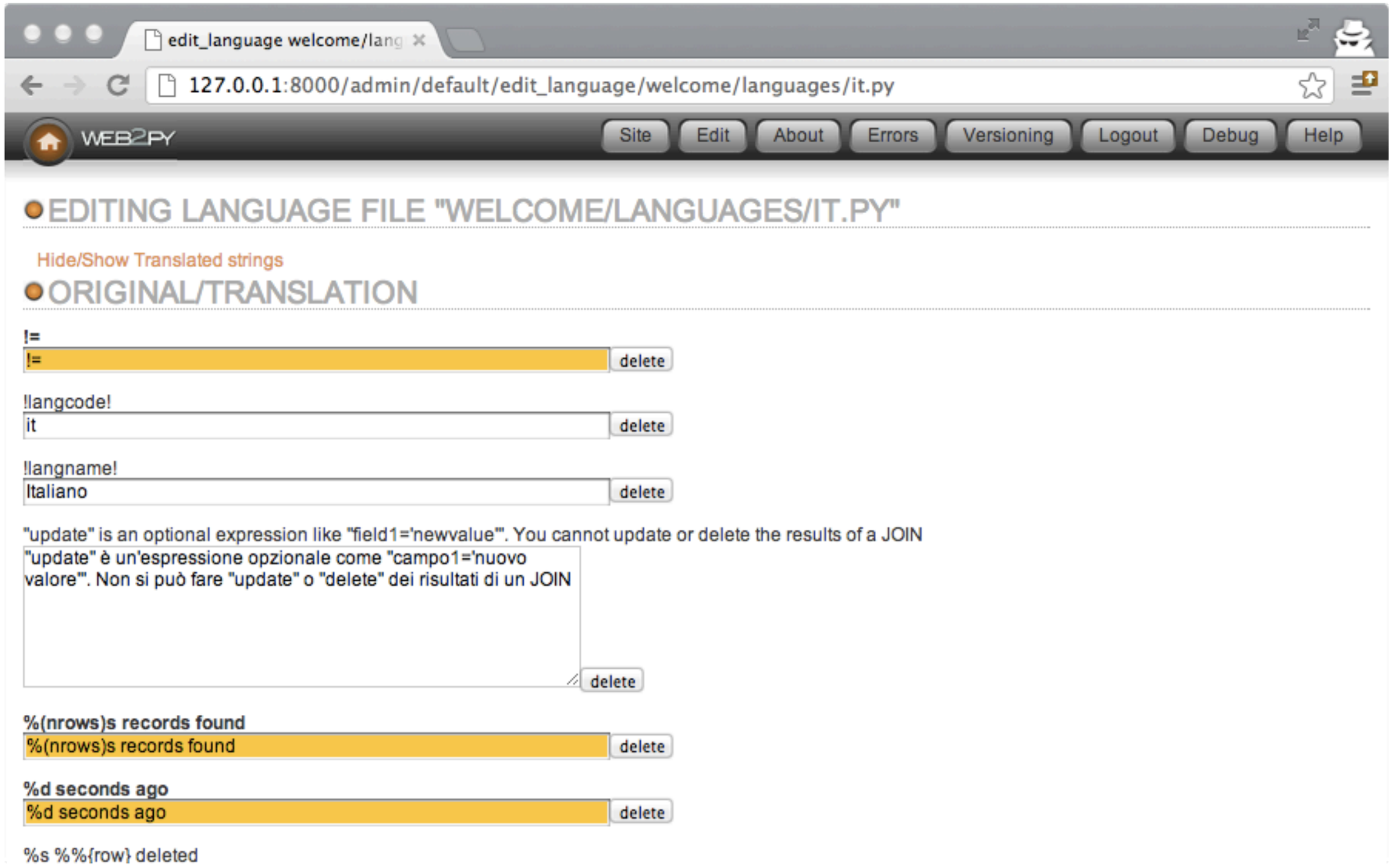
Web based editor (code-mirror)

Web based database administration (per app)

