Things you didn't know about Python

a presentation by Armin Ronacher
for PyCon South Africa 2012

@mitsuhiko
http://lucumr.pocoo.org/

Things you might already know about computers

a presentation by Armin Ronacher
for PyCon South Africa 2012

@mitsuhiko
http://lucumr.pocoo.org/

Things you did[...]

Things you might already know about computers

and the world!!!11

a presentation by Armin Ronacher

th Africa 2012

@mitsuhiko
http://lucumr.pocoo.org/

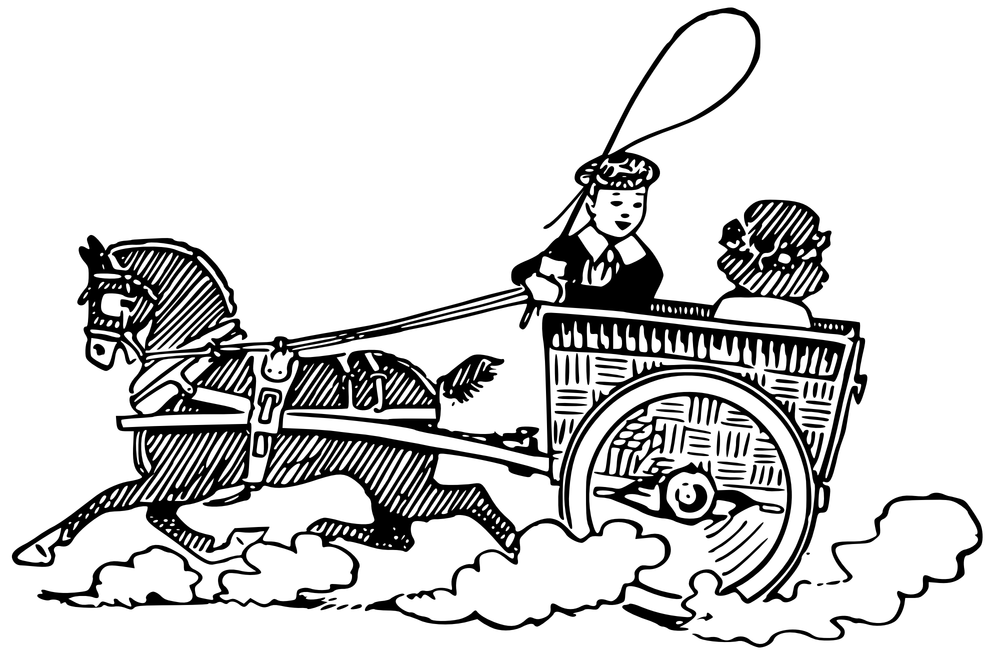
Everything is horrible
and nobody cares

We're doomed

[Untitled]

Motivation for this Talk

Who am I and why this talk



Armin Ronacher

Software Engineer at Fireteam
Game Middleware Provider
@mitsuhiko / @fireteamltd

We're using Python

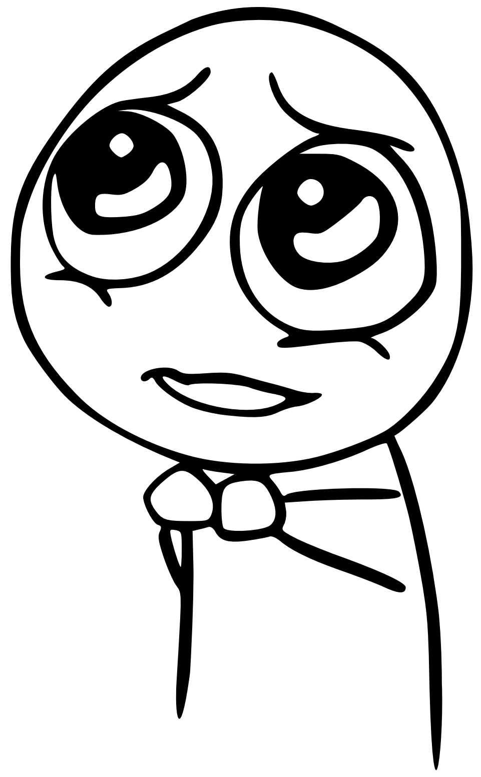
And not just us.

Python has been popular in parts of in the gaming industry

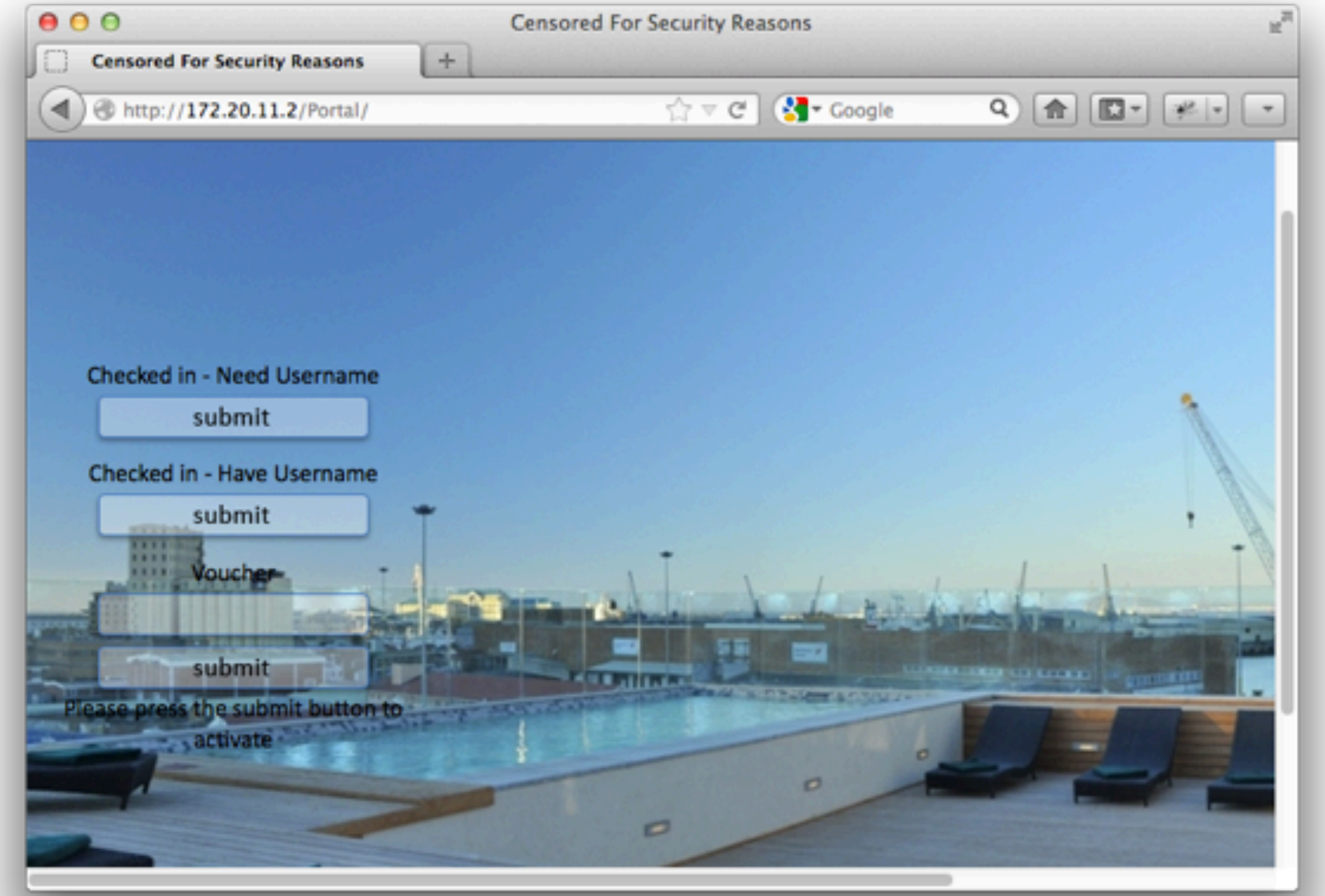
I'm also doing Python Libraries

and help people online using them.

