HTML5 Boot Camp

The future of Web Apps is here! Are you ready for it? ;)



##### 

|  |  |
| --- | --- |
| **Author** | **Globant UI Practice** |
| Version | 2.0 |
| Document creation date | 2011-10-17 |
| Last modification date | 2013-04-08 |
| Last modification by | esteban.abait@globant.com |

Index

1. [Objective](#id.897be14cb359)
2. [Who Should Attend](#id.21dc7aafc174)
3. [Duration](#id.a73ab2f2ac11)
4. [Technical Assistance](#id.dd060dbb7eb5)
5. [Performance Measurement](#id.b5cf29d0ffa0)
6. [Handling advanced Developers](#id.0cc111c885a5)
7. [Materials](#id.f94cb75789dc)
8. [General Guidelines](#id.997a3aca11cc)
9. [Proposed Test Applications](#id.8b92757fa279)
10. Environment Setup
11. [Learning Days](#id.b50bdb532fdc)
    * [Topic 1: Javascript Intro and jQuery](#id.8dc70ccbbe50)
    * [Topic 2: jQuery Mobile](#id.2a30572bd13a)
    * [Topic 3: OOP and Inheritance in Javascript](#id.1351de2332d5)
    * [Topic 4: AMD + RequireJS](#id.857777062f13)
    * [Topic 5: Template Engines](#id.0f8ff196f7f0)
    * [Topic 6: MVC + Backbone](#id.1bf98acf8811)
    * [Topic 7: HTML5 Tags and CSS3](#id.1a41cc68fed7)
    * [Topic 8: HTML5 APIs](#id.b350a3595b11)
    * [Topic 9: Modernizr + yepNope](#id.b4608850fbe7)
12. [Your First Project Starts Now!](#id.605681dc8138)

Objective

This course teaches the basics of HTML5 for gaming and mobile apps development.

→ [index](#id.48b72f1065a3)

Who Should Attend

The training will start at a low level and does not require in depth knowledge of the platform in question. Desirable participant profile: trainees and outside Globant candidates.

→ [index](#id.48b72f1065a3)

Duration

Five weeks total.

Three weeks for guided learning and two weeks for app development. Given holidays that occur at the middle, the course will last six weeks in total.

→ [index](#id.48b72f1065a3)

Technical Assistance

You can contact other bootcamp participants or any available tutor if you need technical assistance. We will create one chat for boot camp members only and another one for boot camp members and tutors when boot camp starts.

Performance Measurement

1. Code review after each practice and sprint
2. Checkpoint completion after Learning stage with your assigned tutor

→ [index](#id.48b72f1065a3)

Handling Advanced Developers

Developers that move faster than average can complete tasks from other sprints.

They can also entirely skip any of the Learning Days which contents they already master and concentrate on completing user stories.

→ [index](#id.48b72f1065a3)

Materials

1. Any web browser with developer tools (use firebug for debugging in Firefox)
2. IDE of choice (we recommend [SublimeText 2](http://www.sublimetext.com/))
3. Shared Google docs
4. Skype Account + headset (audio calls)
5. Create your own [GitHub](https://github.com/) account. Follow this [guideline](https://help.github.com/articles/set-up-git) to setup your account.
6. Install your own LAMP server. Download [WAMP](http://www.wampserver.com/en/) if you are using Windows, or [XAMPP](http://www.apachefriends.org/en/xampp.html) in case you prefer Linux.
7. Create a new repository in [GitHub](https://github.com/) to host the project code.

→ [index](#id.48b72f1065a3)

General Guidelines

The boot camp is organized in the following way:

1. The first three weeks will be used for intensive guided self learning with reading and practices.
2. The next two weeks will be used to develop a test application looking to mimic real life situations and organization.
3. The project manager will coordinate learning days encouraging team communication in daily meetings.
4. The project manager will gather information regarding individual progress so we can look for alternative assistance where needed.
5. The project manager will lead the boot camp sprints as if it where a real project using SCRUM agile methodology.
6. Two boot camp chats will be created for feedback and technical assistance:
   1. Bootcamp HTML - ALL  
      Every person participating in the bootcamp is present here (students, tutors and PMs). Here is the place to ask for technical assistance!
   2. Bootcamp HTML - Assistants  
      Here you will reach just your boot camp fellows for asking question sharing knowledge.
7. Team play is encouraged but the work will be evaluated per person.
8. Sprint duration will be 1 week.
9. The instructions will be vague as they generally are in real life projects. You must look for support and guidance from your PM, team mates and tutors.
10. All code and documentation must be in English.

