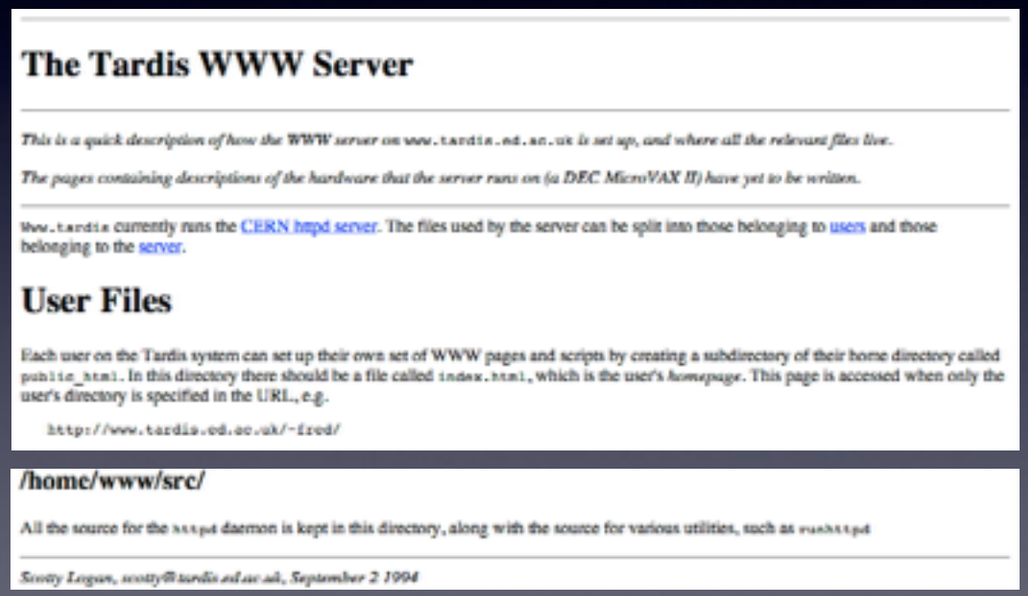
# Mobile: Web & Apps

# The early days



## The Tardis WWW Server

*This is a quick description of how the WWW server on* `www.tardis.ed.ac.uk` *is set up, and where all the relevant files live.*

*The pages containing descriptions of the hardware that the server runs on (a DEC MicroVAX II) have yet to be written.*

`Www.tardis` currently runs the CERN httpd server. The files used by the server can be split into those belonging to users and those belonging to the server.

## User Files

Each user on the Tardis system can set up their own set of WWW pages and scripts by creating a subdirectory of their home directory called `public_html`. In this directory there should be a file called `index.html`, which is the user's *homepage*. This page is accessed when only the user's directory is specified in the URL, e.g.

```
http://www.tardis.ed.ac.uk/~fred/
```

## /home/www/src/

All the source for the `NASpd` daemon is kept in this directory, along with the source for various utilities, such as `runNASpd`

*Scotty Logan, scotty@tardis.ed.ac.uk, September 2 1994*

# Really basic HTML

```
<HTML>
  <HEAD>
    <TITLE>The Tardis WWW Server</TITLE>
  </HEAD>
  <BODY>
    <HR>
    <H1>The Tardis WWW Server</H1>
    <HR>
    <P>
      <I>This is a quick description of how the WWW server on</I>
      <TT>www.tardis.ed.ac.uk</TT>
      <I>is set up, and where all the relevant files live.</I>
    </P>
    <P>
      <I>The pages containing descriptions of the hardware that the
      server runs on (a DEC MicroVAX II) have yet to be written.</I>
    </P>
    <HR>
    <P>
```

Back then that was HTML

Now we call it “Content First”

# It was even “mobile friendly”


http://www.stanford.edu/~swl/tardis.html

# Back to the Future

For a little while

# Survey time

How many of you have an iOS device?

iPod Touch

iPhone

iPad

# Survey time

How many of you have an Android device?

Nexus phone / tablet

Samsung Galaxy

HTC One

# Survey time

How many of you have something else?

Blackberry

Windows Phone

“feature” phone

# Native Apps

Available from app store

Downloaded and installed on device

Available offline

Full device access

camera, location, storage, etc.