What we can learn from Wifi Hotspots



Starting Somewhere


Censored For Security Reasons

http://172.20.11.2/Portal/

Checked in - Need Username

submit

Checked in - Have Username

submit

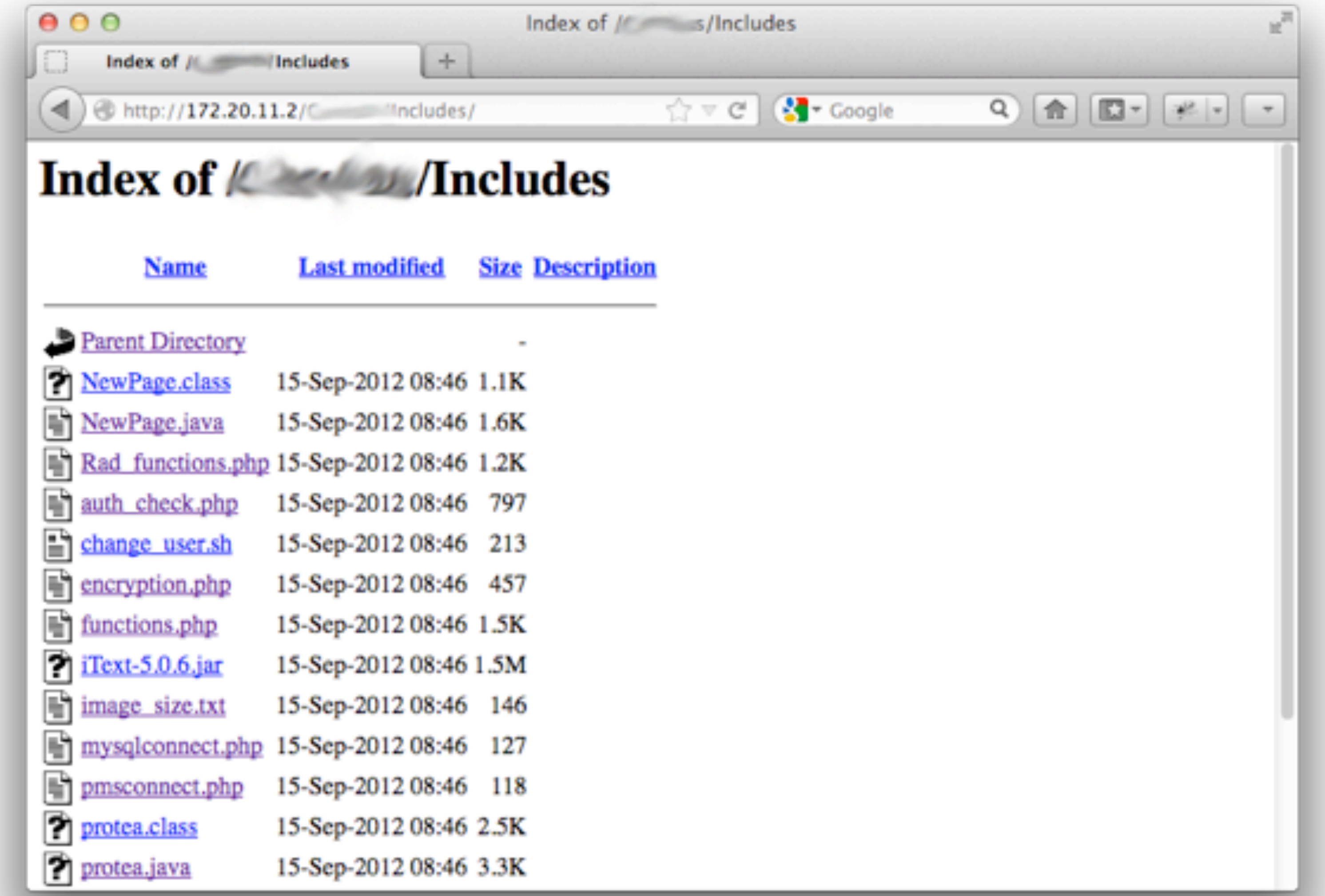
Voucher

submit

Please press the submit button to activate

Intended Login Mask

Down the Rabbit Hole



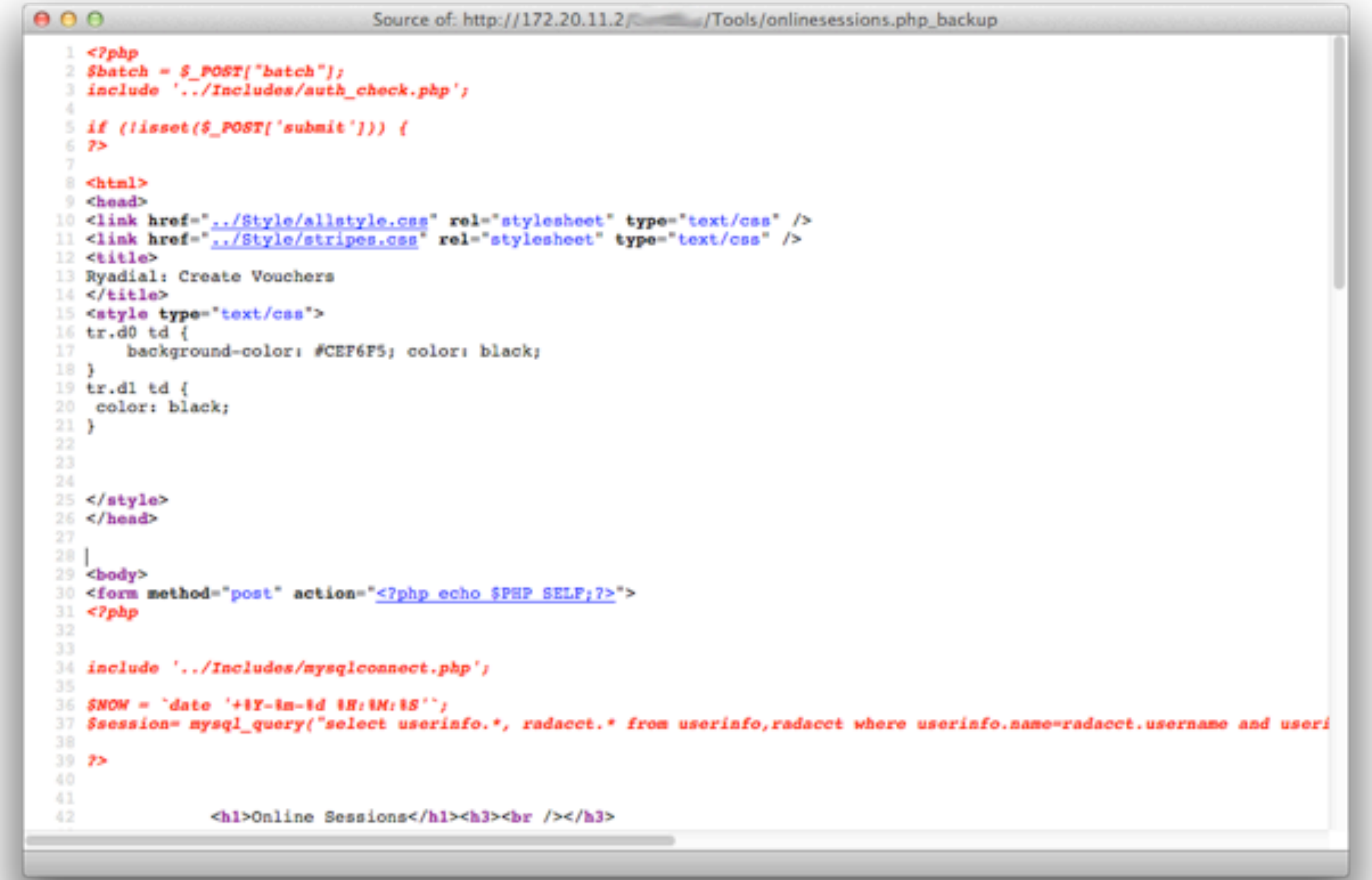
Index of /[redacted]s/Includes

http://172.20.11.2/[redacted]Includes/

```
Index of /[redacted]/Includes

      Name                 Last modified      Size  Description
  ----------------------------------------------------------------
  Parent Directory                              -
  NewPage.class        15-Sep-2012 08:46  1.1K
  NewPage.java         15-Sep-2012 08:46  1.6K
  Rad_functions.php    15-Sep-2012 08:46  1.2K
  auth_check.php       15-Sep-2012 08:46   797
  change_user.sh       15-Sep-2012 08:46   213
  encryption.php       15-Sep-2012 08:46   457
  functions.php        15-Sep-2012 08:46  1.5K
  iText-5.0.6.jar      15-Sep-2012 08:46  1.5M
  image_size.txt       15-Sep-2012 08:46   146
  mysqlconnect.php     15-Sep-2012 08:46   127
  pmsconnect.php       15-Sep-2012 08:46   118
  protea.class         15-Sep-2012 08:46  2.5K
  protea.java          15-Sep-2012 08:46  3.3K
```

Served by Apache, PHP 5.3, Directory Listings

*.php_backup



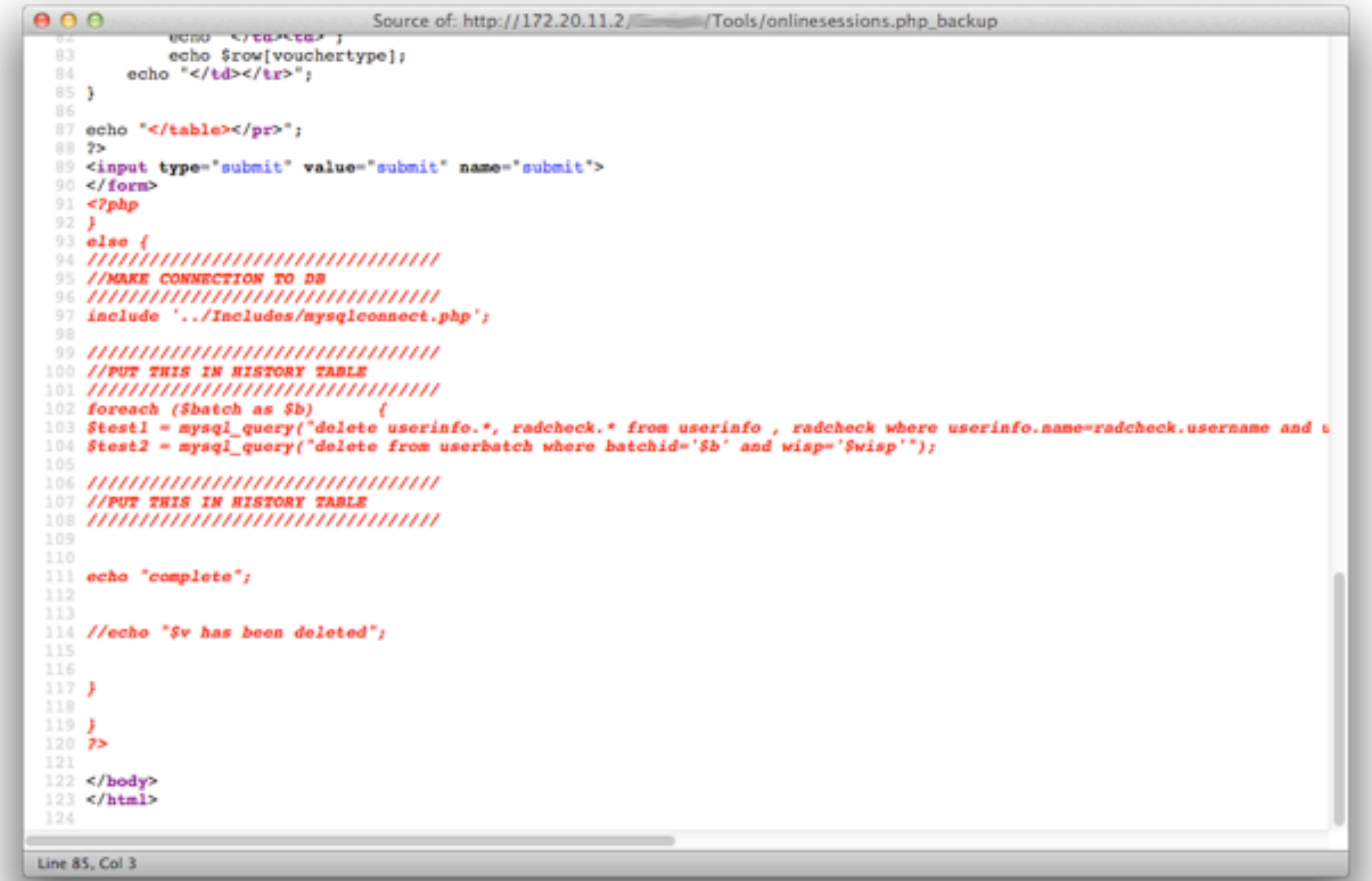
Source of: http://172.20.11.2/[redacted]/Tools/onlinesessions.php_backup

```
<?php
$batch = $_POST["batch"];
include '../Includes/auth_check.php';

if (!isset($_POST['submit'])) {
?>

<html>
<head>
<link href="../Style/allstyle.css" rel="stylesheet" type="text/css" />
<link href="../Style/stripes.css" rel="stylesheet" type="text/css" />
<title>
Ryadial: Create Vouchers
</title>
<style type="text/css">
tr.d0 td {
    background-color: #CEF6F5; color: black;
}
tr.d1 td {
 color: black;
}



</style>
</head>


<body>
<form method="post" action="<?php echo $PHP_SELF;?>">
<?php


include '../Includes/mysqlconnect.php';

$NOW = `date '+%Y-%m-%d %H:%M:%S'`;
$session= mysql_query("select userinfo.*, radacct.* from userinfo,radacct where userinfo.name=radacct.username and useri

?>


            <h1>Online Sessions</h1><h3><br /></h3>
```

source code? Check!