→ [index](#id.48b72f1065a3)

Proposed Test Applications

-

Learning Days

Each day you will grab the fundamentals of the key building blocks of the next generation mobile apps; yeah, web apps! Web apps injected with the latest and coolest toolkits and techniques.

On each learning day you will have:

1. **Reading**:  
   We will provide you with documentation related with current sprint content so you can have a background reference, guide and examples to complete the following practice.
2. **Practice**:  
   You will implement the previously gathered knowledge in simple coding activities.  
   Most important task numbers are listed in the “*Key Points*” section for each day and they should get most of your attention; if you feel you don’t have enough time to complete all tasks, start with these ones when possible.
3. **Commit:**You will commit all your code on a daily basis, when you finish your practice.

**Topic 1: Javascript Intro and jQuery (1 day)**

**Reading:**

1. Beginners: Eloquent Javascript [basic tutorial](http://eloquentjavascript.net/) (in case you need it!)
2. Beginners: Have fun with [Codecademy](http://www.codecademy.com/courses/jquery-and-the-dom) (more experienced devs might find it fun!).
3. Recommended: [jQuery Fundamentals](http://jqfundamentals.com/book/index.html)
4. Explore [jQuery documentation](http://docs.jquery.com/Main_Page)
5. Javascript Style Guide: <https://github.com/rwldrn/idiomatic.js>
6. Javascript Prototypes: <http://www.slideshare.net/Dmitry.Baranovskiy/demystifying-prototypes-6183470>

**Extra documentation:**

* Web Platform Documentation Project: <http://www.webplatform.org/>
* MDN Javascript Reference: <https://developer.mozilla.org/en/JavaScript/Reference>
* Annotated ECMAScript 5.1: <http://es5.github.com/>

**Practice:**

1. Add jQuery to this code: <http://pastebin.com/pSysQTV6> (download it or create a new index.html file and put that code inside)
2. Add a button to your index.html.
3. Show an alert when page has finished loading.
4. Add a textbox with the id “alias”, and put the cursor inside it right after alert closes.
5. Attach an event to the created button which calls a function that gets a response from <http://bootcamp.aws.af.cm/welcome/tunombre>
6. Write the response to a div element.
7. Show div content in red when a server error occurs.
8. Take some free air and then create a function to highlight your name in the server response content. Call it right after setting the response inside the div.
9. AJAX: get the response from <http://search.twitter.com/search.json>[[1]](#footnote-0) with parameter data "q=html5"  
   First show response in firebug’s console log to analyze data, then display tweets inside a floated div in the middle of the screen.  
   For each tweet show: from\_user, text, created\_at, profile\_image\_url.
10. Add a transparent background covering the whole screen behind the tweets div so the background gets dimmed.

**Key Points:**

5, 6, 7

**Mobile Test:**

Test your code in a mobile device or in Android emulator.

**Commit:**

Commit your practice code.

→ [index](#id.48b72f1065a3)

**Topic 2: jQuery Mobile (2 days)**

**Reading:**

Explore [jQuery Mobile documentation](http://jquerymobile.com/demos/1.1.0-rc.1/docs/about/getting-started.html)

**Practice:**

1. Copy your code from the first day to current topic folder.
2. Add jQuery Mobile to your app.
3. Set viewport.
4. Add proper data roles.
5. Show a loading animation when previously created button is clicked.
6. Hide animation when ESC key is pressed.
7. Add three pages and a menu to navigate between them.
8. Play with different transitions between pages.
9. Add some content to your pages and buttons that opens dialogs.
10. Create a new file pages/sample1.html from the code in <http://pastebin.com/VGZYCbS2>. Load its content in a new page (it’s just one link)  
    useful info: [jQuery Mobile linking pages](http://www.google.com/url?q=http%3A%2F%2Fjquerymobile.com%2Ftest%2Fdocs%2Fpages%2Fpage-links.html&sa=D&sntz=1&usg=AFQjCNG6HCD3dhtVr-RfDn7Nusyvwgvfag)
11. Translate your movie.getTop listing into a jQuery Mobile list with thumbs.
12. Show a dialog with more movie details and bigger box art when a movie is clicked.

