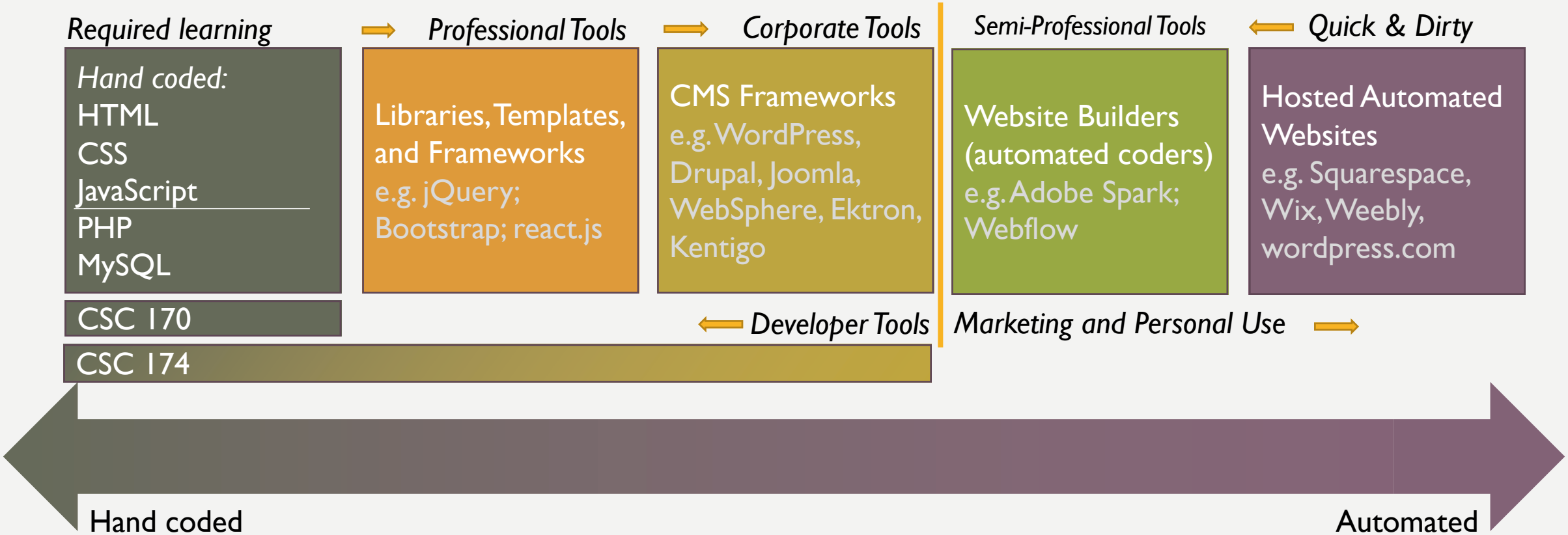
TEMPLATES, LIBRARIES, FRAMEWORKS

TOOLS OF THE TRADE

FRONT-END WEBSITE-TECHNOLOGIES CONTINUUM



LIBRARIES, TEMPLATES, FRAMEWORKS

Why use?

- Common structures – don't have to re-do – don't *reinvent the wheel*

Templates

- Single file – static HTML (e.g. HTML5 Boilerplate)
- HTML-in-JS: ready-made patterns (HTML forms and form processors)