SQL Injection? Check

Further …

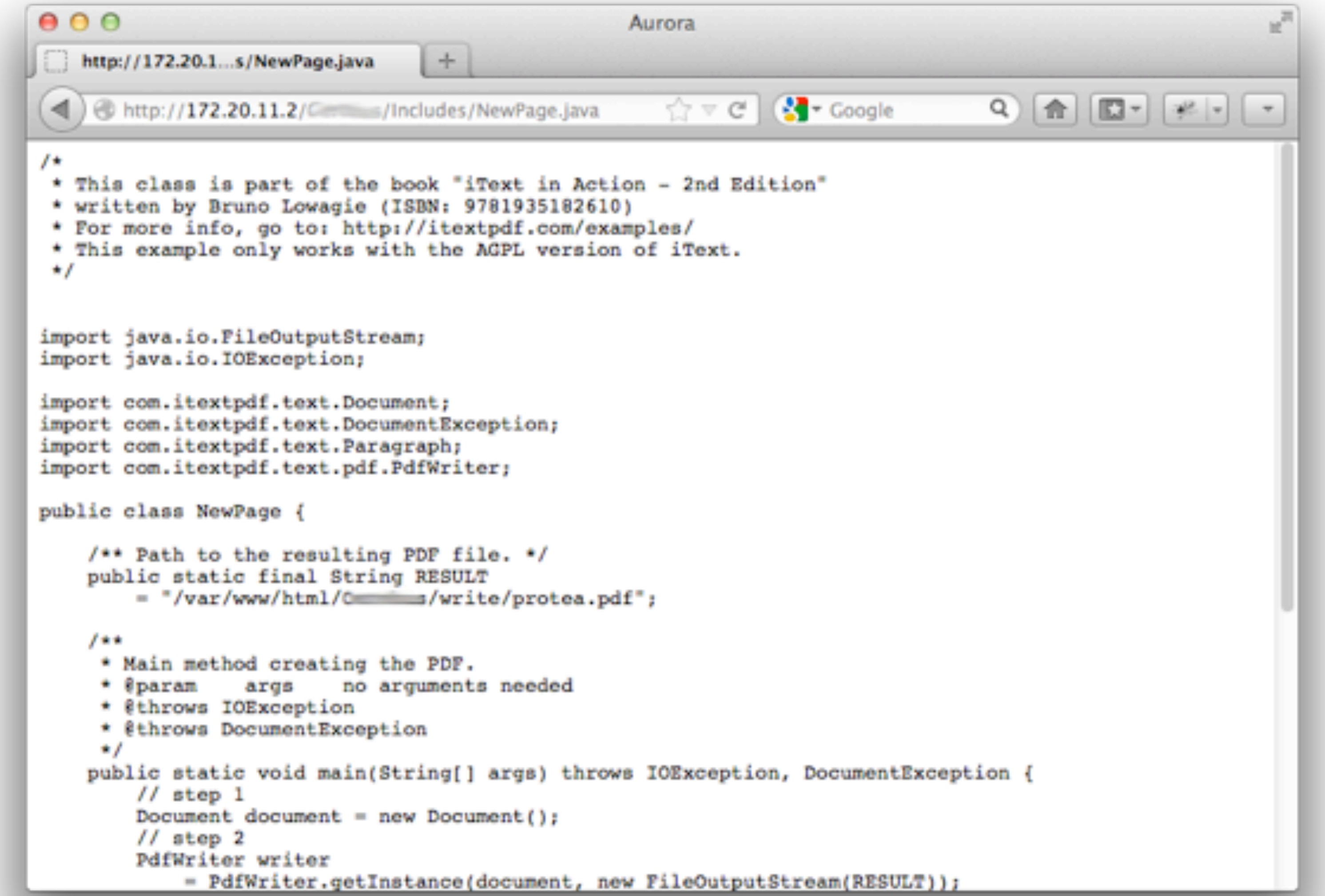

Source of: http://172.20.11.2/[redacted]/Tools/onlinesessions.php_backup

```
        echo "</td><td>";
        echo $row[vouchertype];
    echo "</td></tr>";
}

echo "</table></pr>";
?>
<input type="submit" value="submit" name="submit">
</form>
<?php
}
else {
//////////////////////////////////
//MAKE CONNECTION TO DB
//////////////////////////////////
include '../Includes/mysqlconnect.php';

//////////////////////////////////
//PUT THIS IN HISTORY TABLE
//////////////////////////////////
foreach ($batch as $b)      {
$test1 = mysql_query("delete userinfo.*, radcheck.* from userinfo , radcheck where userinfo.name=radcheck.username and u
$test2 = mysql_query("delete from userbatch where batchid='$b' and wisp='$wisp'");

//////////////////////////////////
//PUT THIS IN HISTORY TABLE
//////////////////////////////////


echo "complete";


//echo "$v has been deleted";


}

}
?>

</body>
</html>
```

Line 85, Col 3

Register Globals? Check
Debug Comments? Check

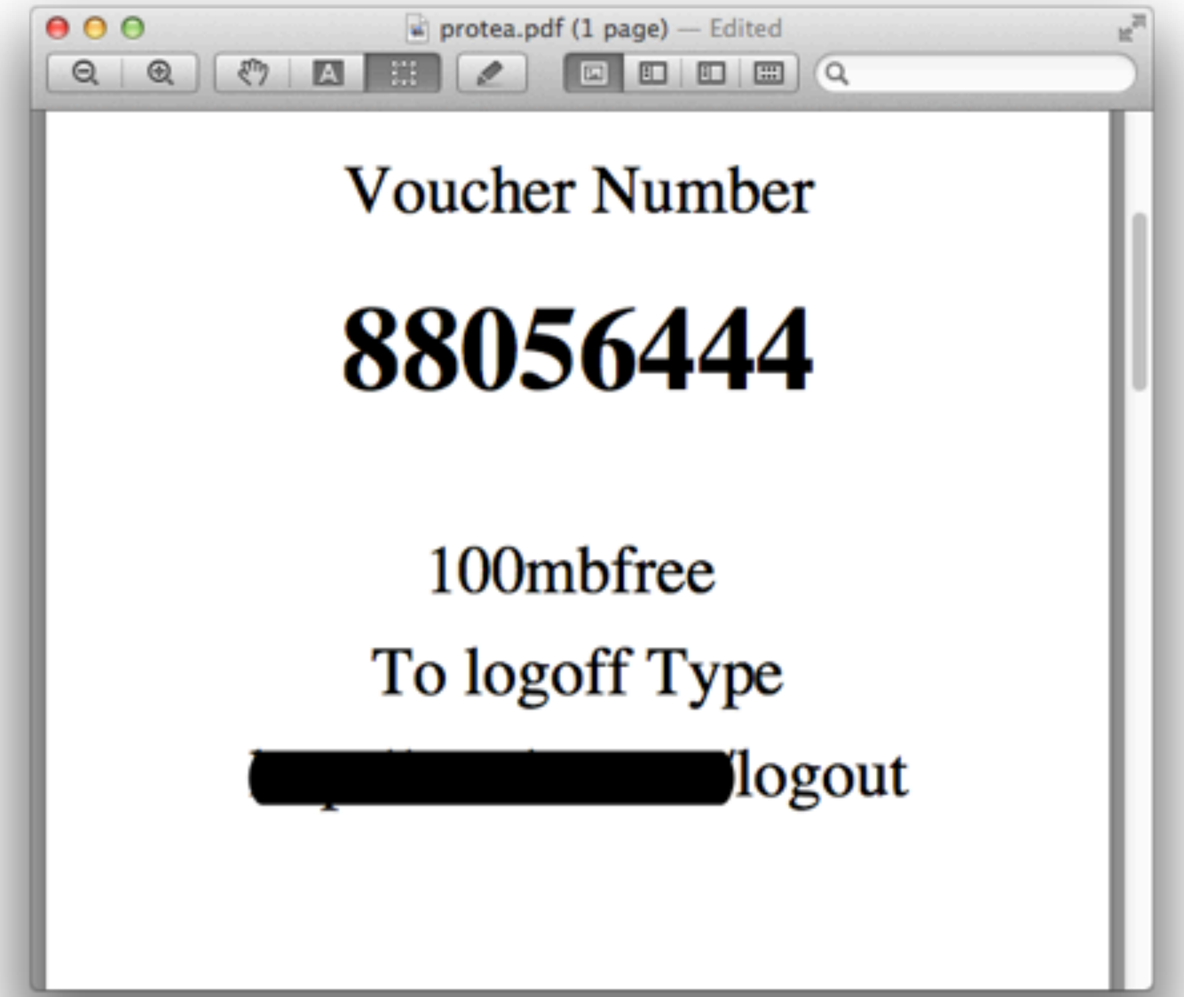
And Further


Aurora
http://172.20.1...s/NewPage.java
http://172.20.11.2/[redacted]/Includes/NewPage.java
Google

```
/*
 * This class is part of the book "iText in Action - 2nd Edition"
 * written by Bruno Lowagie (ISBN: 9781935182610)
 * For more info, go to: http://itextpdf.com/examples/
 * This example only works with the AGPL version of iText.
 */


import java.io.FileOutputStream;
import java.io.IOException;

import com.itextpdf.text.Document;
import com.itextpdf.text.DocumentException;
import com.itextpdf.text.Paragraph;
import com.itextpdf.text.pdf.PdfWriter;

public class NewPage {

    /** Path to the resulting PDF file. */
    public static final String RESULT
        = "/var/www/html/[redacted]/write/protea.pdf";

    /**
     * Main method creating the PDF.
     * @param    args    no arguments needed
     * @throws IOException
     * @throws DocumentException
     */
    public static void main(String[] args) throws IOException, DocumentException {
        // step 1
        Document document = new Document();
        // step 2
        PdfWriter writer
            = PdfWriter.getInstance(document, new FileOutputStream(RESULT));
```

GPL Violation? Check

Yay!