Unique development tools  for each

# Native App Cost

Getting less expensive

Mobile app “factories”

Each additional platform adds cost

Can be offset by charging for app

Only really worth it for iOS and Android

some exceptions for niche markets

# Hybrid Apps

- Uses frameworks / wrappers like PhoneGap
- Created as web apps
- Better device access than web
- May not look completely native
- Easier to support less common

# Web Apps

Like a web site, but more “appy”

HTML, CSS, JavaScript

Inherently multi-platform

One codebase

- might include tweaks for some platforms

# HTML5 Web Apps

- More access to device features
  - Location (GPS and compass)
  - Local storage
  - Offline applications
- Feature support varies by OS and version

# Native, Hybrid or Web App

It depends

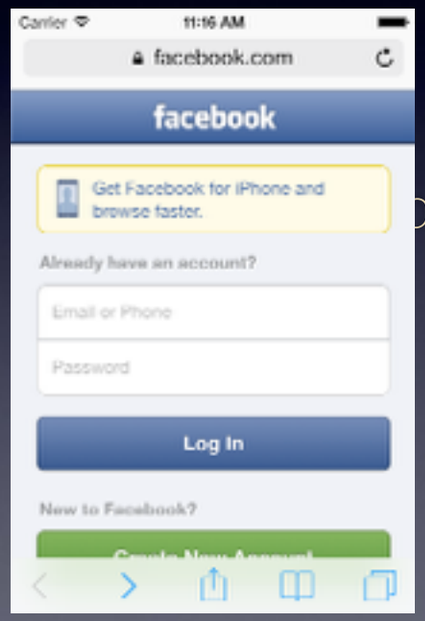
Which platforms do you need to support?

What features do you need?

What's your budget?

How quickly do you need to make

# Mobile Web Required




people

# Web App vs Web Site

- Mobile web site
  - Link to app store(s)
  - No support for devices without apps
  - Might be unpopular with infrequent users
- Mobile web app

# WTF is mobile anyway?

Laptops are not considered “mobile”

One day I used my laptop in

- my home
- my local Starbucks
- my office
- Coupa Cafe (Y2E2)

# Desktop

Large high resolution screen

Large memory

Large storage

Fast CPU

Fast network

Always on network

# Everything else

Small to medium screen

Small to medium resolution

Small memory

Limited storage

Wide range of CPU speeds

Wide range of network speeds

# It’s all relative: my laptop

15” screen (up to 2880x1800)

23” monitor (1920x1200) in office

16GB RAM

500GB fast SSD

2.7GHz quad core cpu

801.11n wifi

# It’s all relative: my first laptop

12” screen (800x600)

15” monitor (1024x768) in office

72MB RAM

2GB slow hard drive

133MHz single core cpu

10Mbps ethernet via PCMCIA card

# It’s all relative: my phone

3.5” screen (960x640)

512MB RAM

64GB flash storage

1GHz dual core cpu

801.11n wifi and 3G GSM

# Network Speed

Laptop:

campus wifi: 91Mbps down / 88Mbps up

home wifi: 35Mbps down / 8Mbps up

Phone:

campus wifi: 21Mbps down / 14Mbps

# Tools

# Editor / IDE

Not getting into editor religions

Pick one that works for you

Should have at least

- formatting help
- syntax highlighting

# Version Control

You need version control

This is not version control:

```
copy thing.html thing.bak
```

Again, pick one…

# git

powerful distributed version control

developed for Linux kernel

GUI and CLI tools, editor/IDE plugins

http://git-scm.com/book/

github.com for publishing / sharing

# git install

already installed on Mac OS X

but use homebrew to upgrade

available as packages on Linuxes

`e.g. apt-get install git`

download for Windows

http://git-scm.com/download/win

# git setup

```
$ git config --global
> user.name "John Doe"
$ git config –global \
> user.email johndoe@example.com
$ git config --global \
> core.editor emacs
```

# start a project

```
$ git init mobileapp
Initialized empty Git repository in
…/mobileapp/.git/
$ cd mobileapp
```

# status

```
$ git status
# On branch master
#
# Initial commit
#
nothing to commit (create/copy files
and use "git add" to track)
```

# first file

```
$ vi README.txt
$ git status
…
#
# Untracked files:
#   (use "git add <file>..." to
include in what will be committed)
#
# README.txt
nothing added to commit…
```

# first checkin

```
$ git add README.txt
$ git status
…
# Changes to be committed:
…
#   new file:   README.txt
$ git commit -m 'First commit'
[master (root-commit) 45ba4bd] First…
 1 file changed, 2 insertions(+)
 create mode 100644 README.txt
```

# clean status

```
$ git status
# On branch master
nothing to commit, working directory
clean
```

# first change

```
$ vi README.txt
$ git status
# On branch master
# Changes not staged for commit:
…
#   modified:   README.txt
#
no changes added to commit (use "git
add" and/or "git commit -a")
```