Libraries

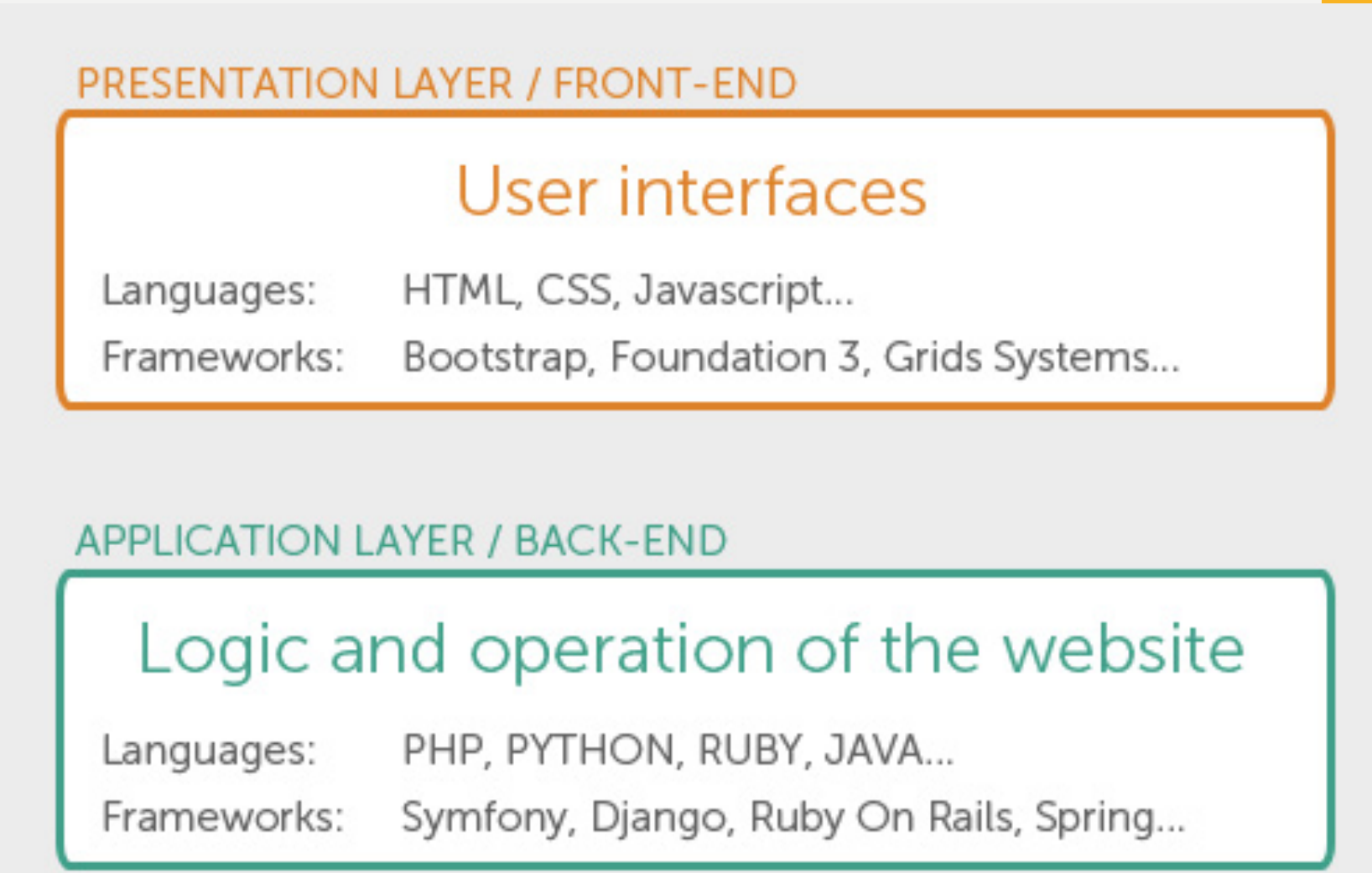
- Single file; for web usually JavaScript; e.g. jQuery
- Ready-made solutions – methods and functions

Frameworks

- Combination of HTML, CSS, and JavaScript; sometimes contains other libraries and templates
- Packages –ready for use and customization
- Front-end (mostly CSS and/or JS to control HTML) including application-focused frameworks

FRAMEWORKS

- Front-end
  - usually HTML, CSS and JavaScript
  - Grid solutions, CSS packages
  - Interesting example: www.muicss.com
- Back-end and Application
  - For application developers to build apps (software for the web)



…and react.js, angular.js, vue.js

ADVANTAGES AND DISADVANTAGES

Advantages

Speeds the mock-up process

Clean and tidy code

Solutions to common CSS problems

Browser compatibility

Learn good practices

Having a single procedure to resolve common problems makes maintaining various projects more straightforward.

Helpful in collaborative work

Disadvantages

Unused code leftover

Mixes content and presentation

Slower learning curve

You don't learn to do it yourself

When is it advisable to use a framework?

When customizing the framework, at what point do you realize it would have been faster or easier to code it yourself from the ground-up?

HOW TO OVERRIDE STYLES IN A CSS FRAMEWORK

After installing a **CSS Framework** (Bootstrap, Semantic UI, whatever)...	Create a new CSS file called **override.css** and LINK to it from your HTML webpages; install the LINK *below* all other CSS LINKs
	In a web browser, use **inspector tools** to find a property you want to change
	Using inspector tools, toggle and **play around with the property** until you get it looking the way you want
	In your override.css file, **write the selector** exactly like (or with more specificity than) the selector from the CSS Framework

Remember: the CSS Framework files ...no touchy!