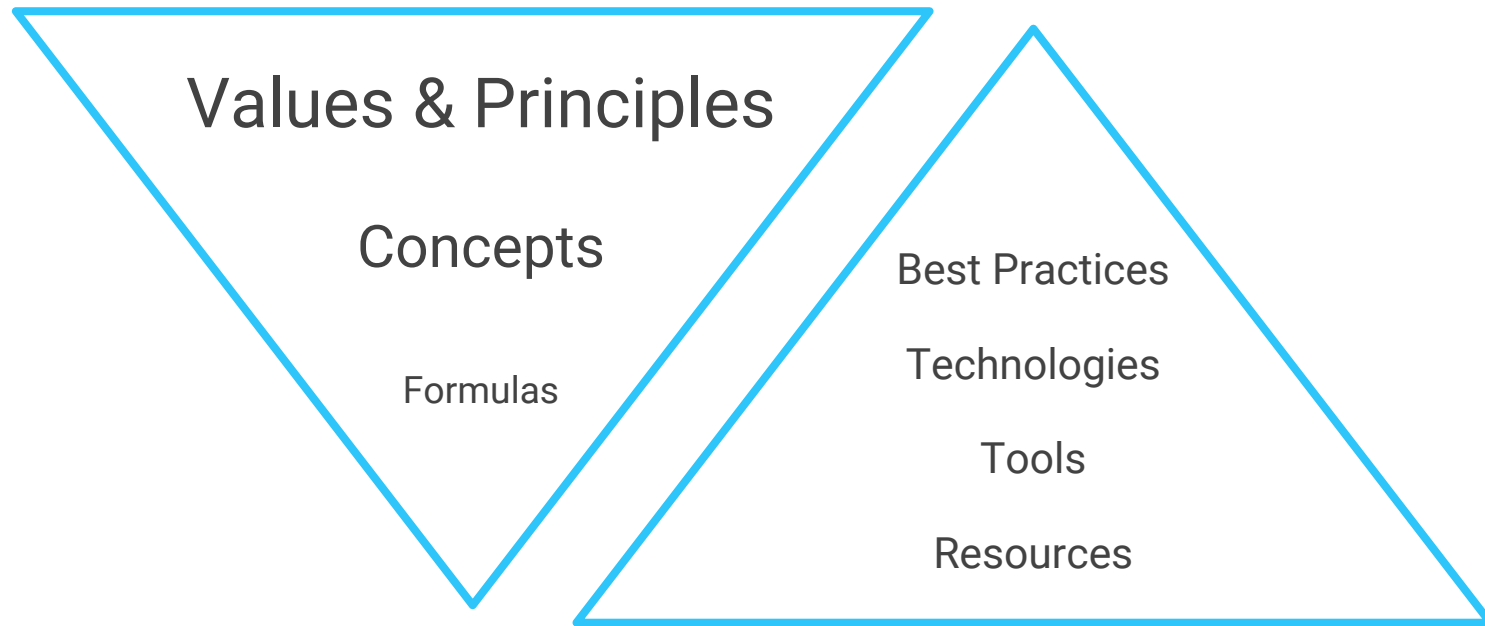


Course Overview



What You Learn



Curriculum - 3 Modules

M1 - WebDev 101

HTML

CSS

Javascript

DOM

OOP

Canvas

M2 - Backend

ES6

HTTP

Node.js

Express

MongoDB

M3 - SPA + REST

jsx

React

REST APIs



How You Learn

Rituals

Lessons

Practice

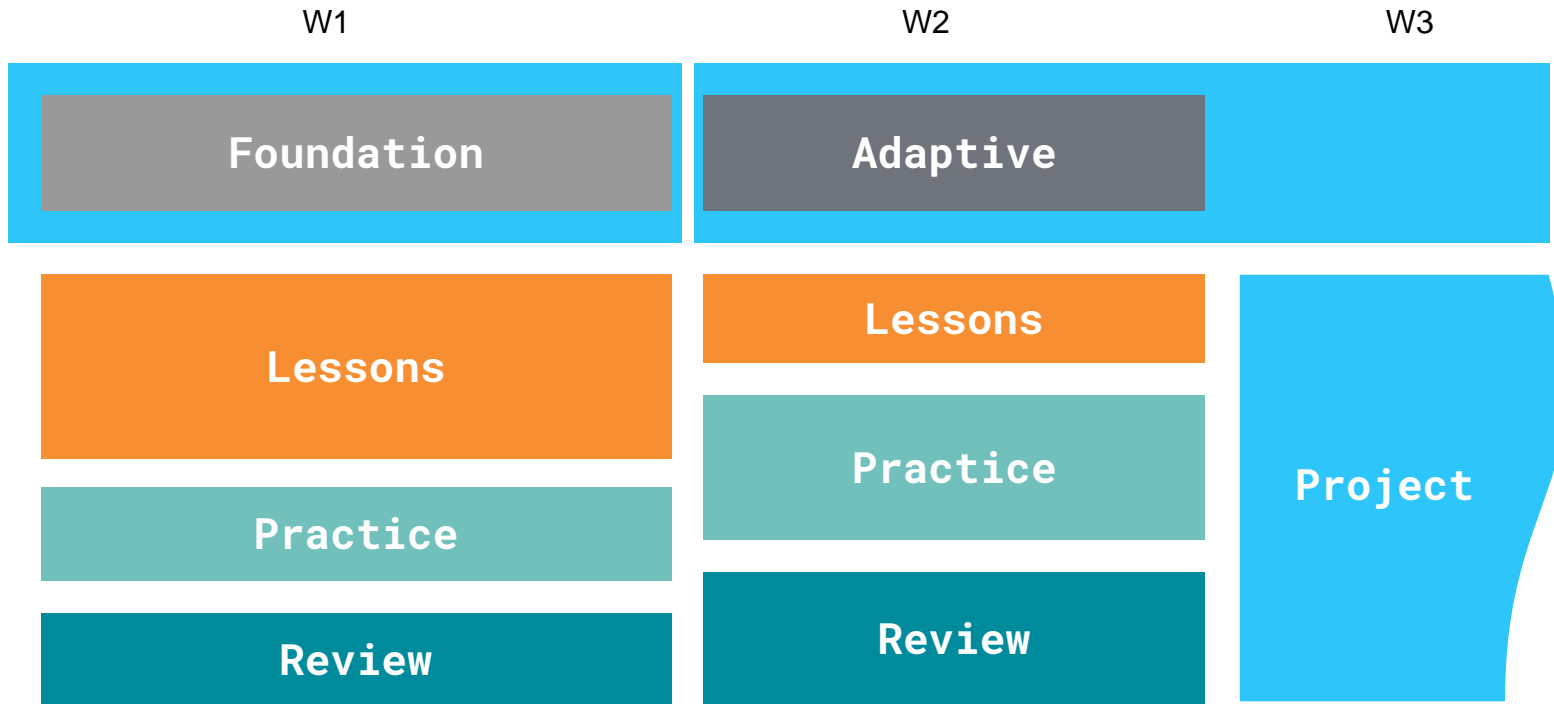
Review

Projects

Resources