# what changed?

```
$ git diff README.txt
diff --git a/README.txt b/README.txt
index 16914a8..bcb4238 100644
--- a/README.txt
+++ b/README.txt
@@ -1,2 +1,2 @@
-My awesome app
+My really awesome app
```

# quick commit

```
$ git commit -a -m 'really awesome'
[master abdc37d] really awesome
 1 file changed, 1 insertion(+), 1
deletion(-)
```

# history

```
$ git log
commit abdc37d6973764dc9e720f3945238f67…
Author: Scotty Logan <swl@stanford.edu>
Date:   Mon Jan 27 12:49:15 2014 -0800

    really awesome

commit 45ba4bd75d54b98ec2cc8e2d87e8263f…
Author: Scotty Logan <swl@stanford.edu>
Date:   Mon Jan 27 11:47:16 2014 -0800

    First commit
```

# time travel

```
$ cat README.txt
My really awesome app
$ git checkout 45ba4bd
Note: checking out ‘45ba4bd’.

You are in 'detached HEAD' state…

HEAD is now at 45ba4bd… First commit
$ cat README.txt
My awesome app
```

# back to now

```
$ cat README.txt
My awesome app
$ git status
# HEAD detached at 45ba4bd
$ git checkout master
Previous HEAD position was 45ba4bd…
Switched to branch 'master'
$ cat README.txt
My really awesome app
```

# oops!

```
$ git log
commit abdc37d…
Author: Scotty Logan <swl@stanford.edu>
Date:   Mon Jan 27 12:49:15 2014 -0800

    made my app really awesome

commit 45ba4bd…
Author: Scotty Logan <swl@stanford.edu>
Date:   Mon Jan 27 11:47:16 2014 -0800

    First commit
```

When I was creating these examples, I used a different commit message

Let’s fix that

# oops!

```
$ git commit --amend -m 'really awesome'
[master 54c8b54] really awesome
 1 file changed, 1 insertion(+), 1
deletion(-)
```

# oops!

```
$ git log
commit 54c8b54…
Author: Scotty Logan <swl@stanford.edu>
Date:   Mon Jan 27 12:49:15 2014 -0800

    really awesome

commit 45ba4bd…
Author: Scotty Logan <swl@stanford.edu>
Date:   Mon Jan 27 11:47:16 2014 -0800

    First commit
```

# undo commits

```
$ git revert HEAD
[master f792743] Revert "really awesome"
 1 file changed, 1 insertion(+), 1
deletion(-)
```

You can undo the most recent commits

# undo commits

```
$ git log
commit f792743…
Author: Scotty Logan <swl@stanford.edu>
Date:   Mon Jan 27 13:10:15 2014 -0800

    Revert "really awesome"

    This reverts commit 54c8b54…

commit 54c8b54…
Author: Scotty Logan <swl@stanford.edu>
Date:   Mon Jan 27 12:49:15 2014 -0800
```

git revert creates a new commit to undo previous commits

# tags

Commit IDs are hard to remember

Tag important commits:

```
$ git tag v1
$ git tag -l
v1
$ git checkout v1
…
```

# branches

```
$ git checkout -b fix1
Switched to a new branch ‘fix1'
$ echo fixed >> README.txt
$ git commit -m 'fixed readme' README.txt
[fix1 c42b782] fixed readme
 1 file changed, 1 insertion(+)
$ git status
# On branch fix1
nothing to commit, working directory clean
```

# branches

```
$ git log --format=oneline --abbrev-commit
c42b782 fixed readme
f792743 Revert "really awesome"
54c8b54 really awesome
45ba4bd First commit
$ git checkout master
$ git log --format=oneline --abbrev-commit
f792743 Revert "really awesome"
54c8b54 really awesome
45ba4bd First commit
```

# merging

```
$ git branch -l
  fix1
* master
$ git merge fix1
Updating f792743..c42b782
Fast-forward
 README.txt | 1 +
 1 file changed, 1 insertion(+)
$ git log --format=oneline --abbrev-commit
c42b782 fixed readme
f792743 Revert "really awesome"
54c8b54 really awesome
45ba4bd First commit
```

# remote repos

```
server$ git init --bare mobileapp
Initialized empty Git repository in …/mobileapp/
local$ git remote add origin ssh://server/~/mobileapp
local$ git remote -v
origin ssh://server/~/mobileapp (fetch)
origin ssh://server/~/mobileapp (push)
local$ git remote show origin
* remote origin
  Fetch URL: ssh://server/~/mobileapp
  Push  URL: ssh://server/~/mobileapp
  HEAD branch: master
  Remote branch:
    master tracked
  Local ref configured for 'git push':
    master pushes to master (up to date)
```

# deploy with git

```
repo$ git init newrepo

local$ git clone ssh://repo/newrepo

www$ git clone ssh://repo/newrepo

local$ git push origin master

www$ git pull origin master
```

# github.com

git as a service

free for public repositories

even organizations

issue tracker and wiki per repo

easy forking and “pull requests”

# Hosting

# Web Hosting

Laptops usually only work for simulators

- Try WAMP / MAMP / etc.

