web programming

inception


oli


rec

agenda

- course info
- final project
- quick survey
- dev environment setup

course info

course goals

- *"After successfully completing this course, you will:*
    - *be able to **create** attractive, small scale web apps that work in most (modern) browsers."*
    - *have the background knowledge to **understand** technical writings/discussions about the web."*
    - *have the foundation to **decide** where to go next."*

course non-goals

- you will **NOT** leave this class knowing:
  - how to code (*)
  - everything there is to know about web programming (**)

(*): unless you already knew how to (**) unless you already knew it

course overview

introductory

you will learn the fundamentals

opinionated

what we think you need to learn as a beginner

hands-on

you will get your hands dirty, YES

course structure

- the class will be divided in **teams** of 5 people that will mimic "real life" dev teams
  - each team will work on a project of **their choice** (*) that will evolve throughout the course, contributing with code and reviews
- **0 exams**
- **1 final presentation**

(*): Must be approved by the teachers.

course deliverables

every GitHub interaction

commits, pull requests, reviews, etc.

web app deploys

URLs of both front & back ends

final presentation

to the rest of the class

course grades

- your participation as an **individual** will be evaluated in the following areas:
  - **solution**: architecture, design, etc.
  - **markup + css**: document structure, semantic tags, etc.
  - **aesthetics**: how much you care about the app's look and feel
  - **team work**: the relationship with your teammates
  - **adherence to standards**: respect for code style and any other convention

less theoretical

more practical

DISCLAIMER

- this is the first offering of the course, which means:
  - **everything** is subject to change
  - there will be **bugs**
  - **feedback** is especially appreciated

final project

quick survey

complete the survey

goo.gl/Tzdi2G

dev environment setup

prereqs

- before continuing, make sure you have installed:
  - git: https://git-scm.com/book/en/v2/Getting-Started-Installing-Git
  - latest **Node.js**: https://nodejs.org/en/download/current/
  - latest **Google Chrome**: https://google.com/chrome/
  - any IDE, we recommend **VS Code**: https://code.visualstudio.com/

repository

- go to github.com
- click on "New repository"
- choose the following options and then click "Create repository":
    - name: `web-programming-boilerplate`
    - public
    - README: yes
    - .gitignore: Node
    - license: MIT
- click on "Clone" and copy the URL

```
$ git clone <repo_url> && \
  cd web-programming-boilerplate
```

```
$ git checkout -b setup
```

npm

- 
- package manager for Node.js
- CLI helps install and maintain your project's dependencies
- online repository with thousands of packages: npmjs.com

```
$ npm init
```

package.json

```
{
  "name": "web-programming-boilerplate"
}
```

reference: https://docs.npmjs.com/files/package.json

linting

- **static analysis** of code
- **code-quality** rules: find problematic patterns in code
- **formatting** rules: find code that doesn't adhere to style guidelines

linting – ESLint

- 
- open source JavaScript linting utility
- we will use it for code-quality rules only (not formatting)
- config: `@ucudal/eslint-config`

set up ESLint

goo.gl/rSAaDP

.eslintrc.json

```
{
  "extends": "@ucudal/eslint-config"
}
```

reference: https://eslint.org/docs/user-guide/configuring

linting – stylelint

- 
- open source CSS linting utility
- we will use it for code-quality rules only (not formatting)
- config: `@ucudal/stylelint-config`

set up stylelint

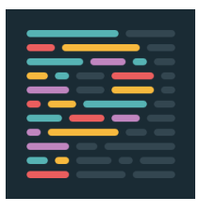
goo.gl/egVWER

.stylelintrc.json

```
{
  "extends": "@ucudal/stylelint-config"
}
```

reference: https://stylelint.io/user-guide/configuration/

formatting – Prettier

- 
- code formatter
- opinionated
- we will use it in conjunction with ESLint and stylelint

set up Prettier

goo.gl/L3wCRN

.prettierrc.json

```
{
    "singleQuote": true,
    "trailingComma": "all"
}
```

reference: https://prettier.io/docs/en/configuration.html

.eslintrc.json

```
{
    "extends": [
        "@ucudal/eslint-config",
        "plugin:prettier/recommended"
    ]
}
```

.stylelintrc.json

```
{
   "extends": [
      "@ucudal/stylelint-config",
      "stylelint-prettier/recommended"
   ]
}
```

testing – jest

- 
- zero configuration
- jsdom built-in

```
$ npm i -D jest
```

jest.config.js

```
module.exports = {
    testEnvironment: 'node',
    verbose: true,
};
```

reference: https://jestjs.io/docs/en/configuration

package.json

```
{
  "name": "web-programming-boilerplate",
  "scripts": {
    "lint:js": "eslint .",
    "lint:css": "stylelint **/*.css",
    "lint": "npm run lint:js && npm run lint:css",
    "pretest": "npm run lint",
    "test": "jest --passWithNoTests"
  },
  "devDependencies": {
    "@ucudal/eslint-config": "...",
    "@ucudal/stylelint-config": "...",
    ...
  }
}
```

continuous integration

- go to travis-ci.org
- click on "Sign Up" and sign up using GitHub
- click on the "+" symbol next to "My Repositories"
- turn on "web-programming-boilerplate"

.travis.yml

```yaml
language: node_js
node_js:
  - '10'
```

reference: https://docs.travis-ci.com/user/languages/javascript-with-nodejs

pull request

- go to the `web-programming-boilerplate` repo page on github.com
- switch to "setup" branch
- click on "New pull request"
- give it a name and click on "Create pull request"
- Wait for Travis to run and merge on success

thanks!