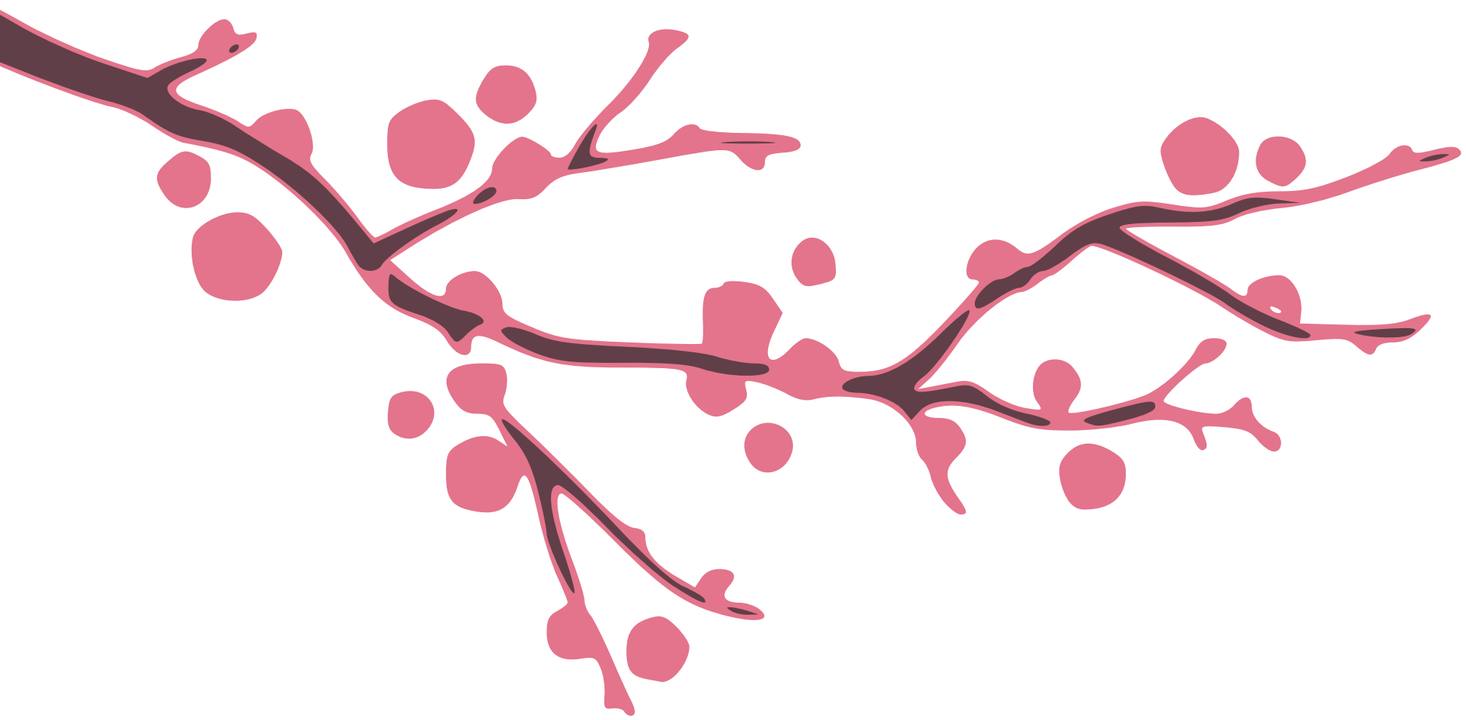
Happiness Through Ignorance

a presentation by Armin Ronacher
for PyCon Japan 2012



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

About the Name

mitsuhiko: name is from the Detective Conan Manga

I don't actually speak Japanese :-(

Foreword

Take everything with a grain of salt

… and that includes this talk

Why Happiness Matters

and why I talk about happiness

Happiness

There is no value in doing something you don't like.
It might work for a while, but you will get grumpy

Happy People are Productive People

If you like your work you are willing to work overtime
Without happiness there would be no Open Source

We Love Python

Many of us are using Python because it makes us happy
(or at least happier than the alternatives)

Why Ignorance Matters

and why being ignorant can be important

Ignorance

We start out ignorant

Education

When we're learning we become less ignorant …

Education

… start learning more and more …

Education

… explore less …

Education

… worry more.

Ignorance is Bliss

Ignorance & dedication gets you far

Wolfire

Indie Game Developer
(known for running the humble indie bundles)

Lugaru

Wolfire's first successful indie game
eventually open sourced under the GPL license

Lugaru

Screenshot from Lugaru

Overgrowth

Screenshot from Overgrowth
(their current game)

void Screenshot(void)

```
// Make an FSSpec
static char buf[256];
if(numscreenshots==0){
buf[0]=26;
buf[1]=':';
buf[2]='S';
buf[3]='c';
buf[4]='r';
buf[5]='e';
buf[6]='e';
buf[7]='n';
/* ... */
buf[26]='0';
}
```

void Game::Tick()

```
{
    declare 40 variables;
    handle network messages;
    handle keyboard input;
    handle main menu code;
    handle all menu pages;
    handle game saving;
    handle game loading;
    handle game sounds;
    handle player movements;
    handle collisions;
    handle attacks;
    handle screenshots;
}
```

Game Ticks

Executed every frame
one function with 10000 lines of C++ code
up to 12 levels of indentation

Dedication

Instead of not doing it
They did it
They made a successful game

Too Much Information

humanity knows so much

I want to make a website

HTML, XHTML, CSS, JavaScript, Python, PHP, Ruby, Templates, Flask, Django, CodeIgnitor, XML, Ruby on Rails, node.js, OpenID, OAuth, Facebook Connect, bcrypt, SSH, SHA1, FTP, HTTP, SPDY, Puppet, Chef, Salt, Backbone JS, MD5, Flash, jQuery, Dojo, DOM, XPath, XInclude, XSLT, Jinja, Genshi, i18n, l10n, unicode, utf-8, MIME, email, websockets, server side events, pubsub, pubsubhubbub, Atom, RSS, …

Where do you even start?