**Key Points:**

3, 7, 10

**Mobile Test:**

Test your code in a mobile device or in Android emulator (feel the magic of viewport and jQuery Mobile!).

**Commit:**

Commit your practice code.

→ [index](#id.48b72f1065a3)

**Topic 3: Design Patterns and OOP in Javascript (3 days)**

**Reading:**

1. Understand Javascript Prototypes: <http://javascriptweblog.wordpress.com/2010/06/07/understanding-javascript-prototypes/>
2. Understand Javascript OO: <http://killdream.github.com/blog/2011/10/understanding-javascript-oop/index.html>
3. Read [this](http://addyosmani.com/resources/essentialjsdesignpatterns/book/) article about Javascript design patterns by Addy Osmani

**Extra documentation:**

* [Javascript Patterns Collection](http://shichuan.github.com/javascript-patterns/)
* [A fresh look at JavaScript Mixins](http://www.google.com/url?q=http%3A%2F%2Fjavascriptweblog.wordpress.com%2F2011%2F05%2F31%2Fa-fresh-look-at-javascript-mixins%2F&sa=D&sntz=1&usg=AFQjCNG2JuQos_dZy63pUOC7fCIwhtirZA)

**Practice:**

1. Create a Movie object:

|  |
| --- |
| Movie |
| -attributes : hashmap |
| + play()  + stop()  + set(attr:string, value)  + get(attr:string) |

1. Instantiate some of your favourite movies and play with them in the console.
2. Add a MovieObserver class that listens for “*playing*” and “*stopped*” events.
3. Publish “*playing*” event on Movie.play().  
   You should be able to do something like this in the console:  
   var terminator = new Movie();  
   terminator.set('title', 'Terminator');  
   ...  
   terminator.play(); //output: Playing Terminator...
4. Publish “*stopped*” event on Movie.stop().
5. Log to console when each event is fired.
6. Refactor Movie class as a Module keeping your previous code for reference.
7. Create a DownloadableMovie that extends from Movie adding a download method.
8. Create a mixin object called Social with the methods: share(friendName) and like().
9. Apply the mixin to Movie object and play with the console output.  
   You should be able to do something like this in the console:  
   var ironman2 = new Movie();  
   ironman2.set('title', 'Iron Man 2');  
   ...  
   ironman2.share(‘V. Rivas’); //output: Sharing Iron Man 2 with V. Rivas
10. Create an Actor class and create some actors from one of your favorite movies.
11. Show how you would add an array of actors to a Movie object.

**Key Points:**

3, 4, 8, 10

**Commit:**

Commit your practice code.

→ [index](#id.48b72f1065a3)

**Topic 4: AMD + RequireJS (1 day)**

**Reading:**

1. Read about [AMD paradigm](http://unscriptable.com/code/Using-AMD-loaders)
2. Read about [module pattern limitations](http://snook.ca/archives/javascript/no-love-for-module-pattern)
3. Read [RequireJS documentation](http://requirejs.org/) and samples

Extra documentation:

* [AMD: The Definitive Source](http://www.sitepen.com/blog/2012/06/25/amd-the-definitive-source/)
* [Writing Modular Javascript](http://addyosmani.com/blog/writing-modular-javascript/)
* [What are the use cases for RequireJS vs. Yepnope vs. LABjs?](http://www.quora.com/What-are-the-use-cases-for-RequireJS-vs-Yepnope-vs-LABjs) (Quora)