Use a server you own, or AWS

- EC2, ElasticBeanstalk, S3 for static

Use github pages for static

What do I mean by static?  Apps that don't rely on backend processes - so no PHP, Rails, Django, etc.

# Hosting with Github

see http://pages.github.com

create new github repository

scottylogan/ghp-test

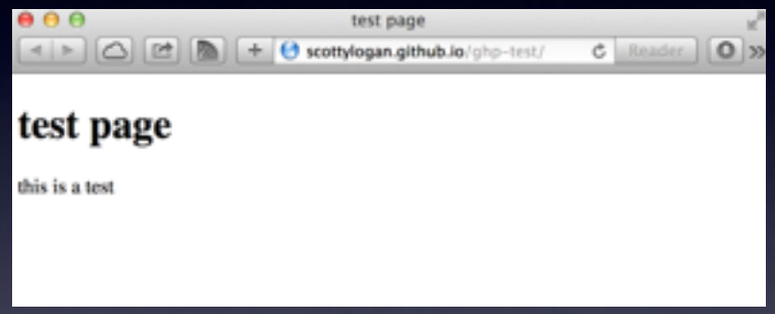
clone, branch, update, push

test

# github pages

```
$ git clone git@github.com:scottylogan/ghp-test.git
Cloning into 'ghp-test'...
…
$ cd ghp-test
$ git checkout -b gh-pages
Switched to a new branch 'gh-pages'
$ vi index.html
$ git add index.html
$ git commit -m 'added index' index.html
[gh-pages d3875c5] added index
…
$ git push --all
To git@github.com:scottylogan/ghp-test.git
 * [new branch]      gh-pages -> gh-pages
```

# github pages test



# Testing

# Desktop Browsers

Chrome, Chrome Canary

Firefox

Safari

IE (at least 9 & 10)

Just in case someone tries to use your mobile web app on a desktop

There’s also Opera, but it’s not common on desktops

# Mobile Browsers

Safari

Android Browser

Chrome (Android, maybe iOS)

Windows 8?

Opera Mini?

There’s no way to set Chrome as the default browser on non-jailbroken iOS devices, so most still use Safari, and other apps (e.g. Mail) will only open links in Safari

Opera Mini still popular outside US, on “dumb” phones - again depends on audience

Blackberry is disappearing, but may be relevant to your audience for some apps

# Devices

- You still need to test on real devices
  - Navigation items
  - Color schemes
    - Dark room vs sunny outdoors
  - Performance on cell networks

# Which Devices?

iOS: iPhone or iPod touch, iPad

Android: pick a few (incl. Samsung)

Windows: Nokia?

Android is tough - many different screen sizes and resolutions, tweaked versions of the Android browser, etc.  Samsung appears to be the largest Android vendor

Try phone stores - AT&T, Verizon, Best Buy, etc.
Apple Store is good for range of current iOS devices

# (E|Si)mulators

iOS Simulator

comes with Xcode

Android Emulator

comes with Android Developer Kit

http://developer.android.com/sdk/

# Differences

The iOS simulator is fast because it runs iOS apps compiled for Mac

The Android emulator is slower (but more “correct”) because it runs apps on emulated hardware

Intel note on spying up android emulator http://software.intel.com/en-us/android/articles/speeding-up-the-android-emulator-on-intel-architecture

# Browser-aaS

Submit URLs to remote services

- http://saucelabs.com/
- http://browserstack.com/

Wide range of browsers

Wide price range

Browserstack also has support for responsive sites

# More than Screenshots?

- BaaS generate static images
  - No interaction
  - No details
  - No DOM inspector

On desktops there’s FireBug for Firefox, Chrome Developer Tools, and Safari Web Inspector

# Remote Inspectors

http://html.adobe.com/edge/inspect/

http://ghostlab.vanamco.com

http://labs.iqfoundry.com/muir/

http://browsync.net/

Run an app on your desktop, connect mobile devices, control them all at once
Some tools will copy form data to mobile devices
Some tools allow for remote DOM / CSS / JS inspection