

# Creating Faster Websites Faster

GDG Devfest Cordillera-Baguio  
November 9, 2013

# Hello.

I am Katherine Pe.

- Software Developer
- Been developing websites since 2003.
- Rubyist who tries other cryptic languages out of need.
- Education: Majored in Financial Management WITHOUT ANY REGRET.

<http://kat.pe>

# Focus

## Google Technologies

- Chrome Developer Tools
- PageSpeed Insights

# Focus

## Best Practices

- 14 Rules for faster-loading websites

# Focus Framework

- Ruby on Rails
- <http://rubyonrails.geekcampbaguio.com>



# Presumed Knowledge of Web Technologies

- HTML5
- CSS
- JavaScript

# Fast vs Slow



Programming is 90%  
thinking & 10% typing

Several disagree & think  
that programming  
is just typing

(I am not a fan of)  
**Jeff Atwood**

Most software that  
have metrics measure  
activity like ODesk

“You may be very active  
but not productive.”

@garybernhardt

Good **planning** takes  
time

but it's **always** worth it.

On going slowly

“Do not be afraid of going slowly, be afraid only of standing still.”



Photo taken by myself at Genting, Malaysia (November 2011)



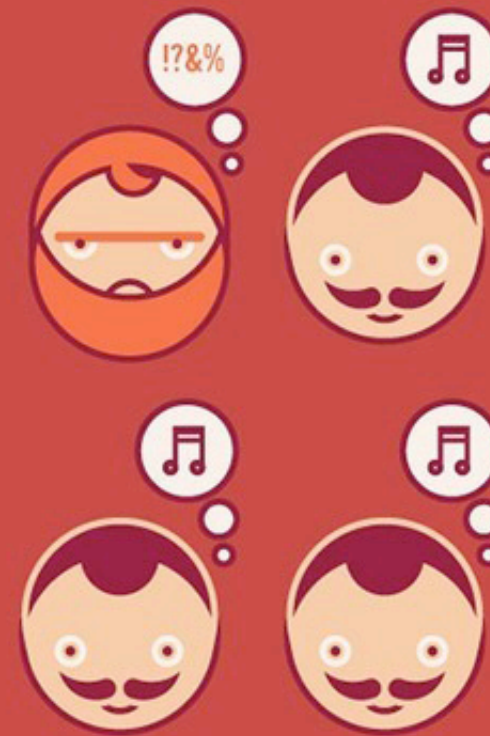
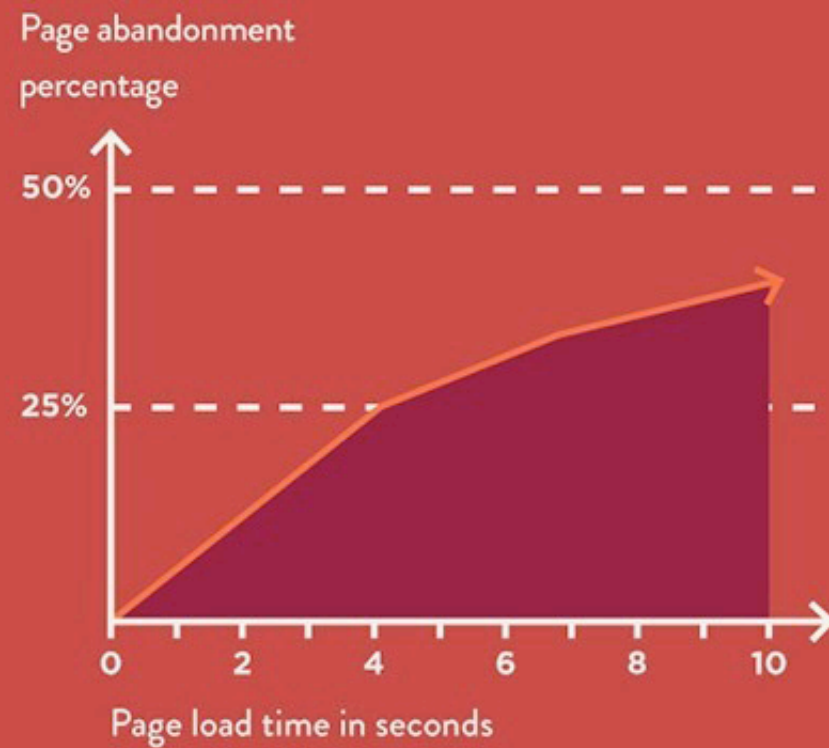
A **slow** site is obviously  
a disadvantage

Optimization is not a  
priority

"We should forget about small  
efficiencies, say about 97% of the time:  
**premature optimization** is the root of all  
evil." — Donald Knuth

Why do premature  
optimization?

