Fluent Knowledge Share

Jefferson Lam
May 6th, 2015
“Fluent is for everyone who has a hand in web development, from front-end to back-end and everything in between.”
Notes & Resources

• Web Development & Standards
  • Git, JS, CSS, Performance
• Design
  • Performance
• Culture
  • Diversity, Learning
Gitting More Out of Git
Jordan Kasper (Strongloop)
Gitting More Out of Git
Jordan Kasper (Strongloop)

• Advanced Git tips and techniques
  • resetting
  • reflog
Resetting

```bash
~$ git reset --hard HEAD^  
~$ git reset --hard HEAD^^  
~$ git reset --hard origin/master
```
Reflog

```bash
~$ git reset --hard HEAD^  
HEAD is now at 84e1f6f

~$ git reflog

84e1f6f HEAD@{0}: reset: moving to HEAD^  
d078d5e HEAD@{1}: commit: Fix a giant bug  
84e1f6f HEAD@{2}: commit: Add a new feature!  
142b98d HEAD@{3}: commit: First commit
```
Reflog

~$ git reflog
HEAD is now at 84e1f6f

~$ git reset --hard HEAD^
Uh oh, I reset and lost some important commits!
The Reflog never forgets (commits)

~$ git reflog

84e1f6f HEAD@[0]: reset: moving to HEAD^  
d078d5e HEAD@[1]: commit: Fix a giant bug  
84e1f6f HEAD@[2]: commit: Add a new feature!  
142b98d HEAD@[3]: commit: First commit
Undoing Things with Reset & Reflog

~$ git reset --hard HEAD@{1}
HEAD is now at d078d5e

~$ git reflog

d078d5e  HEAD@{0}: reset: moving to HEAD@{1}
84e1f6f  HEAD@{1}: reset: moving to HEAD^
d078d5e  HEAD@{2}: commit: Fix a giant bug
84e1f6f  HEAD@{3}: commit: Add a new feature!
142b98d  HEAD@{3}: commit: First commit
Undoing Things with Reset & Reflog

~$ git reset --hard HEAD@{1} 
HEAD is now at d078d5e

~$ git reflog

d078d5e HEAD@{0}: reset: moving to HEAD@{1}
84e1f6f HEAD@{1}: reset: moving to HEAD^ 
d078d5e HEAD@{2}: commit: Fix a giant bug
84e1f6f HEAD@{3}: commit: Add a new feature!
142b98d HEAD@{3}: commit: First commit
• This dude is alive!
• JS compilers are pretty good
• JS is ever evolving. Embrace the change
ECMAScript Harmony
Brendan Eich (JS Creator)

Video
https://www.youtube.com/watch?v=PlmsweSNhTw&list=PL055Epbe6d5ZqIHE7NA5f61q_bZNjuWvS&index=4
Eliminate JS Smells
Elijah Manor (Ramsey Solutions)

• Standards for non-smelly JS code
• Techniques for eliminating smells
Eliminate JS Smells
Elijah Manor (Ramsey Solutions)

Slides
http://elijahmanor.github.io/talks/js-smells/#/
SMACSS Your Sass Up
Mina Markham (IBM Design)
SMACSS Your Sass Up
Mina Markham (IBM Design)

- A concept on how to architect your Sass
- Keep your Sass scalable and modular
“CSS is bullshit.”
- Mina Markham
“This is not a homepage. It’s a collection of modules that make up the homepage.”

- Mina Markham
Example SMACSS Architecture

+ scss/
  | + base/ # reset, typography, site-wide
  | + layout/ # major components, e.g., header, footer etc.
  | + modules/ # minor components, e.g., buttons, widgets etc.
  | + states/ # js-based classes, alternative states e.g., success or error
  | + themes/ # (optional) separate theme files
  | + utilities/ # non-CSS outputs (i.e. mixins, variables) & non-modules
  | + _shame.scss # because hacks happen
Example SMACSS Architecture

+ scss/
  |
  | + base/ # reset, typography, site-wide
  |
  | + layout/ # major components, e.g., header, footer etc.
  |
  | + modules/ # minor components, e.g., buttons, widgets etc.
  |
  | + states/ # js-based classes, alternative states e.g., success or error
  |
  | + themes/ # (optional) separate theme files
  |
  | + utilities/ # non-CSS outputs (i.e. mixins, variables) & non-modules
  |
  | + _shame.scss # because hacks happen
Example SMACSS Architecture

+ scss/
  |
  | + base/  # reset, typography, site-wide
  |
  | + layout/  # major components, e.g., header, footer etc.
  |
  | + modules/  # minor components, e.g., buttons, widgets etc.
  |
  | + states/  # js-based classes, alternative states e.g., success or error
  |
  | + themes/  # (optional) separate theme files
  |
  | + utilities/  # non-CSS outputs (i.e. mixins, variables) & non-modules
  |
  |   + _shame.scss  # because hacks happen
Example SMACSS Architecture

```
+ scss/
 | + base/  # reset, typography, site-wide
 | + layout/  # major components, e.g., header, footer etc.
 | + modules/  # minor components, e.g., buttons, widgets etc.
 | + states/  # js-based classes, alternative states e.g., success or error
 | + themes/  # (optional) separate theme files
 | + utilities/  # non-CSS outputs (i.e. mixins, variables) & non-modules
 | + _shame.scss  # because hacks happen
```
Example SMACSS Architecture