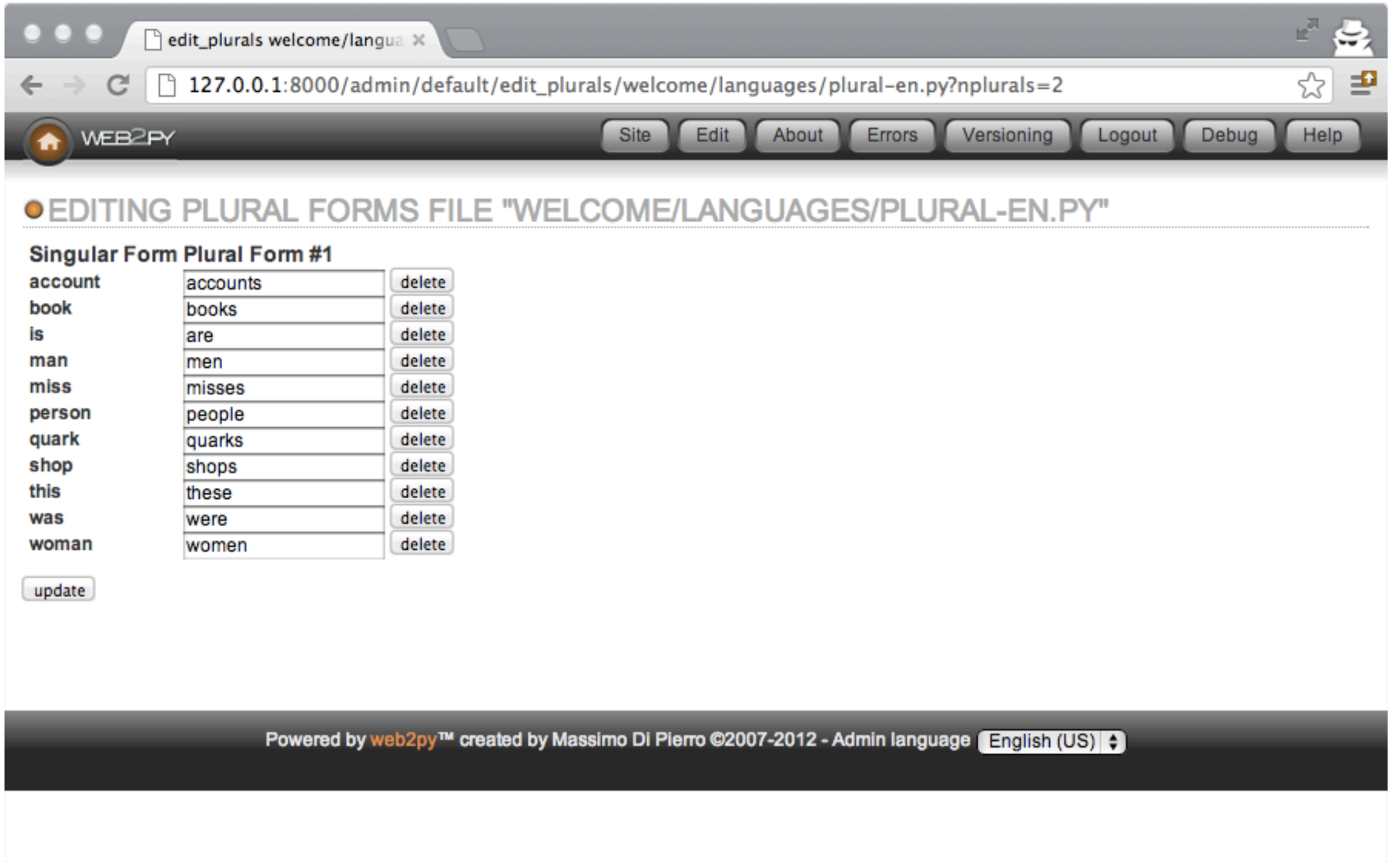
SQLite, MySQL, PotsgreSQL, MSSQL, Firebid, Oracle, DB2, Ingres, Informix, Ingres, Sybase, GAE, ...