Voucher Number

88056444

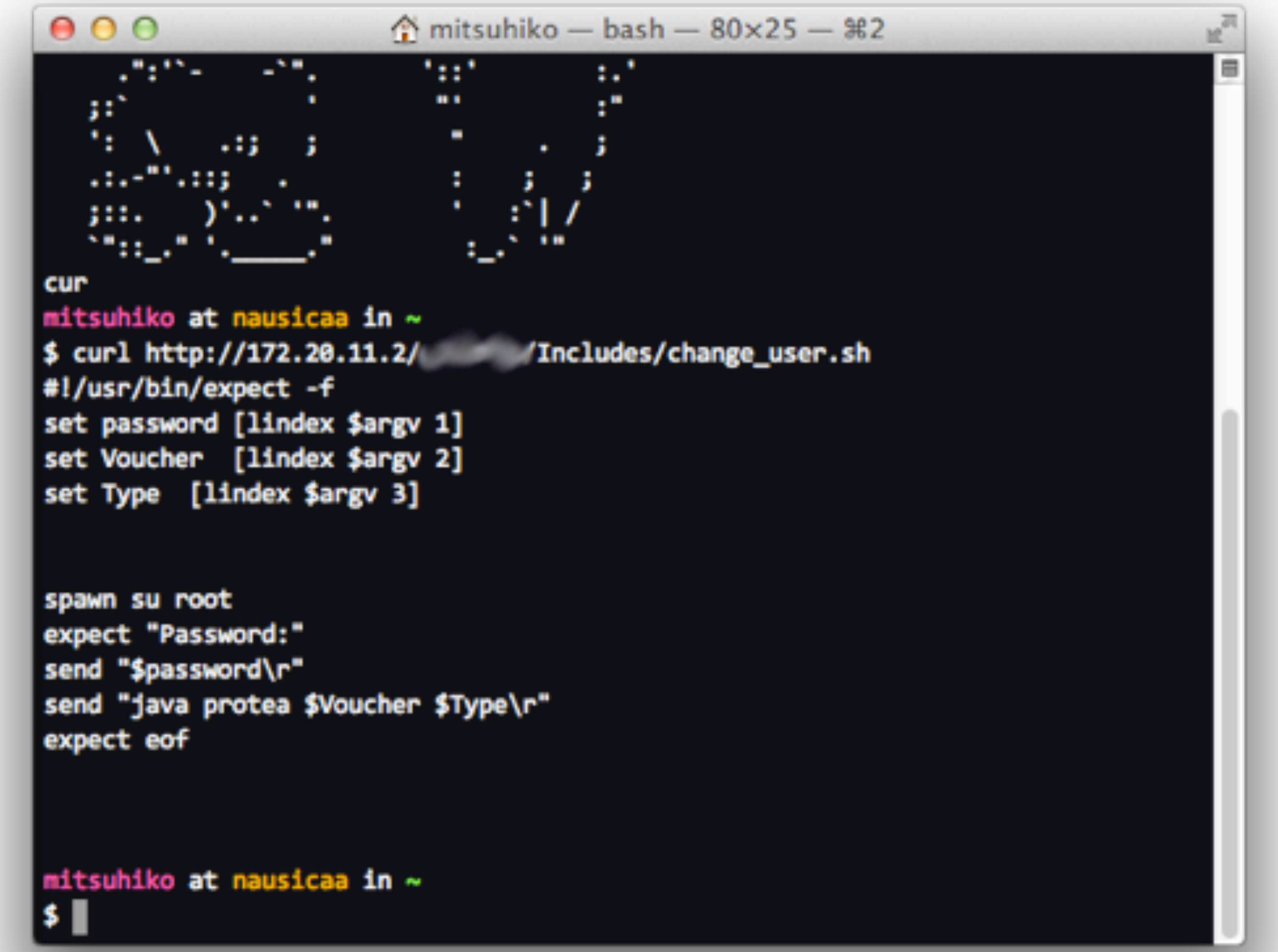
100mbfree

To logoff Type

[redacted]logout

Pre generated voucher PDF? Check

To Round it all Up

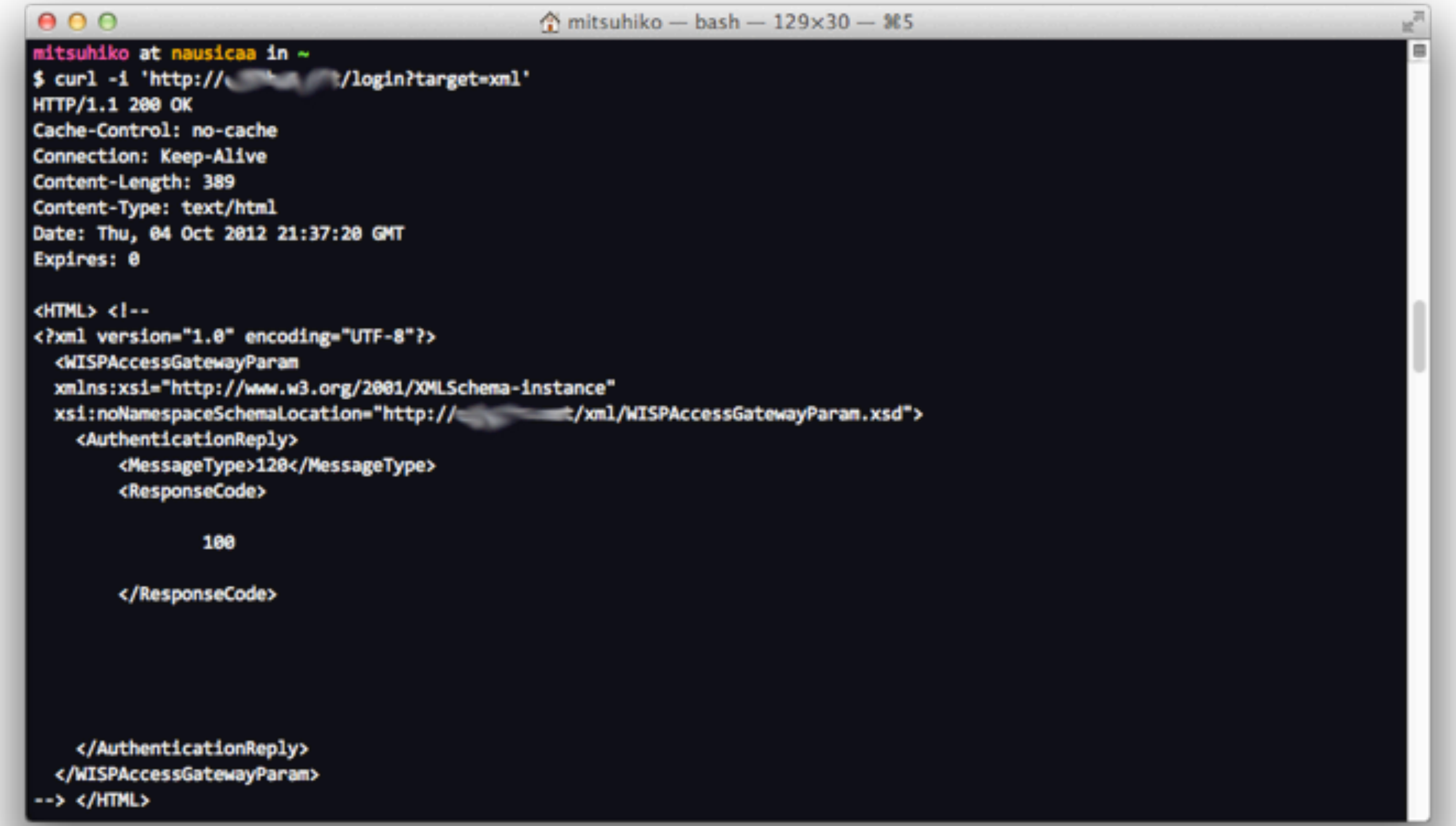


mitsuhiko — bash — 80×25 — ⌘2

```
     ."':'`-    -`".       ':'       :.'
   ;:`              '      ":        :"
   ': \    .:;   ;          "      .   ;
   .:.-"'.::;    .          :    ;   ;
   ;::.    )'..` '".        '   :`| /
   `"::_." '______."          :_.` '"
cur
mitsuhiko at nausicaa in ~
$ curl http://172.20.11.2/[redacted]/Includes/change_user.sh
#!/usr/bin/expect -f
set password [lindex $argv 1]
set Voucher  [lindex $argv 2]
set Type  [lindex $argv 3]


spawn su root
expect "Password:"
send "$password\r"
send "java protea $Voucher $Type\r"
expect eof



mitsuhiko at nausicaa in ~
$ 
```