+ scss/
  |
  | + base/    # reset, typography, site-wide
  |
  | + layout/  # major components, e.g., header, footer etc.
  |
  | + modules/ # minor components, e.g., buttons, widgets etc.
  |
  | + states/  # js-based classes, alternative states e.g., success or error
  |
  | + themes/  # (optional) separate theme files
  |
  | + utilities/ # non-CSS outputs (i.e. mixins, variables) & non-modules
  |
  |   + _shame.scss # because hacks happen
Example SMACSS Architecture

```
+ scss/
  |  
  | + base/  # reset, typography, site-wide
  | 
  | + layout/  # major components, e.g., header, footer etc.
  | 
  | + modules/  # minor components, e.g., buttons, widgets etc.
  | 
  | + states/  # js-based classes, alternative states e.g., success or error
  | 
  | + themes/  # (optional) separate theme files
  | 
  | + utilities/  # non-CSS outputs (i.e. mixins, variables) & non-modules
  | 
  | + _shame.scss  # because hacks happen
```
Example SMACSS Architecture

+ scss/
  |
  | + base/  # reset, typography, site-wide
  |
  | + layout/  # major components, e.g., header, footer etc.
  |
  | + modules/  # minor components, e.g., buttons, widgets etc.
  |
  | + states/  # js-based classes, alternative states e.g., success or error
  |
  | + themes/  # (optional) separate theme files
  |
  | + utilities/  # non-CSS outputs (i.e. mixins, variables) & non-modules
  |
  | + _shame.scss  # because hacks happen
Example SMACSS Architecture

+ scss/
  |
  | + base/       # reset, typography, site-wide
  |
  | + layout/     # major components, e.g., header, footer etc.
  |
  | + modules/    # minor components, e.g., buttons, widgets etc.
  |
  | + states/     # js-based classes, alternative states e.g., success or error
  |
  | + themes/     # (optional) separate theme files
  |
  | + utilities/  # non-CSS outputs (i.e. mixins, variables) & non-modules
  |
  |   + _shame.scss # because hacks happen
SMACSS Your Sass Up
Mina Markham (IBM Design)

Slides
http://minamarkham.github.io/smacss-sass-up/#/
Performance

How Users Perceive the Speed of The Web
Paul Irish (Google Chrome)

Extreme Mobile Web Performance
Maximiliano Firtman

Design + Performance
Steve Souders (SpeedCurve)

Automate Your Site's Front-End Performance
Kitt Hodsden (CodingClan LLC)
In summary: Bad performance is bad UX.
In 1993,

based on research from 1968 & 1991

100ms
1000ms
10 seconds

Response Time Limits: Jakob Nielsen (1993)
100ms, 1000ms, 10s

100ms gives the feeling of **instantaneous** response.

Results feel immediate.

Any longer and the connection between action and reaction is broken.

1000ms keeps the user's flow of thought **seamless**.

Things feel part of a natural and continuous progression of tasks.

Beyond it, the user loses focus and attention.

Beyond **10 seconds** you've lost the user's **attention**.
Three Takeaways for Web Devs after 2 Weeks of Painfully Slow Internet

- “Some sites never loaded”
- “Takeaway 1: Develop in extra slow emulated mode”

https://medium.com/@zengabor/three-takeaways-for-web-developers-after-two-weeks-of-painfully-slow-internet-9e7f6d47726e
"I used to be a reckless designer."

- Yesenia Perez-Cruz (Philadelphia, PA)
REQUESTS: 136
PAGE WEIGHT: 5.9MB
LOAD TIME: 2M 46S
For example…

*image removed*
Tools & Techniques

- Chrome Dev Tools
  - throttling, timeline
- http://www.webpagetest.org
- http://whatdoesmysitecost.com/
- In-browser reminders during development
  - e.g. Etsy engis use in-browser performance widget
- Have a performance budget
  - ms, kb
Welcoming the Web
Estelle Weyl (standardista.com)

Diversity in Tech: The Distorted Truth
Christina Truong

Making Badass Developers
Kathy Sierra (SeriousPony)
• The web development community isn’t very welcoming to everyone (particularly non-male)
“50% of women in STEM leave due to hostile work environments.”

- Estelle Weyl
There are cultural biases in the tech community
“It wasn’t long before I realized I was different, and my presence made other people curious, or even uncomfortable.”

"I learned to make myself less different, just so I could make my life a little easier."

"What I didn't realize at the time was that I was casually erasing my identity."

- Christina Truong
“English-speaking employers in Montreal, Toronto, and Vancouver were 35% more likely to interview an applicant with an English sounding name.”

“In a study where the same resume was screened under two different names - Jennifer and John - Jennifer was perceived as less competent than John.”

- Christina Truong
"It's difficult to confront or admit you may harbor feelings that go against your beliefs about equality, but identifying it is the first step to correcting it."

- Christina Truong
"Privilege is when you think something is not a problem because it's not a problem to you personally" - @davidgaider #GDC
• Learn better by understanding the science of cognitive resources!
“When you ask experts ‘how did you do that?’, they don’t know, they just know.”

“Whatever you practice, the longer you practice being a beginner at it, the better you get at staying a beginner.”

- Kathy Sierra
Making Badass Developers
Kathy Sierra (SeriousPony)

Video

https://www.youtube.com/watch?v=FKTxC9pi-WM
Web Development & Standards

• Git
  • Get comfortable with the reflog!
• JS
  • Use linters!
  • Embrace the change
Takeaways

Web Development & Standards

• CSS
  • Use Sass!
  • SMACSS your Sass up
• Performance
  • Make it a priority!
  • Use performance testing tools
Takeaways

Design

• Performance
  • Performance is a UX concern!
  • Design within a performance budget
Takeaways

Culture
• Diversity
  • Be conscious of & identify cultural & gender biases
• Learning
  • Don’t make people choose cake
  • Utilize the A-B-C method
Thank you!

Notes & Resources