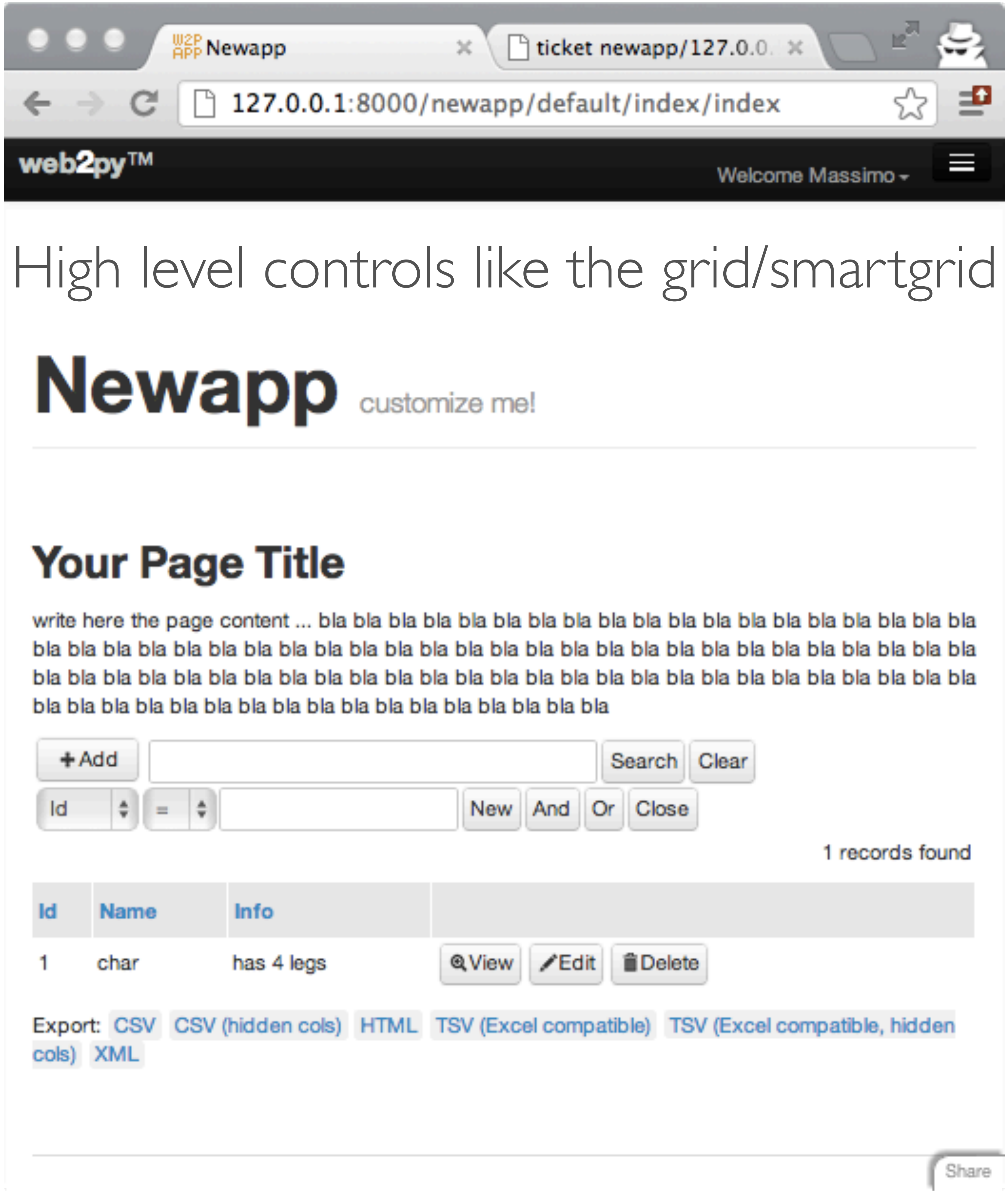
Web translation page for internationalization (per app)



Built-in pluralization system



High level controls like the grid/smartgrid



SYNTAX


```
public class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello, World");
    }
}
```

VS

```
print “hello world”
```

BOTTLE EXAMPLE

```
from bottle import run, route, get, static_file


@get('/index')
def index()
    return ‘hello world’

@route('/static/<filename>')
def server_static(filename):
    return static_file(filename, root='static')

run(host='localhost', port=8080)
```

required inputs

routing logic

action

handler for static files

start web server

FLASK EXAMPLE

```
from flask import Flask, request



app = Flask(__name__)
app.config.from_object(__name__)

@app.route('/index',methods=['GET'])
def index()
    return ‘hello world’

app.run(port=8080)
```

required inputs

boilerplate config logic

routing logic

action

start web server

TORNADO EXAMPLE

```
import tornado.ioloop
import tornado.web


def index(request):
    return 'hello world'

class MainHandler(tornado.web.RequestHandler):
    def get(self): return index(self.request)

application = tornado.web.Application([
   (r"/index", MainHandler),
   (r"/static/(.*)",tornado.web.StaticFileHandler,{"path": "static"})])

application.listen(8080)
tornado.ioloop.IOLoop.instance().start()
```

