

CSE 104 - Web Programming

General Organization

Contacts

Damien Rohmer (teacher in charge)

damien.rohmer@polytechnique.edu

Fabio Roda

fabio.roda@polytechnique.edu

Maxime Buron

maxime.buron@polytechnique.edu

Pauline Olivier

polivier@lix.polytechnique.fr

Presentation

Damien Rohmer: *damien.rohmer@polytechnique.edu*

Office in LIX - Alan Turing Building

Prof: 50% Teaching, 50% Research

Research at **LIX**

Laboratoire d'Informatique de l'Ecole Polytechnique

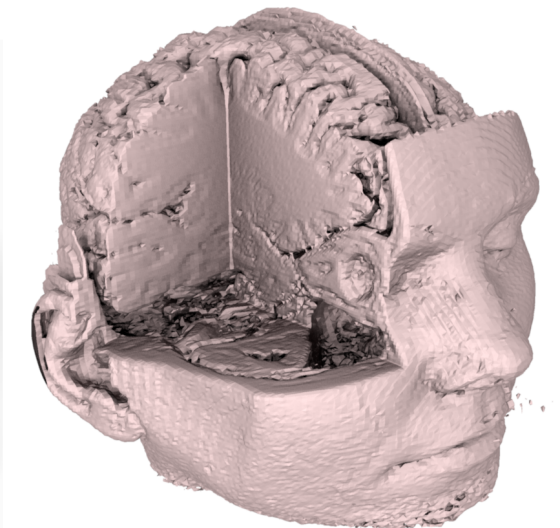
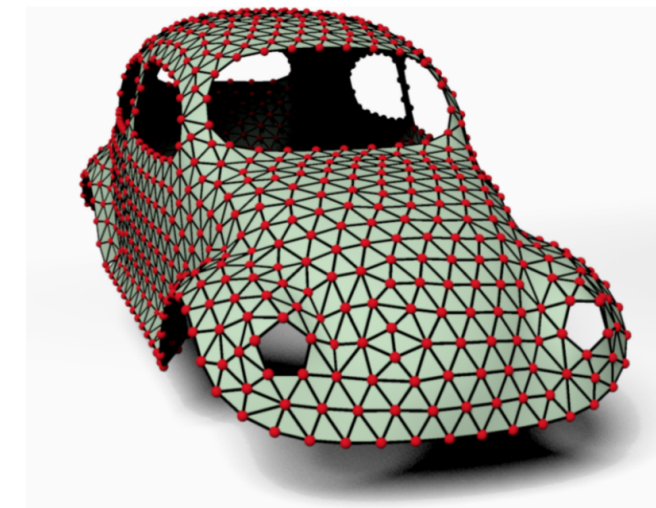
- Part of the pole: Modeling, Simulation & Learning
- Head of the team: **Geometric & Visual Computing**

www.lix.polytechnique.fr/geovic

- Research: **Computer Graphics** - 3D Modeling & Animation

Developing algorithm to ease the creation of 3D virtual content

Application in Entertainment (Cinema, VFX, Video Game), Design & Fabrication, Natural Sciences



Member of the Department of Informatics (DIX)

- Contact between you and the departments
 - Questions about research in CS at LIX, Research seminars, etc.

General objective of CSE104

- Be able to **create webpages from scratch** which are
Valid, standard compliant, responsive using **HTML and CSS**
Coding dynamic behavior of webpages using **JavaScript**
- Basic notions on server developement (PHP, nodeJS)
- Improve general computer development skills
General CS skills: Command line in Unix, Files handling
Good practices: Validity checking
Autonomy: Documentation research, Debug

What this class is not

- Not a network class

Focus on the use of language, not communication protocols

- Not a theoretical/conceptual class

Focus on practical development (few classes, lots of practice)

- Not a framework class (no bootstrap, jQuery, Django, Symfony, etc)

Focus on current language standard

- Not an advanced server side class (no login, security/hack, DB, etc)

Most time spent on client-side development, server development will be only an introductory level

Methodology

- Mostly **practical** computer lab class
 - Set of online tutorials and exercises
 - You code, develop, try, at your speed
 - You develop a small-project

⇒ Get your hands in the code !

- Some sparse general presentation before each new concepts
 - HTML/CSS
 - JavaScript
 - Server programming

Provisional schedule

- HTML/CSS , JavaScript , Server start with ~1h lecture.
- Schedule may be adapted

10/02 HTML/CSS 1

Lecture + Lab

17/02 (No Class)

24/02 HTML/CSS 2

02/03 HTML/CSS 3

09/03 JavaScript 1

*Lecture + **Test 1** + Lab*

16/03 JavaScript 2

23/03 JavaScript 3

30/03 JavaScript 4

06/04 JavaScript 5

*Lab + **Test 2***

13/04 (No Class)