Comes with Instructions

Priorities

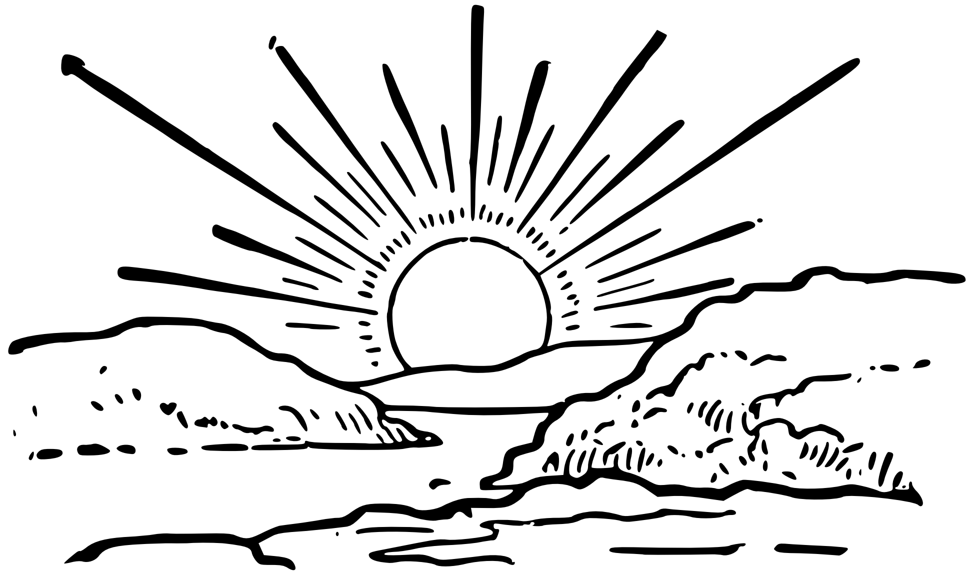


```
mitsuhiko at nausicaa in ~
$ curl -i 'http://[redacted]/login?target=xml'
HTTP/1.1 200 OK
Cache-Control: no-cache
Connection: Keep-Alive
Content-Length: 389
Content-Type: text/html
Date: Thu, 04 Oct 2012 21:37:20 GMT
Expires: 0

<HTML> <!--
<?xml version="1.0" encoding="UTF-8"?>
  <WISPAccessGatewayParam
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="http://[redacted]/xml/WISPAccessGatewayParam.xsd">
    <AuthenticationReply>
        <MessageType>120</MessageType>
        <ResponseCode>

                100

        </ResponseCode>





    </AuthenticationReply>
  </WISPAccessGatewayParam>
--> </HTML>
```

It's not secure if it does not have XML

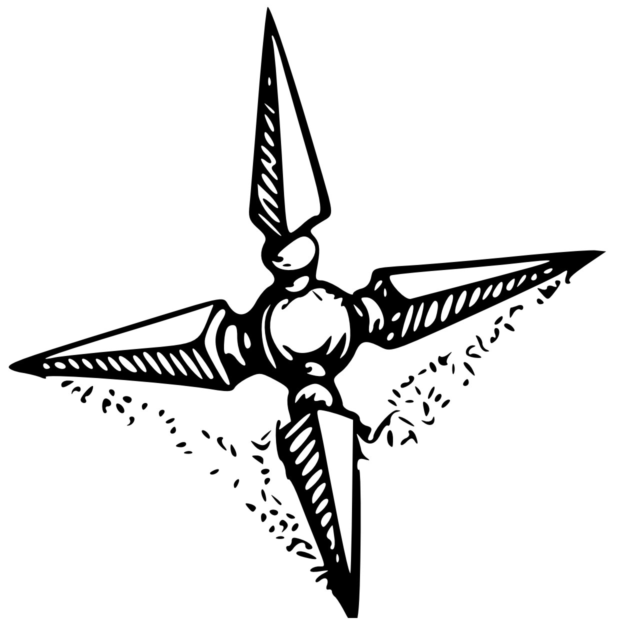
A Step Back

What do Wifi Hotspots have to do with anything?



Python is not perfect

… but the criticism is very high level

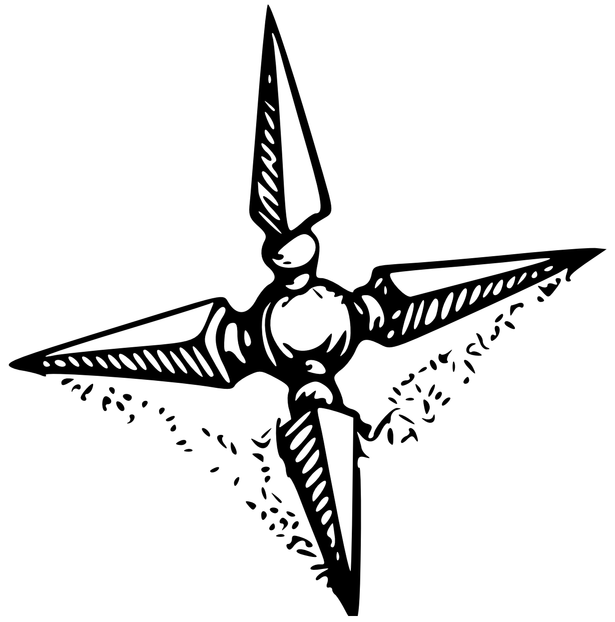


“First World Problems”

Most of our problems with Python are not stopping us from using the language. It just might make it less pleasant.

Who is using Python?

Let's start with the marketing bits



Big Players

NASA, Disney, Youtube, Google, etc.

Trivia: Microsoft shipped Python in 96

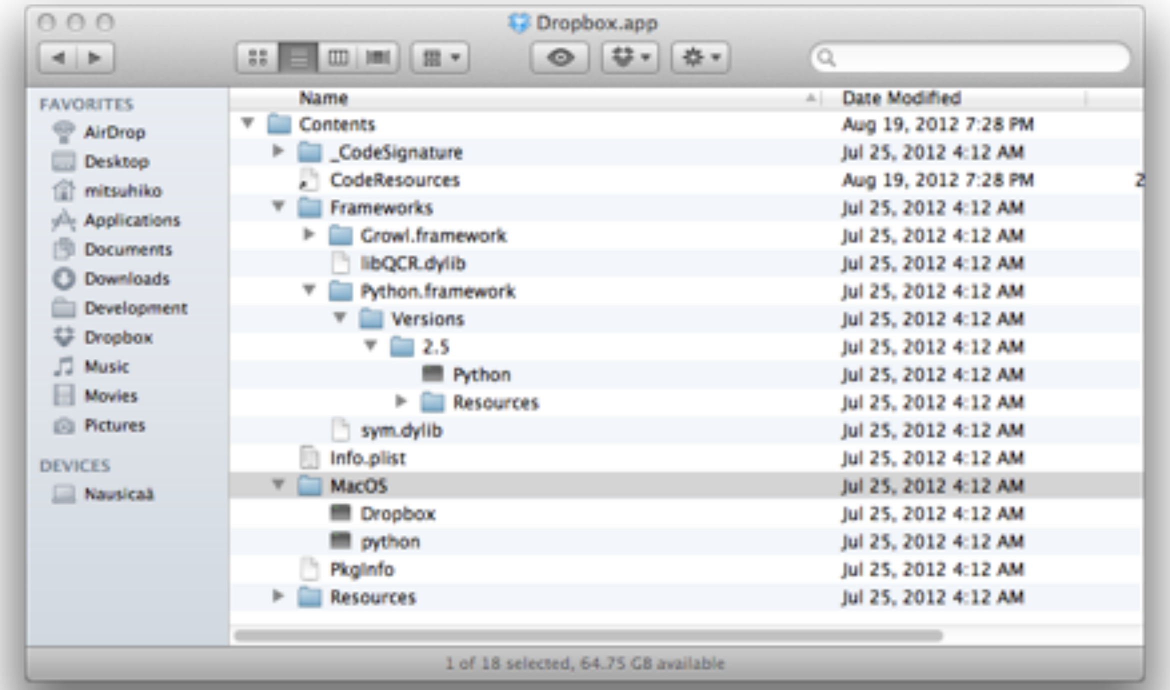
Microsoft Merchant Server was written in Python in 1996

But really everybody

Python is one of the things that just shows up.
If for nothing else, then build scripts.

Trivia: Dropbox uses Python

Not just on the server, the client is also implemented in Python!



```
Name                                Date Modified
▼ Contents                          Aug 19, 2012 7:28 PM
    ▶ _CodeSignature                Jul 25, 2012 4:12 AM
      CodeResources                 Aug 19, 2012 7:28 PM
    ▼ Frameworks                    Jul 25, 2012 4:12 AM
        ▶ Growl.framework           Jul 25, 2012 4:12 AM
          libQCR.dylib              Jul 25, 2012 4:12 AM
        ▼ Python.framework          Jul 25, 2012 4:12 AM
            ▼ Versions              Jul 25, 2012 4:12 AM
                ▼ 2.5               Jul 25, 2012 4:12 AM
                      Python        Jul 25, 2012 4:12 AM
                    ▶ Resources     Jul 25, 2012 4:12 AM
          sym.dylib                 Jul 25, 2012 4:12 AM
      Info.plist                    Jul 25, 2012 4:12 AM
    ▼ MacOS                         Jul 25, 2012 4:12 AM
          Dropbox                   Jul 25, 2012 4:12 AM
          python                    Jul 25, 2012 4:12 AM
      PkgInfo                       Jul 25, 2012 4:12 AM
    ▶ Resources                     Jul 25, 2012 4:12 AM

1 of 18 selected, 64.75 GB available
```

Gaming uses Python

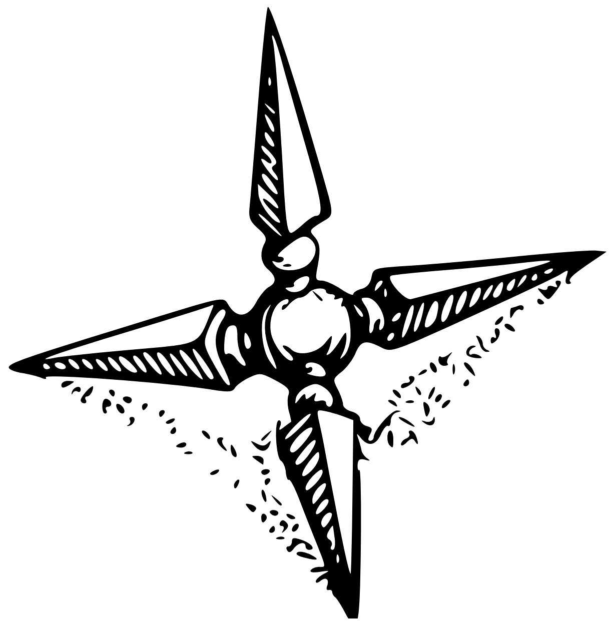
Demonware, Agora, EA/ESN, Fireteam

Nobody got fired for choosing Python

We are the IBM of Dynamic Languages

History Lessons

A few interesting bits about the past



1991: Where all started

- Exceptions
- Multiple Inheritance
- C inspired IO system

Trivia: string.py was horrible

It had $O(n^2)$ upper/lower functions

Trivia: what did this do until 2.5?

```
raise ((a, b), c), d
```

answer: raises exception a with value d

Trivia: mutex.py

a module called mutex
not actually a mutex
survived until 2.7

1995: The Huge Jump to 1.5

- Regular Expressions
- Exceptions as classes
- Built-in package support
- Embeddable

Trivia: did you know re is 50% python?

the regular expression compiler is written in Python

You notice that when you **`python -mtrace`**

Trivia: why are builtin types lowercase?

because they used to be functions

the types where in types.py

types.StringType was the type of a string (camelcase)

2000: Modern Python: Python 2.0

- Unicode Support
- Augmented assignments (+= etc.)
- Garbage Collector
- PEPs

2004: Python as you know it

- File encoding cookies
- Boolean types
- sets
- reverse iteration
- generator expressions

Trivia: 2.2.1 introduced True and False

… but no boolean type.