required inputs

action

routing logic

handler for static files

start web server

PYRAMID EXAMPLE

```
from wsgiref.simple_server import make_server
from pyramid.config import Configurator
from pyramid.response import Response
from pyramid.static import static_view

def index(context, request):
    return Response('hello world')

config = Configurator()
config.add_route('index', '/index')
config.add_view(index, route_name='index')
config.add_static_view(name='static', path='static')
app = config.make_wsgi_app()
server = make_server('0.0.0.0', 8080, app)
server.serve_forever()
```

required inputs

action

routing logic

handler for static files

start web server

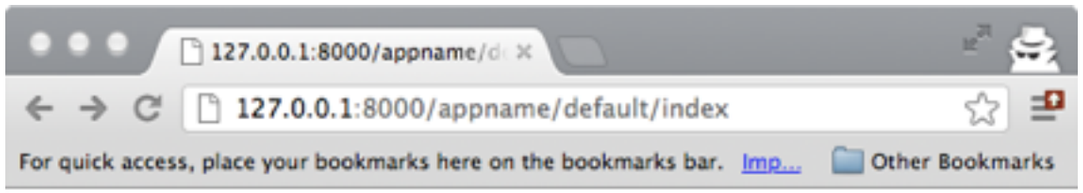
WEB2PY EXAMPLE

http://127.0.0.1:8000/appname/default/index

```
def index():
    return 'hello world'
```

call

action

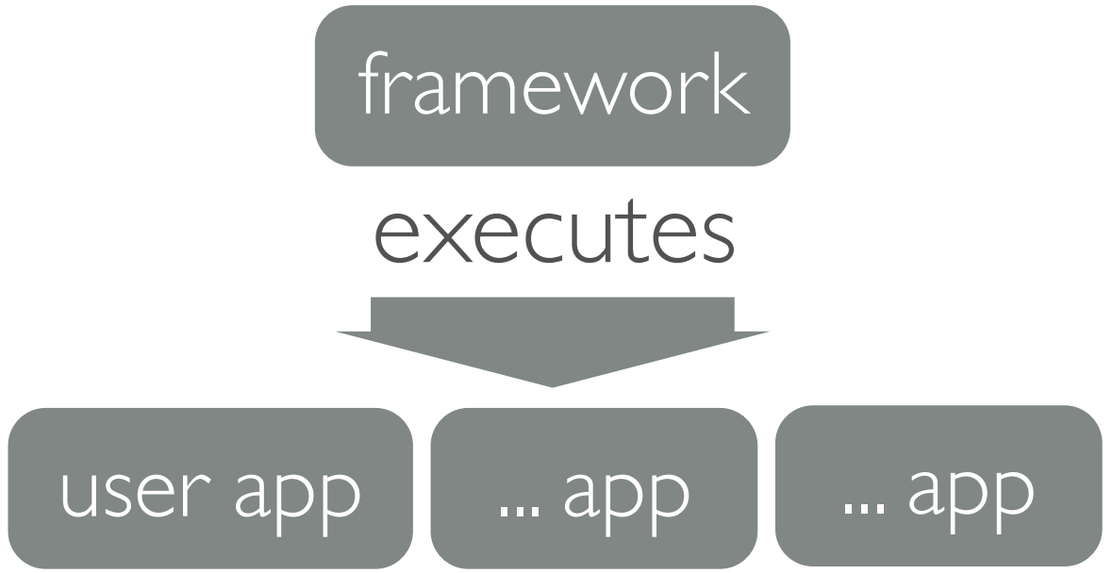


IMPORT VS EXEC

IMPORT VS EXEC


```
from bottle import ...
from flask import ...
from tornado import ...
from pyramid import ...
```