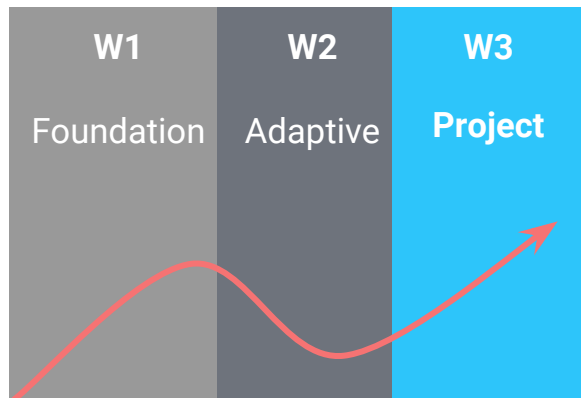
IRON
HACK

Module Structure

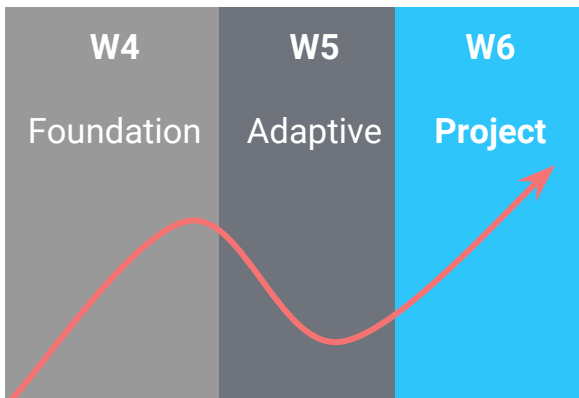


Flow - Intensity

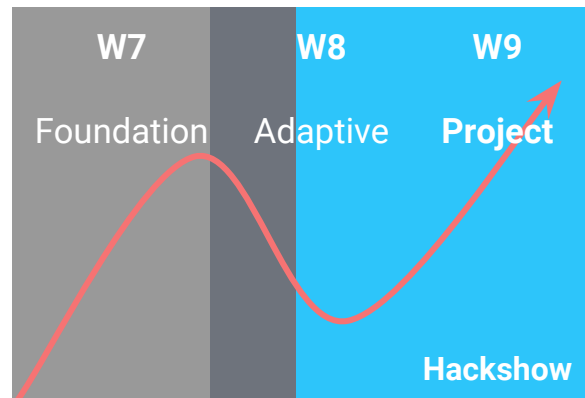
M1 - WebDev 101



M2 - Backend



M3 - SPA + REST



Rituals



Class Master

Calendar

Stand-up

Daily kick-off

Class Master

One TA per week

Maintain calendar

Run schedule

Reminders

Fix blockers

What is?

