📄 CYF CORE COMMUNITY: The job market for junior web de…

↩ Curriculum re-write planning 2023

Programme 1, ITD 30 days

Leaders: Zachee ,Sayo, Mohsen
Members:

- Sayo

- 20-40 hours work
- Tech - 60% and PD - 40%

A mix of Scrimba and Applied Digital Skills from Google, with one CYF challenge at the end

1. Write a CV in Google docs with [Generative AI](#)
2. Evaluate data with Google Sheets: [a career in data analytics](#)
3. Build and deploy five websites to Netlify [with Scrimba](#) including the capstone project step with review
4. Complete [CYF Blocks](#)

~~Build a Personal Home Page that passes all the tests~~

By the end of ITD, you should have
1. A CV which you have written and revised
2. A spreadsheet you have made in Google sheets
3. A portfolio website deployed to Netlify
4. A basic understanding of programming: variables, loops, control flow

It seems like Fundamentals should be broken up : blocks go into ITD, user stories etc into ITP
✅

Programme 2 ITP 12 sprints

Leaders: Barath Vignarajah, Caroline Scherf
Members:
- Sherif Eid
- Jack Rogers
- Jay Mayer
- Gary Lei
- Kirill Kovzel

https://cyf-programming.netlify.app/
https://github.com/orgs/CodeYourFuture/projects/179/views/3

4 modules of 3 sprints (not weeks as it will be self paced)
https://codeinplace.stanford.edu/ <= start with this - it's 6 weeks long so can cover Module 1 and 2

1 - User focused data 3 sprints

https://cyf-programming.netlify.app/user-data/
Topics
1. Git (moved from Induction)
2. **SQL (in spreadsheets)**
3. Product (user stories etc) from Fundamentals
4. Browser (devtools, asking questions, validating)
5. Python
6. ===> Move components into JS1

 User focused data goals
- Translate requirements into high-level design outlines
- Break down solving problems into testable steps
- Ask questions systematically using a given formal language
- Play computer #1 - predict explain loop
- Identify variables, scope, and state
- Demonstrate good citizenship on GitHub by participating in code review on PRs

2. Structuring and testing data 3 sprints

https://cyf-programming.netlify.app/structuring-data/
JS1 as was

- Identify bugs in someone else's code (person/machine)
- Translate requirements into unit tests
- Play computer #2 -
- Distinguish interface from implementation
- Interpret test feedback
- Describe the intended behaviour of a program and its actual behaviour

3. Data groups 3 sprints

https://cyf-programming.netlify.app/data-groups/
JS2 as was

- Use data structures to model some part of a problem domain
- Break a problem down into different test cases
- Identify acceptance criteria for a problem
- Build tests to constrain the output of an LLM

4. Data flows 3 sprints

https://cyf-programming.netlify.app/data-flows/
JS3 as was

- Describe how data is mapped to a UI
- Break down a UI into components
- Reason about code and time. *Develop our mental model of runtime. When does code run?*
- Define, use, update, and derive state

At the end of ITP, you should be able to SEE

1. Predict and explain the execution of a computer program
2. Reason about where and when software is executed
3. Refactor and iterate code based on new requirements
4. Extract simple composable functions for common use cases
5. Interpret business requirements and express them as test cases
6. Create a strategy for implementing an application based on a set of user stories
7. Design interactions with a provided data set
8. Verify the accessibility of a web application
9. Describe how processes and file descriptors really work
10. Open a pull request and iterate on it in response to feedback

- Codewars 5
- The Bandit 30

Programme 3: Piscine 21 days

Leaders: Daniel, Nirmeet, Ali
?? Groups

- Pairing - if your pair is hopeless you lose out -this is a problem
- Can we provide pre-completed levels

- Can we have three unrelated pieces of work: eg here is a JSON blob, give me the most listened to song
- Must apply with a tailored CV and cover letter
- Must pass the interview (which includes the tech demo, tech test and competency-based evaluation)

The selection committee for the SDC will be small, initially run by Syllabus, and volunteers will be admitted by a process not just by rocking up

https://githubnext.com/projects/copilot-workspace/ <= assuming this will be available, we could require a live code entry in 2025

Programme 4: SDC (new style) 12 weeks + 4 weeks + 6 weeks

https://sdc.codeyourfuture.io/

📄 New style SDC

NOTES

Immediate Tasks

- Update ITD and hand over
- Rm cohort progress rules from first 4 modules ✅

- Move Github stuff from Induction into "HTML-CSS" and rename this module to… ? ✅
- Move CYF induction to after JS3 ✅
- Build yet another home page! In fact build yet another tenant with a new home page ✅
- Design an assessment
    - First filter: We produce a library (which probably accepts callback functions) and have people solve problems using our library
    - Second filter: A short pairing session
-

Piscine Goals

- agree initial criteria for new TV Show piscine

1st requirements (end of ITP, before start of piscine)

B2 English CEFR
Functional Maths 3
Cover letter <- minor manual check.
Codility, get 100% on the kata?
Rainbird check

2nd filter (end of piscine, before start of main course)

- Logical reasoning exercise
- Coding session
- 3 weeks, 3 projects, 2 teams (pair) 1 individual

- parcel up work

See https://github.com/CodeYourFuture/immersive-go-course/blob/main/CONTRIBUTING.md for guidance

Handle data in JS
Shopping cart - JS core 3 assessment
Splitwise?

Manipulate data in some meaningful way

 Mitch - Identify functional skills and partners to obtain free stuff - at least 5 potential partners (June 7th)
 Daniel - Establish project contributing criteria! What are the rubrics? What does success look like? - E.g. no auth! (June 7th)
Sally (June 13th) - Build out the sift 1 interface

To be answered…
How does this work?
Who is going to do this?
What volunteer presence do we want?