2.2.0: no true/false

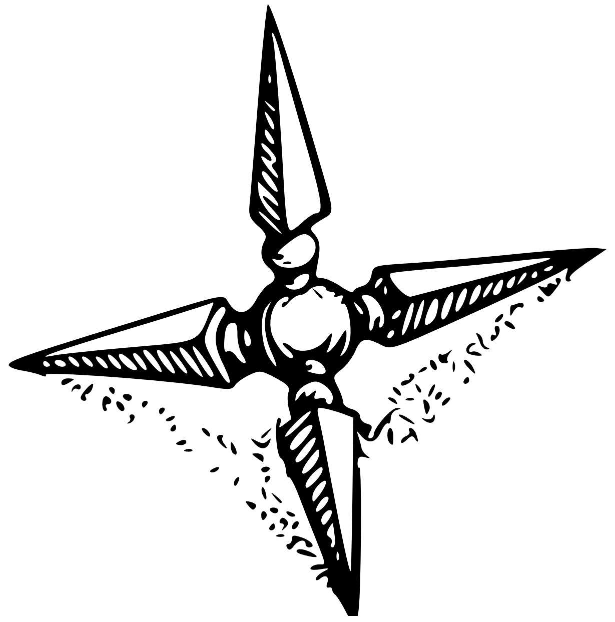
2.3.0: real boolean type

Today: Evolving Language

- PyPy
- Python 3

Reasons for Popularity

key adopters and killer-apps



Really Early Adopters

Math and Scientific Community

Python's operator overloading and simple syntax was very convenient for scientific uses.

Trivia: Math Driven Syntax

```
foo[1,...,2]
     ==
foo[(1, Ellipsis, 2)]
```

Other Factors

Python was easy to extend with C extensions and starting with distutils it was possible to distribute them

Windows!

Python has had excellent Windows support in the past unlike many other programming languages that were created in the POSIX environment

Trivia: Battlefield 2 used Python

And since the engine is still used today there are free to play versions of Battlefield still using Python for scripting

Web Development

We slowly and steadily became a proven platform for the web
Python is not the final answer there but an amazing platform to start

Twisted

If you wanted to do networking a few years ago Twisted was the answer

Trivia: Early XMPP Transports

Most of the XMPP to X transports were written in Python with Twisted

But Really …

It's fun!

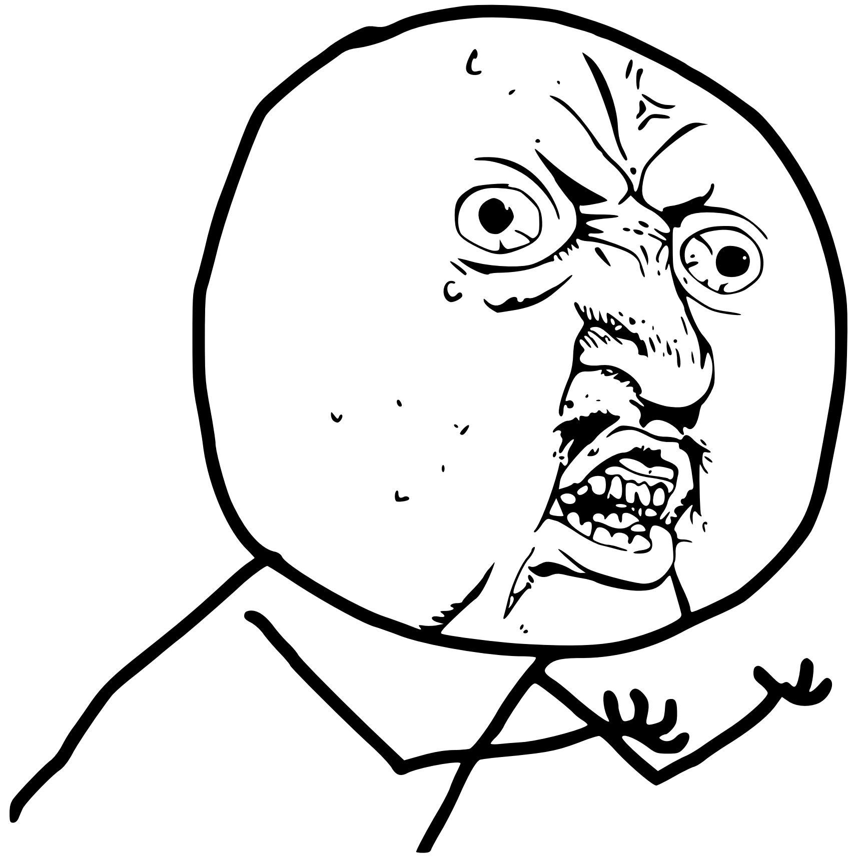
People enjoy working with the language

I have yet to see a Wifi Hotspot Portal Page
that is written in Python
and sucks

Disclaimer: I understand that this statement is very optimistic and bad Python code exists in practice, that there are frameworks in the Python community that advocate for sloppy code, that there are widely used modules with security problems or bad general design, that there are indeed Wifi hotspot login pages that are horrible and indeed written in Python, that there are well written Wifi login pages in PHP (I don't actually believe that), that I am hugely biased and that my sample size is waaaaaaaaaaaaaaaaay too small.

“FUN!?”

What is this?

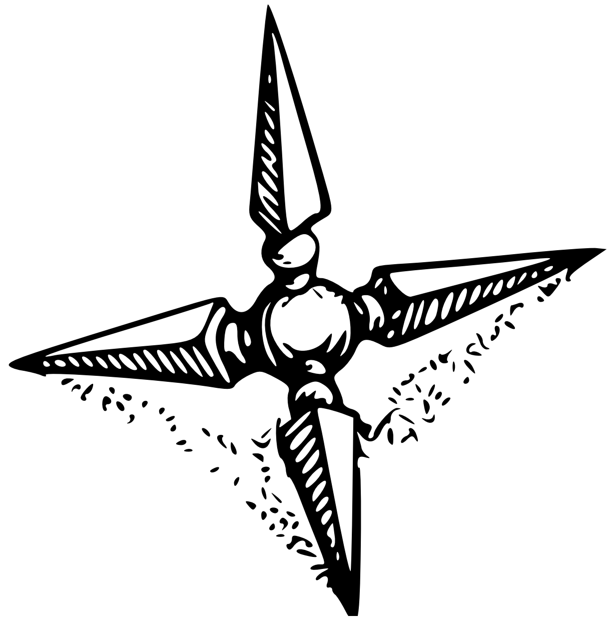


Y U NO WORK

No Running into Walls

Descriptors

Python's most important language feature



What are Descriptors?

- __get__
- __set__
- __delete__
- Common descriptors: functions, properties

Trivia: Functions are Descriptors

that's what makes them methods if placed within classes

Example: Basic Descriptor Lookup

```
>>> class Foo(object):
...  def my_function(self):
...   pass
...
>>> Foo.my_function
<unbound method Foo.my_function>
>>> Foo.__dict__['my_function']
<function my_function at 0x1002e1410>
>>> Foo.__dict__['my_function'].__get__(None, Foo)
<unbound method Foo.my_function>
>>>
>>> Foo().my_function
<bound method Foo.my_function of <__main__.Foo object at 0x1002e2710>>
>>> Foo.__dict__['my_function'].__get__(Foo(), Foo)
<bound method Foo.my_function of <__main__.Foo object at 0x1002e2750>>
```

Example: Everyday Decorators

```
>>> class Foo(object):
...  @property
...  def foo(self):
...   return 'hello pycon'
...
>>> Foo().foo
'hello pycon'
```

Cached Properties

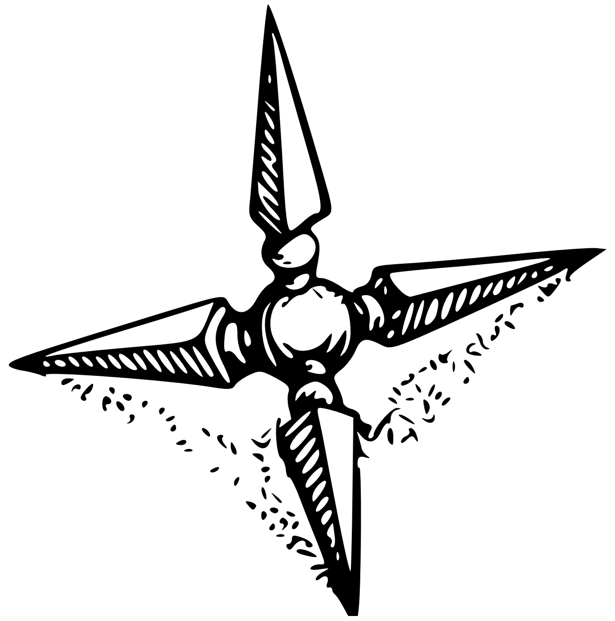
```python
missing = object()

class cached_property(object):

    def __init__(self, func):
        self.func = func
        self.__name__ = func.__name__
        self.__doc__ = func.__doc__
        self.__module__ = func.__module__

    def __get__(self, obj, type=None):
        if obj is None:
            return self
        value = obj.__dict__.get(self.__name__, missing)
        if value is missing:
            value = self.func(obj)
            obj.__dict__[self.__name__] = value
        return value
```

Example: Cached Properties

```
class Post(object):

    def __init__(self, text):
        self.text = text

    @cached_property
    def rendered_text(self):
        return markdown_to_html(self.text)
```

Duck Typing

“if it's not a penguin it must be a duck”



ABDT: *Abstract Base Duck Typing*

abstract bases for improved duck typing

Abstract Base Duck Typing

- `abc.ABCMeta` — metaclass for abstract bases
- `collections.*` — common abstract bases