# 1 IN 4 PEOPLE ABANDON A WEB PAGE THAT TAKES MORE THAN FOUR SECONDS TO LOAD.



Source: Mashable.com

**What is fast enough?**

- less than 0.1s/ 100ms - fast
- greater than 0.1s/ 100ms - not snappy
- greater than 1s - feels sluggish
- greater than 10s - frustrating
- less than 3s - sane and easy to achieve

# Measuring Latency



- <http://tools.pingdom.com>
- Facebook.com - 25 requests and loads in 831ms
- Google.com - 14 requests and loads in 743ms
- Yahoo.com - 83 requests and loads in 1.07s
- CNN.com - 189 requests and loads in 3.29s

# 14 Rules for Better Web Performance

**Make Fewer HTTP Requests**

# Use a Content Delivery Network

- Or use **Rack::Cache** if you do not want to pay for services
- It is useless to use a **CDN** and **Rack::Cache** at the same time
- **Rack::Cache** will not speed up asset delivery if you use a CDN

# Add an Expires Header

- For **static** components, set **far future Expires** header
- For **dynamic** components, use **Cache-Control** header

# Gzip Components

- All browsers since 1998 support GZIP compression
- Accept-Encoding: gzip, deflate

```
# On a Rack project like  
Sinatra  
# config.ru  
use Rack::Deflater
```

Put **Stylesheets** at the **Top**



Put JavaScripts at the Bottom

**Make JavaScript and CSS  
External**

```
doctype 5
html
  head
    title A Basic Slim Template
    meta charset="utf-8"
    meta name="viewport"
content="width=device-width, initial-
scale=1.0"
  = stylesheet_link_tag 'application'
body
  section
    article
      # A lot of code
    = javascript_include_tag 'application'
```

# Avoid CSS Expressions

- Write efficient CSS
- Avoid the use of `@import` rule
- Avoid complex child selectors that affect performance

// do not write this way

div > div > div > p > a.class | | { }

# Reduce DNS Lookups

// Used to view DNS records  
dig kat.pe

# Minify JavaScripts

- Use an Asset Manager
- Node.js - **Connect Assets**
- Ruby on Rails/ Sinatra - **Sprockets** or "**Asset Pipeline**"
- Use a JavaScript Compressor like **uglifyer**



# Minify Stylesheets

- Use a CSS Compressor
- Node.js - **Connect Assets**
- YUI Compressor
- CSS Condense (Node.js)

# Avoid Redirects

**Remove Duplicate Scripts**  
(also suggests DRY/ Don't Repeat Yourself)

# Configure ETags

Rack::ETag

# Make Ajax Cacheable

# Google Chrome Developer Tools

# Google PageSpeed Insights

[https://developers.google.com/speed/  
pagespeed/insights](https://developers.google.com/speed/pagespeed/insights)

# Ruby on Rails Intro



# Install Ruby on Rails

# Windows

<http://railsinstaller.org>

# Linux or Mac

<http://www.blog.geekcampbaguio.com/2013-rails-install>

Make sure you have new Rails version

```
$ gem install rails
```

```
$ rails -v
```

```
$ Rails 4.0.1
```

# Create a simple photo management app

- Upload photos
- Add a title and a description

```
$ rails new app
```

```
$ cd app
```

```
$ rails g scaffold photo title:string  
description:text
```

```
$ rake db:migrate
```

```
# app/models/photo.rb
```

```
class Photo < ActiveRecord::Base
```

```
  validates :title, presence: true
```

```
  validates :description, presence: true
```

```
end
```

```
$ rails s
```

```
$ open http://localhost:3000/photos
```



# New photo

**2 errors prohibited this photo from being saved:**

- Title can't be blank
- Description can't be blank

Title

Description

Create Photo

[Back](#)

```
$ rails g migration add_picture_to_photo  
picture:string  
$ rake db:migrate
```

```
# app/views/photos/_form.html.erb
```

```
<div class = "field">
```

```
  <%= f.label :picture %>
```

```
  <%= f.file_field :picture %>
```

```
</div>
```

```
# app/views/photos/show.html.erb
```

```
<%= image_tag( @photo.picture_url ) %>
```

```
# app/controllers/photos_controller.rb
```

```
def photo_params
```

```
  params.require(:photo).permit(:title,  
    :description, :picture)
```

```
end
```

# The Asset Pipeline

- Allows easy management of static assets
- Assets are precompiled for deployment

// Usually found on a deployment file or  
done locally

**\$ rake assets:precompile**

**RAILS\_ENV=production**

# Assets are minified or compressed

- YUI compressor - <http://yui.github.io/yuicompressor/>
- uglifier - <https://github.com/lautis/uglifyer>



# Performance Optimization for Ruby on Rails

- Page Caching
- Action Caching
- Fragment Caching

# Resources

- Web Performance Best Practices - [https://  
developers.google.com/speed/docs/best-practices/rules\\_intro](https://developers.google.com/speed/docs/best-practices/rules_intro)
- 14 Rules for faster-loading websites - [http://  
stevesouders.com/hpws/rules.php](http://stevesouders.com/hpws/rules.php)
- Ruby on Rails documentation - <http://guides.rubyonrails.org>
- Geekcamp Baguio Ruby on Rails Guide - [http://  
rubyonrails.geekcampbaguio.com](http://rubyonrails.geekcampbaguio.com)