**Practice:**

1. Create a new index.html file as you did it the first day.
2. Add RequireJS .
3. Create the same Movie class as in the previous practice, but inside a module; the module will be an external file.
4. Create a Director class inside a module and set it as a dependency on the Movie module.
5. Add Director name:string, and a quotes:array properties, and a speak() methods; calling speak() will return director’s quotes.
6. Add a Director to a Movie  
   You should be able to do something like this in the console:  
   var alien = new Movie();  
   ...  
   var ridleyScott = new Director(‘Ridley Scott’); //sets name in constructor  
   ridleyScott.set('quotes', ['Cast is everything.', 'Do what ...']);

alien.set('director', ridleyScott);  
 alien.get('director').speak(); //output: Ridley Scott says: 'Cast is...'

1. Add jQuery as a module (hint: use the [shim](http://requirejs.org/docs/api.html#config-shim) support).
2. Add jQuery Mobile as a module (hint: use the [shim](http://requirejs.org/docs/api.html#config-shim) support).
3. Make Movie’s Director speak random quotes inside jQuery Mobile dialogs.

**Key Points:**

2, 3

**Commit:**

Commit your practice code.

→ [index](#id.48b72f1065a3)

**Topic 5: Template Engines (1 day)**

**Reading:**

1. Know [Handlebars](http://handlebarsjs.com/).
2. Know [Underscore](http://documentcloud.github.com/underscore/) template parsing capabilities.
3. Know [{dust}](http://www.google.com/url?q=http%3A%2F%2Fakdubya.github.com%2Fdustjs%2F&sa=D&sntz=1&usg=AFQjCNHxT9p9k0f-C7NBell-5HYVPgpHaw).

**Extra documentation:**

* Client-side template engine comparison: <http://engineering.linkedin.com/frontend/client-side-templating-throwdown-mustache-handlebars-dustjs-and-more>

**Practice:**

1. Create a new index.html file and link the three template engines. The three files must be linked from a libs directory.
2. Write a template for a professional profile (linkedIn alike but simpler).
3. Create a JSON file with your professional profile data.
4. When document finishes loading parse the template with your data.
5. If you didn’t add work experience to the JSON file do it now using an array.
6. Adjust your template and parse it using template engine facilities for iteration.
7. Repeat with the next template engine from the reading list.

**Key Points:**

4, 6

**Mobile Test:**

Test your code in a mobile device or in Android emulator.

If it does not look good fix it with minimal changes.

**Commit:**

Commit your practice code.

→ [index](#id.48b72f1065a3)

**Topic 6: MVC & Mixins (2 days)**

**Reading:**

1. Read about the [MVC design pattern](http://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller)
2. [MVC Architecture for Javascript Applications](http://michaux.ca/articles/mvc-architecture-for-javascript-applications)
3. Know [Backbone.js](http://documentcloud.github.com/backbone/); <https://github.com/addyosmani/backbone-fundamentals>

**Practice:**

1. Using Backbone, create a movie listing with your favorite movies.
2. Show movie details in a details view.
3. Allow to add / edit / remove movies from the list.

**Key Points:**

1

**Commit:**

Commit your practice code.

→ [index](#id.48b72f1065a3)

**Topic 7: HTML5 Tags and CSS3 (1 day)**

**Reading:**

1. Read about [new HTML tags](http://diveintohtml5.info/semantics.html).
2. Read about the basics of CSS: <http://learn.shayhowe.com/html-css/>
   1. **Make sure you understand the** [**CSS's Box Model**](http://learn.shayhowe.com/html-css/box-model)
      1. <http://reference.sitepoint.com/css/boxmodel> (optional)
      2. <https://developer.mozilla.org/en-US/docs/CSS/box_model> (optional)
3. Responsive Web Design
   1. Read this [great article about responsive design](http://www.alistapart.com/articles/responsive-web-design/).
   2. Read about *Layout Patterns*: <http://www.lukew.com/ff/entry.asp?1514>
   3. Look at this collection of layout patterns: <http://www.maxdesign.com.au/articles/css-layouts/>
4. Check out examples of responsive web design: <http://mediaqueri.es/>
5. If you feel strong about CSS, then go pro: <http://learn.shayhowe.com/advanced-html-css/>