It's increasingly difficult to learn things
people tell you to learn Technology X
when you're done, X gets replaced with Y

Step by Step

You start somewhere and go small steps from there

Quick Iteration

every small step is a achievement

Learn to love and hate

instead of taking hackernews' word that PHP sucks
you can learn it first hand

A Healthy Balance

Ignorance requires a healthy balance
start ignorant — don't end there

Cargo Cult Programming

"why didn't you?"

“Why didn't you use X?”

Chances are that if you present something you did someone will ask why you didn't do it with technology X instead of Y

But it's O(n)!

There is theory and there is practice
Something that's slow in theory could still be
a valid solution in practice

Infinite is a lie

n often really is a constant
think about it

Scripting languages are slow

Can't program computer games in it

Unreal Engine 3 has considerable amount written in Unreal script

Complexity kills Happiness

Examples from the real world

SOAP

Simple Object Access Protocol

SAML 2.0

Security Assertion Markup Language

SAML 2.0

. . . is an XML-based open standard for exchanging authentication and authorization data between security domains, that is, between an identity provider and a service provider.

Specification Breakdown

SAML 2.0, XML, XPath, XPath Filter 2.0, XPointer, XLST, HTTP, XMLENC, X509, XMLDSIG, Canonical XML

This is no Sign-in protocol

. . . it's a way to make money of SAML because barely anyone has the resources to implement it securely

SSO 101

Shared Secret + HMAC + encapsulated payload

SSO 101

```
import hashlib, hmac, json

class BadSignature(Exception):
    pass

def get_signature(payload):
    m = hmac.new(SHARED_SECRET, digestmod=hashlib.sha1)
    m.update(payload)
    return m.hexdigest()

def sign(payload):
    payload = json.dumps(payload)
    return get_signature(payload) + '.' + payload

def get_payload(data):
    if '.' not in data:
        raise BadSignature()
    signature, payload = data.split('.', 1)
    verify_sig = get_signature(payload)
    if verify_sig != signature:
        raise BadSignature()
    return json.loads(payload)
```

Is it secure?

For as long as you have a long secret key which you don't lose.
Takes 10 minutes to implement and is easy to understand.
Would you know if SAML is secure?

Pluggable Applications

All the over-engineering in the WSGI community in the end
just gave us systems that look like J2EE.
Meanwhile Django has a global settings module and is popular

PHP

Barely a programming language, but hugely successful.
No consistent language design but fast iteration speeds.

C

No namespaces, no OOP, not functional, no type safety, bad standard library, worst string type, theoretically hard to optimize, no form of GC — the pillar of modern software development

Personal Guidelines

things I follow because I think they make sense

Disclaimer

Personal experience

I have not nearly done enough to tell others what to do

Learn Asking Questions

And then ask the right ones

I notice many times (on myself and others) that we ask the wrong questions

Avoid Global State

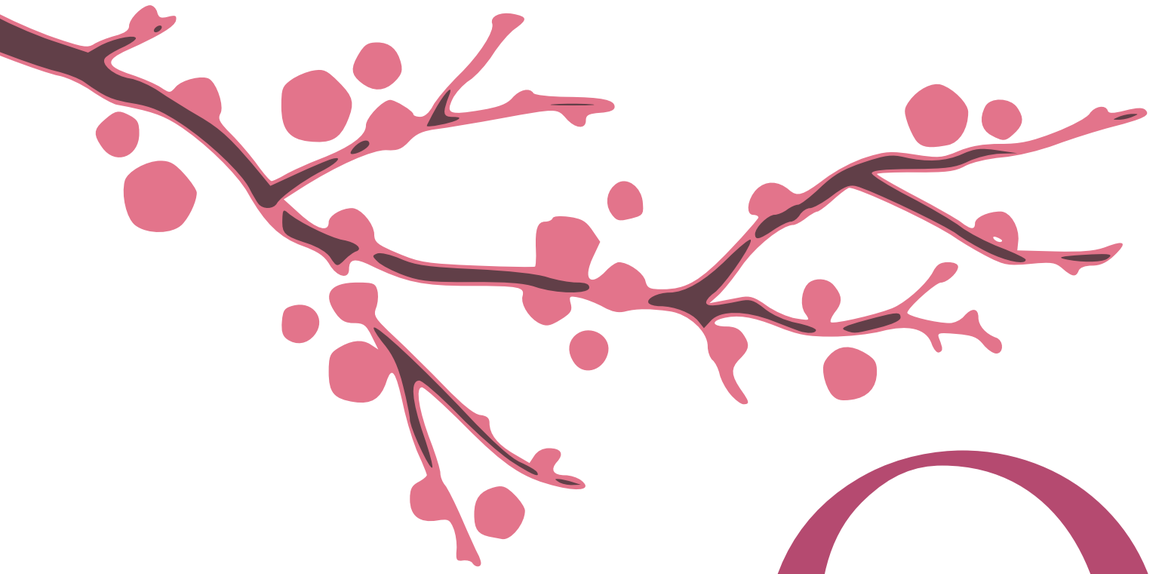
Just avoid it. It's easy to do.
If you think the API suffers consider thread/context locals.
But really. Avoid global state.

Refactor often

At the end of an iteration/milestone go over the code
and try to see if implementation can be simplified

Examples First

I always write APIs and I start with the examples.
Often shows when something does not make sense.



Q&A

FIRETEAM™

http://fireteam.net/ — Armin Ronacher — @mitsuhiko