Producer

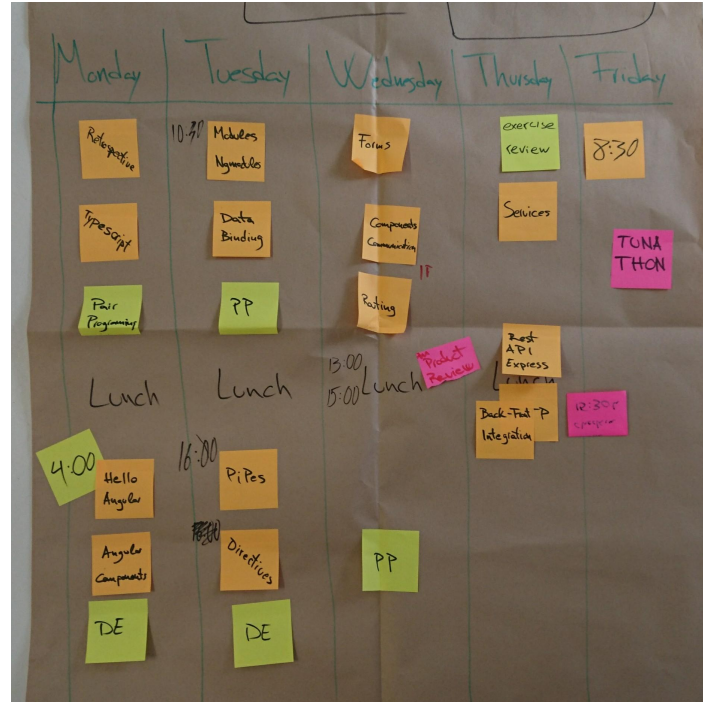
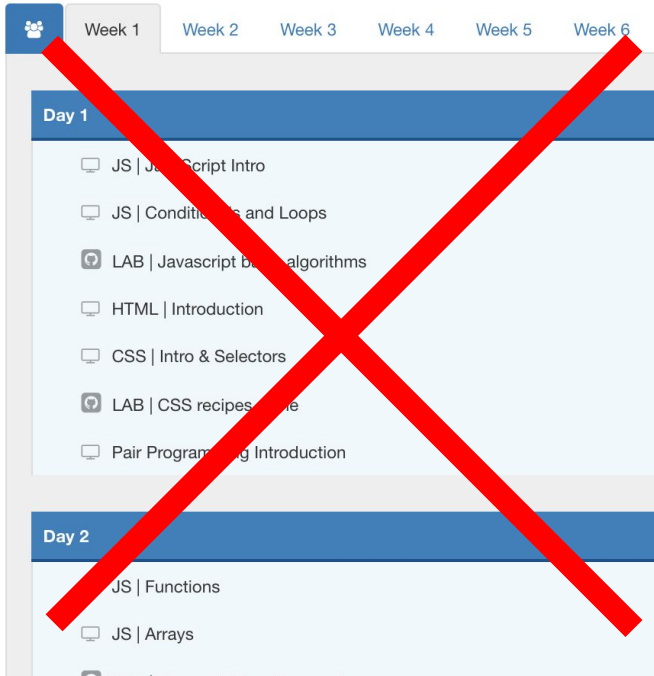
Facilitator

Servant leader

“Scrum” Master



Calendar



Stand-up (~15 ')

Everyday practice

9 AM

Standing up

15 mins

Practicing and improving everyday

Everyone speaks

How are you feeling?

What did you achieve yesterday?

PR?

What do you expect from today?

Is there something blocking you?



Daily Kick-off (~10')

Review agenda/options for the day

- exercise reviews
- learning objectives
- scope of practice sessions
- activities with a purpose
- **+ Adaptive** decide together on lessons, practice and reviews

Make changes to calendar

Agree

Recap & Go



Lessons



Lectures

Activities

Quick Demos

Code Alongs

+ Adaptive Lessons

Lectures (1h ~ 2h)

Lecturing is hard for everyone

Students should:

- stay focused and on topic OR quiet
- not be on laptops if no laptop needed
- ask questions
- questions not answered? take notes for later
- ask to switch to code/browser and fix zoom
- ask for a break if they need



Activities (30' ~ 60')

Learning is rewarding for everyone

- role play
- diagramming
- mobbing

Students should:

- be open to go out of their comfort zone



Quick Demos (~30')

Teachers code, students watch

Introduction to concepts and technologies

Students should:

- not code, just focus
- ask questions
- take notes for later



Code Alongs (2h ~ 4h)