20/04 Project

27/04 Project

04/05 Project

11/05 Server

Lecture + Lab

18/05 Server

25/05 Server

?? Server

*Lab + **Test 3***

Evaluation

- 3 small tests (~50%) (maybe only 2, maybe 4)
 - Practical tests on computer and/or quizz
- 1 project (~50%)
 - Develop a dynamic website with HTML/CSS, JavaScript (client-side).
 - Work in pair (or alone if you prefer)
- No final exam

Note - Tutorials and exercises are not aimed to be graded

- Don't hesitate to ask questions
- Work on them seriously
- Make sure you understand it before going to the next one

Be aware of plagiarism in your project

- Web code is easy to share and reuse
- It is ok normal to share ideas and get inspiration on internet!
- It is **not ok** to use other's ideas and code **without citing it!**

Rule - If you inspire from

- Some internet website
- a book
- your colleagues, previous students
- ... (non exhaustive list)

⇒ **Cite them !** ⇐

Better to cite "too much" than not enough

Web languages

You can find *everything* on web development on the web !

- *everything* can be good - comprehensive, advanced notion.
- *everything* can be bad - outdated, non standard, overkill, dangerous practices.

⇒ Use internet resources, but wisely !

- Do not follow blindly every website that seems to have *a solution*
- Keep analyzing the code
- Double check comments and suggestions

Common issues: Out-date practices, Over-use of heavy library/frameworks

- HTML has evolved a lot through years
- Many *non developer* users (as long as it works)
- Old clever tricks may be replaced by simpler/cleaner standard code
CSS animation, flexbox, grid-layout, new JS standard, etc.

Programming rules (even in non web development)

1- Aim at **most simple readable** solution

- No tricky code (even if it works): un-readable => un-maintainable => un-reliable
- KISS principle (Keep It Simple, Stupid)

(Writing simple and readable code may not be simple!)

2- Prefer simple **standard** code rather than using existing libraries

Standard will be maintained through time, libraries may not

Libraries add extra cost (bandwidth, latency, maintainance, bugs, etc.)

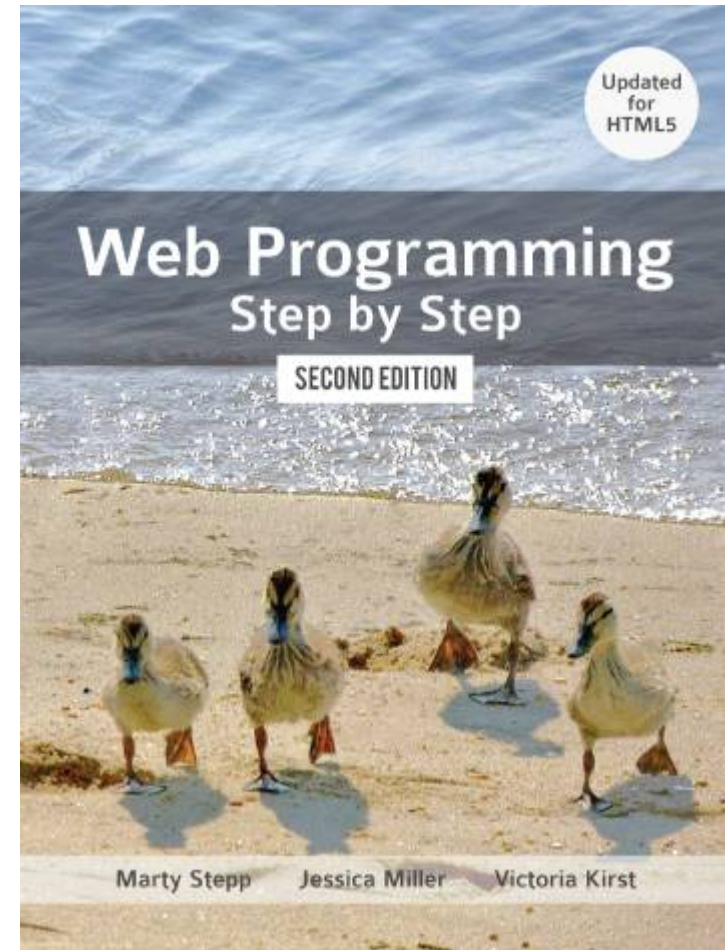
3- But prefer existing libraries to complex home made tricks

Do not re-invent the wheel for real projects (only to learn)

General Reminder

Main references

Victoria Kirst, CS193X, Web programming Fundamentals, Stanford.



Marty Stepp, Jessica Miller, Victoria Kirst

Ressources

- All lab class in computer rooms (35,36)
- In general: working on your own laptop is fine
 - You need a text editor and a modern browser ([see list](#))
 - You need Python, PHP, and Node.js for Server class (in May)

Warning : Don't trust too much your browser

- Web browsers try to hide your errors as much as they can
 - May looks ok on your browser/OS, but not with others*
- Always check (in particular your project) on **computer room browser** (the reference)
- Check, if possible, with Chrome and Firefox
 - It is ok if IE doesn't work (didn't implement latest standard)*