Suggested readings:  
[CSS Specificity: Things You Should Know](http://www.google.com/url?q=http%3A%2F%2Fcoding.smashingmagazine.com%2F2007%2F07%2F27%2Fcss-specificity-things-you-should-know%2F&sa=D&sntz=1&usg=AFQjCNG3MN8l-eWjjx4yhsrtp8Y0TkAVSg)

**Practice:**

1. Create a basic page using the new HTML5 tags. That page must follow a fixed-desktop-layout.
2. From the *layout patterns* article, implement the following patterns:
   1. Mostly Fluid
   2. Column Drop
   3. Layout Shifter
   4. Off Canvas

**Key Points:**

1, 2

**Commit:**

Commit your practice code.

→ [index](#id.48b72f1065a3)

**Topic 8: HTML5 APIs (1day)**

**Reading:**

1. Discover the [new APIs](http://slides.html5rocks.com/#landing-slide)
2. [Read more](http://blog.frontendforce.com/2010/04/html5-javascript-api-whats-new/) about them.
3. [Play](http://playground.html5rocks.com/#localstorage) a bit with them.

Suggested readings:   
[Drag and Drop in desktop and mobile browsers](http://www.google.com/url?q=http%3A%2F%2Fcaniuse.com%2Fdragndrop&sa=D&sntz=1&usg=AFQjCNH7Vu-7BQkc2mv2khzX5QGmsMTUcA), [Native Drag and Drop](http://html5doctor.com/native-drag-and-drop/)

[GMaps Javascript API](https://developers.google.com/maps/documentation/javascript/tutorial?hl=es)

**Practice:**

1. Write a simple javascript code tester that will execute your javascript code from a textarea.
2. Add the option to persist your test code in the browser’s storage.
3. Add drag and drop support to load files.
4. Obtain geolocation and show it on a map.
5. Create a web worker which sends a message each 60 seconds.
6. Open a web socket and test it against [this echo service](http://websocket.org/echo.html).
7. Add a cache manifest to the application. Play a little with it and test the download speed improvement. Try changing cached content with and without changing the manifest.

**Key Points:**

2, 3, 7

**Commit:**

Commit your practice code.

→ [index](#id.48b72f1065a3)

**Topic 9: Modernizr + yepNope (1day)**

**Reading:**

Read this [great tutorial](http://www.sitepoint.com/regressive-enhancement-with-modernizr-and-yepnope/).

**Practice:**

1. Copy your code from topic **HTML5 APIs** and add Modernizr and yepNope to it.
2. Reuse your geolocation code adding a polyfill if not supported.
3. Reuse your file reader adding a polyfill if not supported.
4. Reuse your web worker adding a polyfill if not supported.
5. Reuse your web socket adding a polyfill if not supported.
6. Load your site only if media-queries are supported.
7. Try it in different browsers.

**Key Points:**

3, 6

**Commit:**

Commit your practice code.

→ [index](#id.48b72f1065a3)

Your First Project Starts Now!

It’s time for the real thing.

You will work on a project to achieve a fully working multi platform mobile app developed in Javascript and using the latest tags and APIs available in HTML5.

You will apply all the knowledge obtained during the learning weeks.

You’ll be given with a backlog of user stories you will estimate with your PM.

Once estimated, user stories will be divided into sprints of one week.

You will participate on daily scrum meetings.  
Will you be able to consult documentation? Of course! You will be able to use any resource you know that helps you complete your user stories, be it going back to documentation sites, tutorials or just googling what you need. Luckily the web is plenty of awesome resources waiting for you to grasp them!

**Commits**

You will commit your code to GitHub on a daily basis.

**Reviews**Tutors will give you feedback at the end of each sprint by reviewing and commenting your commited code in GitHub. If possible tutors will give you even more frequent feedback, sometimes at the end of the day.

→ [index](#id.48b72f1065a3)

…

Thanks for reading!

1. <https://dev.twitter.com/docs/api/1/get/search> [↑](#footnote-ref-0)