Teachers code, students follow

Broken into steps

Students should:

- understand the purpose of each step
- ask questions related to the step
- take note of other questions for later
- once you're done, help other
- Focus and silence, no conversation
- Answer clearly to "CAN WE CONTINUE" and "WHO IS BLOCKED"



Adaptive Lessons (2h ~ 4h)

All class OR smaller groups

+ Adaptive

- redo a concept or topic
- deep dive into a concept or topic
- advanced concepts and topics

Students should:

- ask for it
- decide when/what/who



Practice



Pair Programming

Daily Exercise

+ Adaptive Practice

Pair Programming (1h ~ 3h)

Students working together on same activity, in groups of 2 or 3

1 codes (driver), the other(s) dictates (navigator)

Students should:

- decide role in activity
- use 1 laptop
- communicate
- switch roles after 45'

Steps:

- fork + clone
- commit + push
- create pull request
 - BCN - name + name
 - tag your pairs @name
 - add comments (questions/blockers)



Daily Exercise (1h ~ 3h)

Helps consolidating knowledge

Go a step further than what we talked in class

Students should:

- do 10% at least
- submit before stand-up

Steps:

- fork + clone
- commit + push
- create pull request
 - BCN - name + name
 - tag your pairs @name
 - add comments (questions/blockers)



Adaptive Practice (2h ~ 4h)

All class OR smaller groups

- redo/improve exercises
- custom exercises
- special (mini) projects

+ Adaptive

Students should:

- ask for it
- decide when/what/who



Review

(Daily) Exercise Review

+ Adaptive Review



(Daily) Exercise Review (~30')

Class decides if/when to do it

Not a review of an entire exercise

No official solution

- review and comment one or more PR
- solve one problem/bug in one PR

Students should:

- ask questions (prepare for it)
- share their ideas
- share their solutions



Adaptive Review (1h ~ 2h)

All class OR smaller groups

+ Adaptive

- on-demand Q&A sessions
- review PRs of exercise

Students should:

- ask for it
- decide when/what/who
- prepare for it (scope, questions, blockers)

Projects



Projects

Retrospectives

Projects (6d ~ 8d)

Kick-off (2d ~ 3d)

Idea & Validation

Planning & Sign-off

NOTE: until all projects are signed-off, reviews and advanced classes are still be in progress

Execute (4d ~ 5d)

Stand-up

Kata

Code / Test / Refactor / Deploy

Presentation

Demo

Replan

Deliver (1d)

Code freeze

Presentations



Retrospectives (2h)

Reflect on how the team is working together

What is working well and what to improve

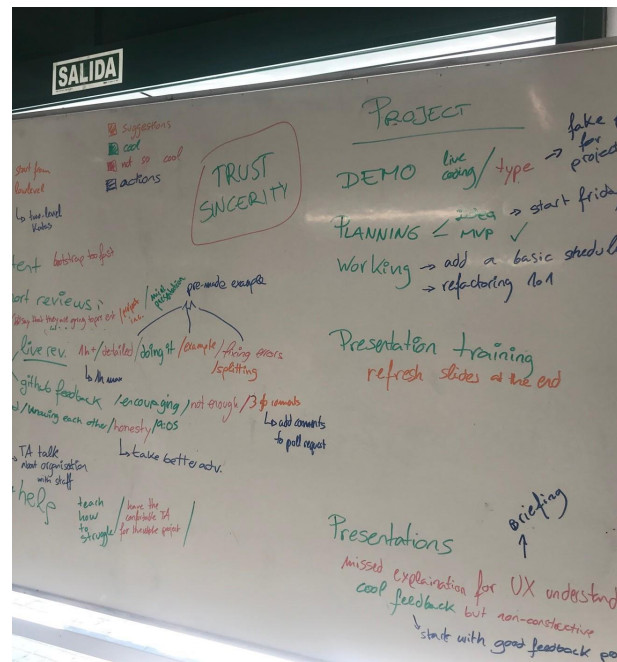
Generate insights

Take actions

Improve your experience

Improve the next module

Improve the bootcamp



Collaborations

UX/UI - WebDev feedback

Hackathon



UX/UI feedback (~30')

M2 WebDev projects

UX/UI students feedback

Wireframes

UI

UX/UI coding week

WebDev feedback

Personal portfolio

HTML, CSS, Javascript



Hackathon (1d)

One day challenge

Project from start to end

Real world experience

UX/UI - WebDev Teams

2-3 UX/UI - 2-3 WebDev

Experience

Work under pressure

Collaboration

Communication

Learn from others



Resources



> [Class Guidelines](#)

> [Module 1 Guide](#)

> [Module 2 Guide](#)

> [Module 3 Guide](#)

> [Materials](#)

> [Cheat-sheet](#)

> [Mantras](#)

> [TA Article](#)