explicit better than implicit

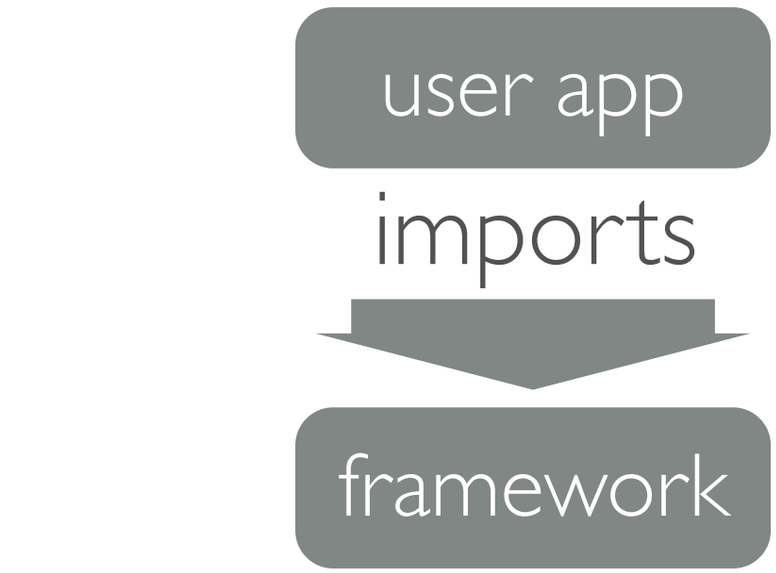


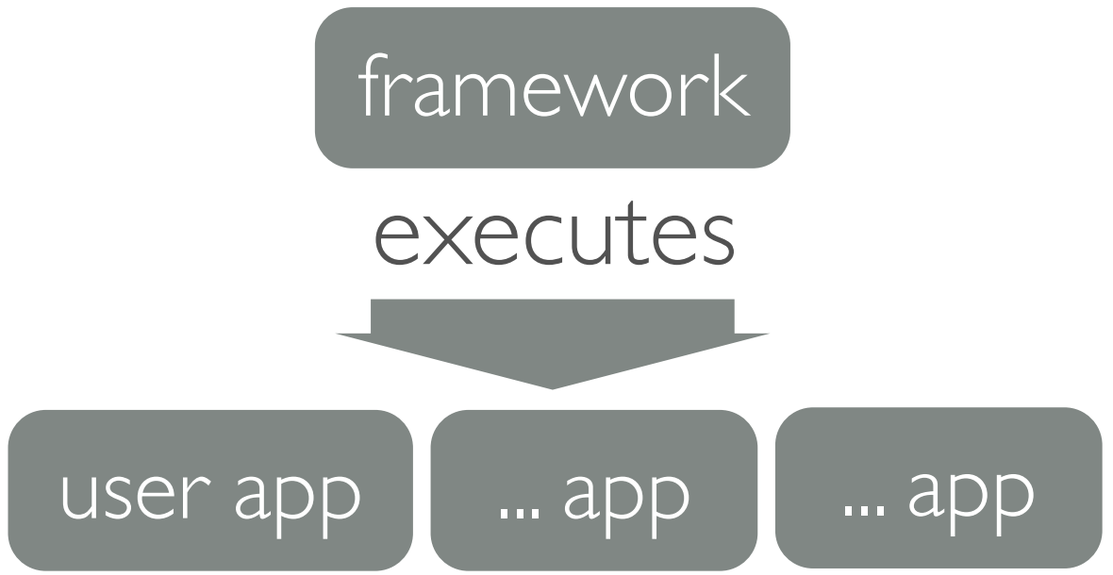

```
env = build_environment(request)
app = find_application(request)
exec app in env   (oversimplification)
```

do not repeat yourself

convention over configuration

IMPORT VS EXEC





faster (for simple apps)
more flexibility
no “magic”

less code (for simple apps)
how swap of code
multi app/multi project
homogeneous environment
“magic”

LAYERS OF CODE

SQL inside Python

(DAL or ORM)

HTML inside Python

(helpers)

CODE in HTML

(MVC)

JS in HTML

```
execute('select * from users where id=1')
```

```
db(db.users.id==1).select()
```

```
return '<div><h1>%s</h1></div>' % x
```

```
return DIV(H1(x))
```

```
<div>{{if x}}check{{endif}}</div>
```

```
<div>{{if x:}}check{{pass}}</div>
```

```
<div><script>alert('hi!')</script></div>
```

```
<div>{{=LOAD('action',ajax=True)}}</div>
```

WEB2PY DAL

- SQLite, MySQL, PotsgreSQL, MSSQL, Firebid, Oracle, DB2, Ingres, Informix, Ingres, Sybase, GAE, ...
- automatic migrations
- multiple dbs, connection pooling, Round Robin redundancy, distributed transactions
- joins, left joins, aggregates, nested selects, recursive selects

```
db = DAL('sqlite://storage.sqlite')
db.define_table('person',Field('name'))
db.define_table('thing',Field('name'),Field('owner',db.person))
db.thing.insert(name='mac',owner=db.person.insert(name='Max'))

ownership = db.person.id == db.thing.owner
n_things = db.thing.id.count()
for row in db(ownership).select(db.person.name, n_things,
                                groupby=db.person.id):
    print row.person.name, row(n_things)
```