Abstract Base Duck Typing

```
callable(x)             ->  isinstance(x, Callable)
tryexcept(hash(x))      ->  isinstance(x, Hashable)
tryexcept(iter(x))      ->  isinstance(x, Iterable)
tryexcept(len(x))       ->  isinstance(x, Sized)
tryexcept(hasattr(x, ‘__contains__’))
                        ->  isinstance(x, Container)

                        ->  isinstance(x, Mapping)
                            isinstance(x, Set)
                            isinstance(x, Sequence)
                            isinstance(x, MutableMapping)
                            isinstance(x, MutableSet)
                            isinstance(x, MutableSequence)
```

Example: Abstract Base Duck Typing

```
>>> from collections import Iterator
>>> class MyIter(object):
...   def __iter__(self):
...     return self
...   def next(self):
...     return 42
...
>>> isinstance(MyIter(), Iterator)
True
```

Custom Ducks

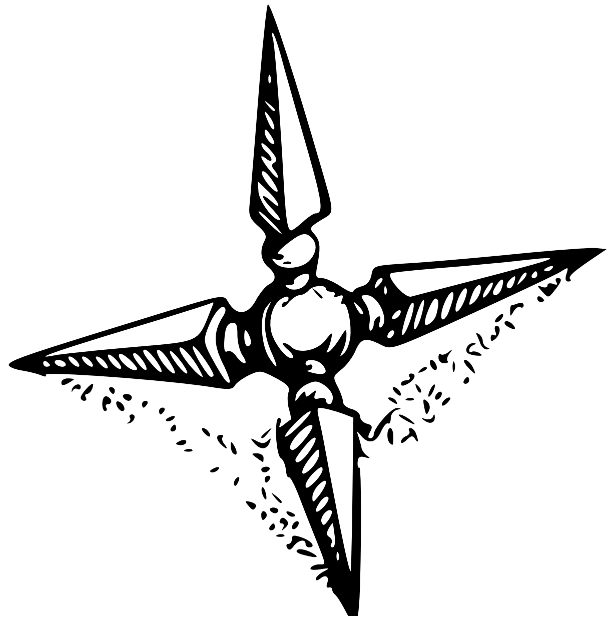
```python
from abc import ABCMeta, abstractmethod


class Markedup(object):
    __metaclass__ = ABCMeta

    @classmethod
    def __subclasshook__(cls, C):
        if cls is Markedup:
            if hasattr(C, "__html__"):
                return True
        return NotImplemented
```

Example: Custom Ducks

```
>>> class Markup(unicode):
...  def __html__(self):
...   return self
...
>>> isinstance(Markup('test'), Markedup)
True
```

Debugging Helpers

use internals to track down bugs



Tracking Imports

```python
import sys
import __builtin__
real_import = __builtin__.__import__

def debug_import(name, locals=None, globals=None, fromlist=None, level=-1):
    glob = globals or sys._getframe(1).f_globals
    importer_name = glob and glob.get('__name__') or 'unknown'
    print '%s imports %s' % (importer_name, name)
    return real_import(name, locals, globals, fromlist, level)

__builtin__.__import__ = debug_import
```

Example: Tracking Imports

```
>>> import urlparse
__main__ imports urlparse
urlparse imports collections
collections imports _abcoll
collections imports _collections
collections imports operator
collections imports keyword
collections imports sys
collections imports heapq
heapq imports itertools
heapq imports operator
heapq imports bisect
bisect imports _bisect
heapq imports _heapq
collections imports itertools
```

Interpreter Frames

```
def print_frame_info(frame):
    print 'module: %s' % frame.f_globals.get('__name__')
    print 'filename: %s' % frame.f_code.co_filename
    print 'current line: %d' % frame.f_lineno
    loc = dict((k, v) for k, v in frame.f_locals.iteritems()
               if not k.startswith('__'))
    print 'local variables: %s' % loc
```

Example: Interpreter Frames

```
>>> import sys
>>> print_frame_info(sys._getframe())
module: __main__
filename: <stdin>
current line: 1
local variables: {
  'a': 2,
  'b': 4,
  'sys': <module 'sys' (built-in)>,
  'print_frame_info': <function print_frame_info at 0x100484668>
}
```

Dumping Threads

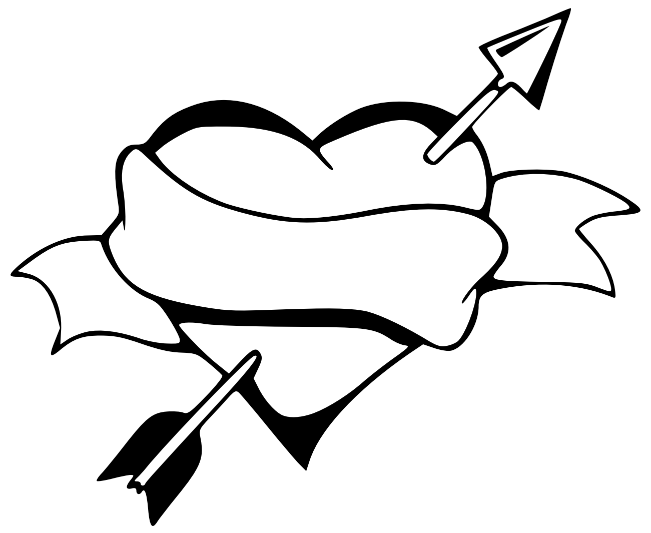
```
import sys
import traceback

def dump_threads():
    for thread_id, frame in sys._current_frames().iteritems():
        print 'Thread #%d' % thread_id
        print ''.join(traceback.format_stack(frame))
```

Example: Dumping Threads

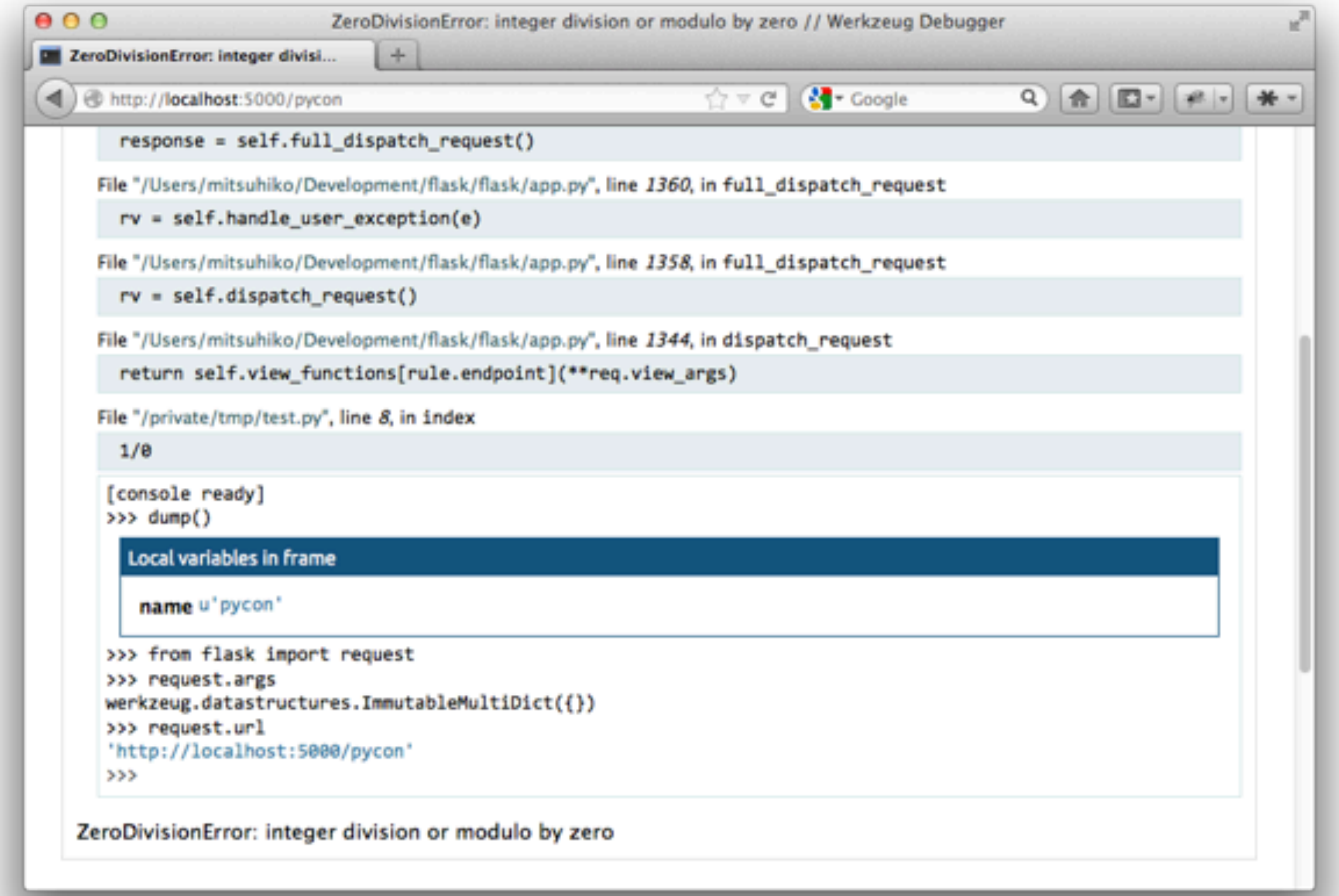
```
>>> import time, threading
>>> def foo():
...  for x in xrange(10):
...   time.sleep(1)
...
>>> threading.Thread(target=foo).start()
>>> dump_threads()
Thread #4302381056
  File "lib/python2.7/threading.py", line 483, in run
    self.__target(*self.__args, **self.__kwargs)
  File "<stdin>", line 3, in foo
    time.sleep(1)

Thread #140735295412576
  File "<stdin>", line 1, in <module>
    dump_threads()
  File "<stdin>", line 4, in dump_threads
    print ''.join(traceback.format_stack(frame)).rstrip()
```

Why we love Python

and why we don't use other things



Win because awesome Fail


ZeroDivisionError: integer division or modulo by zero // Werkzeug Debugger

http://localhost:5000/pycon

```
  response = self.full_dispatch_request()
File "/Users/mitsuhiko/Development/flask/flask/app.py", line 1360, in full_dispatch_request
  rv = self.handle_user_exception(e)
File "/Users/mitsuhiko/Development/flask/flask/app.py", line 1358, in full_dispatch_request
  rv = self.dispatch_request()
File "/Users/mitsuhiko/Development/flask/flask/app.py", line 1344, in dispatch_request
  return self.view_functions[rule.endpoint](**req.view_args)
File "/private/tmp/test.py", line 8, in index
  1/0
[console ready]
>>> dump()
Local variables in frame
  name u'pycon'
>>> from flask import request
>>> request.args
werkzeug.datastructures.ImmutableMultiDict({})
>>> request.url
'http://localhost:5000/pycon'
>>>
```

ZeroDivisionError: integer division or modulo by zero

This is how I “sell” Python

Slow Execution Monitor

- Dump stacktrace if an API request runs longer than N seconds
  - Permanent background thread
  - Request start -> set marker
  - Request end -> remove marker
  - If marker active for more than N seconds -> log stacktrace

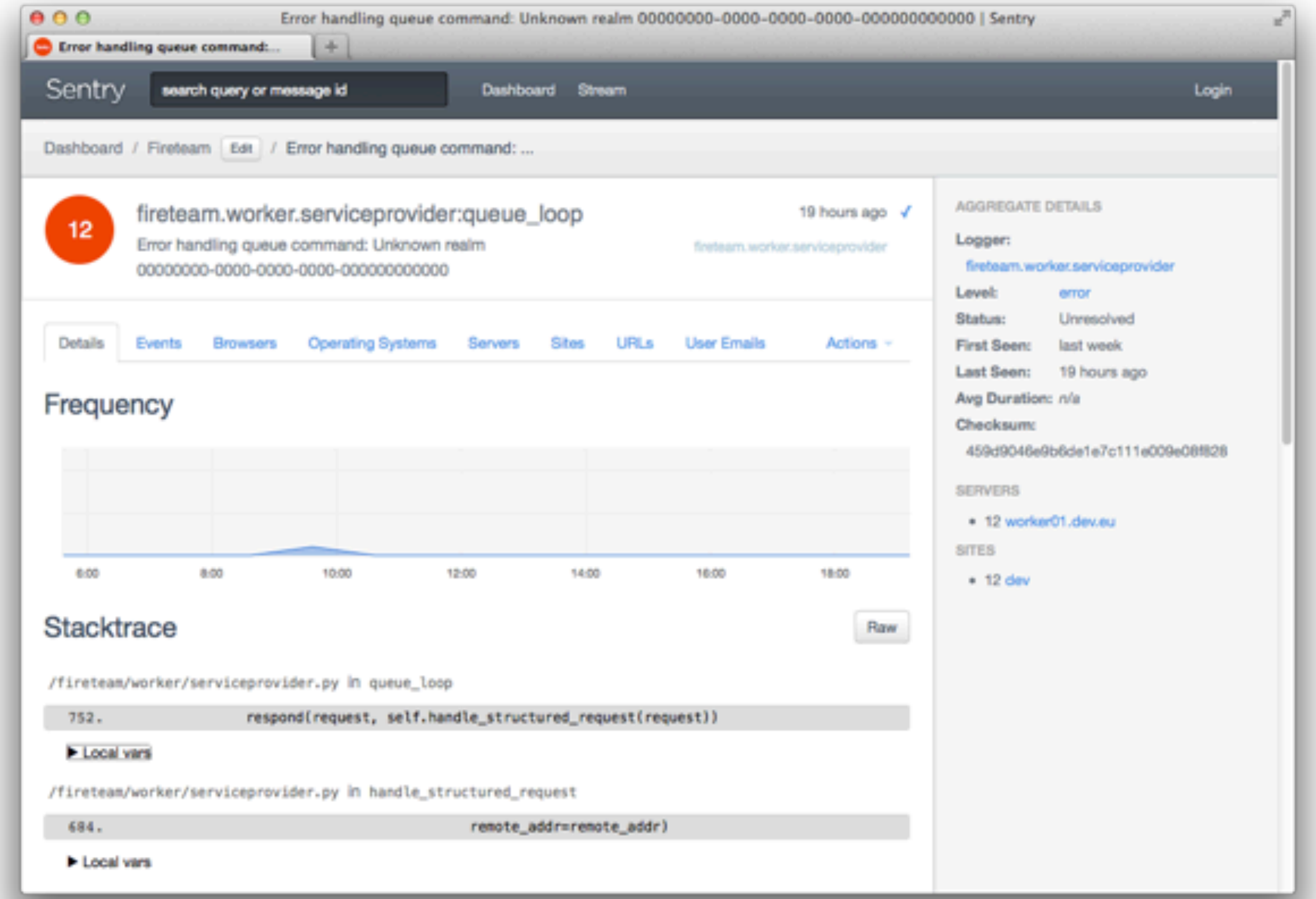
Remote Console

- All Python code has a thread that listens for requests on redis
- Can be used to execute arbitrary Python code for debugging
- Sends results back to redis

Rich Logging

- We log into Sentry error groups
- all stacktraces on dev environments include local variables for all frames
- Also speaks *$language*

Sentry



Error handling queue command: Unknown realm 00000000-0000-0000-0000-000000000000 | Sentry

Error handling queue command:...
Sentry
search query or message id
Dashboard
Stream
Login
Dashboard / Fireteam Edit / Error handling queue command: ...

12 fireteam.worker.serviceprovider:queue_loop

Error handling queue command: Unknown realm 00000000-0000-0000-0000-000000000000

19 hours ago ✓

fireteam.worker.serviceprovider

Frequency

Stacktrace

```
/fireteam/worker/serviceprovider.py in queue_loop
752.                respond(request, self.handle_structured_request(request))
▶ Local vars
/fireteam/worker/serviceprovider.py in handle_structured_request
684.                                  remote_addr=remote_addr)
▶ Local vars
```

AGGREGATE DETAILS

Logger: fireteam.worker.serviceprovider

Level: error

Status: Unresolved

First Seen: last week

Last Seen: 19 hours ago

Avg Duration: n/a

Checksum: 459d9046e9b6de1e7c111e009e08f828

SERVERS

- 12 worker01.dev.eu

SITES

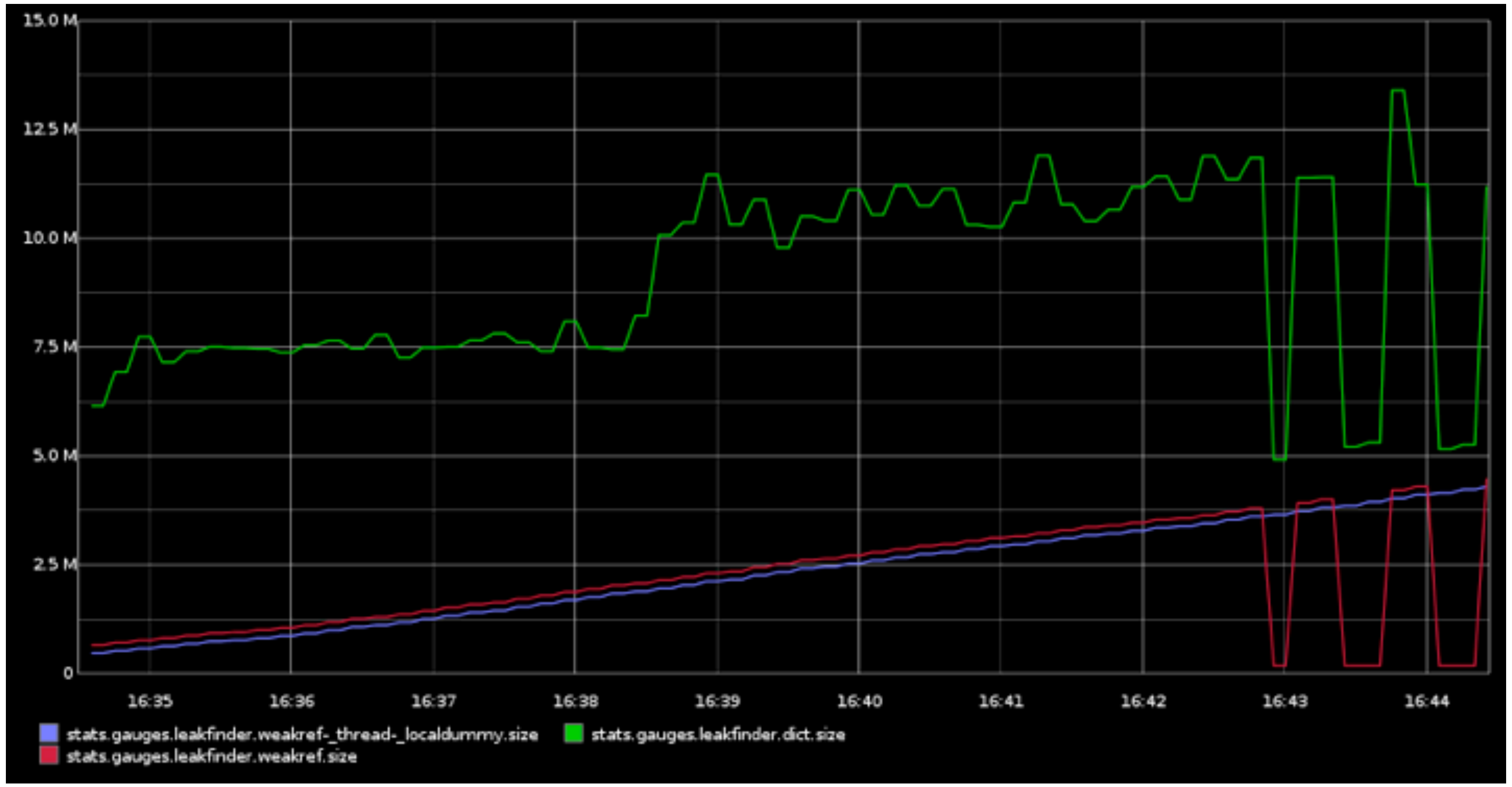
- 12 dev

Includes all information, groups automatically

Memory Leak Finding

- Walk over all objects the garbage collector can reach
- Resolve weak references
- Group by type
- Log to top grossing to graphite every second

Finding a Greenlet Leak



Easy to track down what exactly is happening, ~40 lines of code

Killer Libraries

- SQLAlchemy
- lxml
- *WSGI
- $webframework

virtualenv

- it does not matter that packaging or the import system is broken
- it could be so much worse
- virtualenv makes the big chaos into many separate small chaoses

Easy to Learn

- Language can be picked up in weeks
- It's a fun language to learn
- It's very dynamic and leaves room for (crazy) experimentation

not insane™

An Amazing Community

- Developers and people all around the world
- embraces simple and truly open licenses
- loves documentation
- … now also loves testing

screw hackernews

Worry Less
&
get stuff done

Q&A