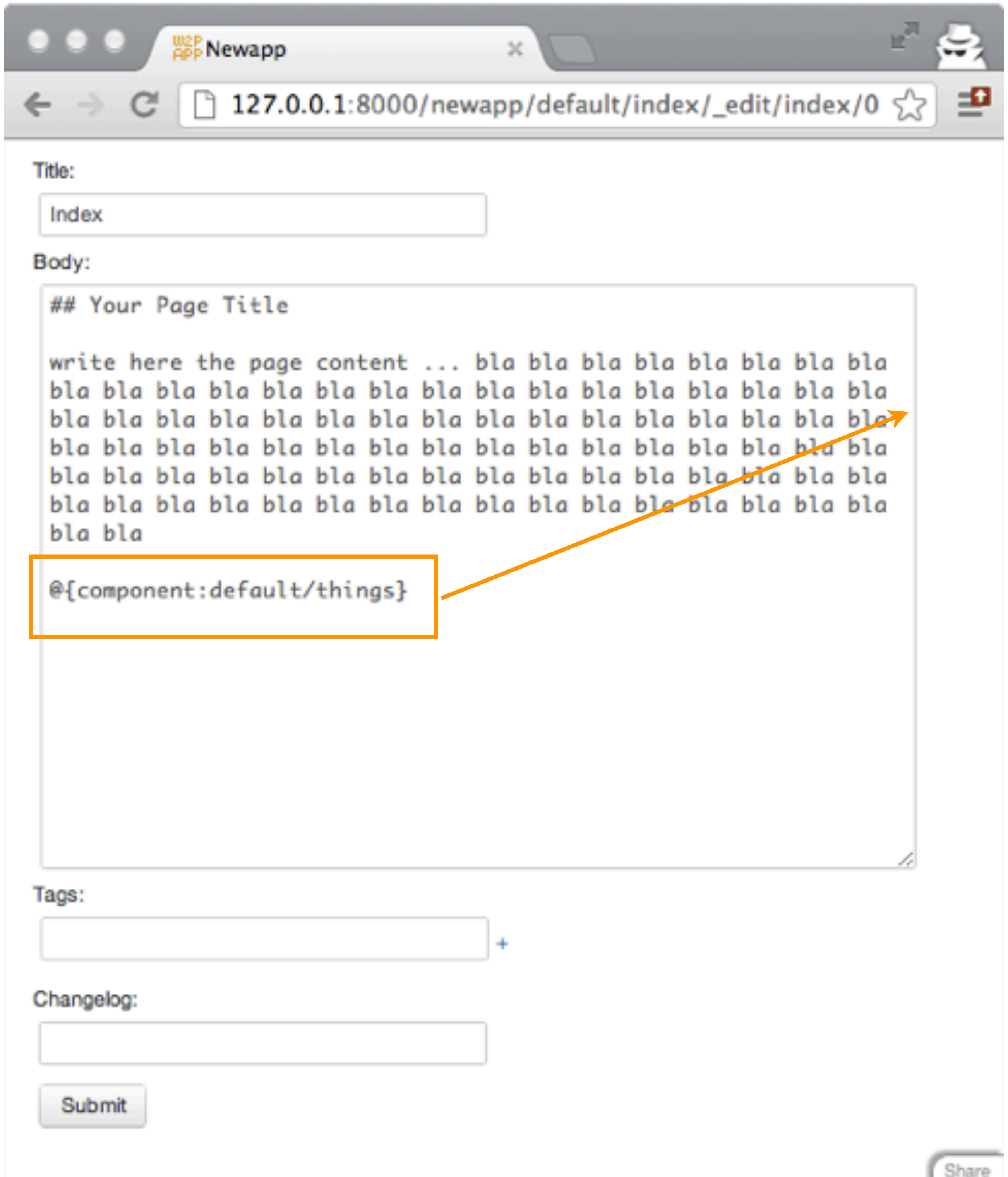
PROGRAMMING AS WIKI

```
# models/db.py
db.define_table('thing',
    Field('name'),
    Field('info','test'))

# controllers/default.py
def index():
    return auth.wiki()

def things():
    return SQLFORM.grid(db.thing)
```

PROGRAMMING AS WIKI



CONCLUSIONS

- The elitist approach to programming leads us to the wrong path
- There is not only one solution
- There is not only one web framework
- We need to learn from each other
- We need to build a society where technology is controller by people, not by large corporation
- We need to build tools that are easier to user and effective and allow more people to use technology for public good