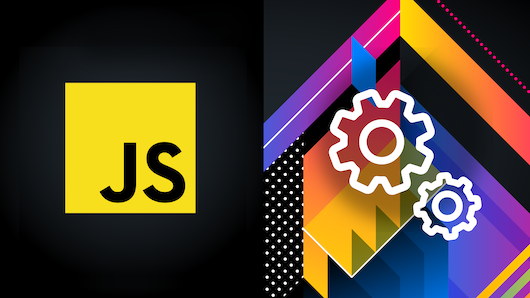
**W3Cx JS.0x - JavaScript Introduction**

Welcome!



Thank you for registering to this course where you'll learn JavaScript in the browser and hence interact a lot with HTML/CSS/standard APIs. This course is targeted to beginners who'd like to quickly have fun while bringing interactivity to HTML documents, writing small games, playing with multimedia, etc.

### Course outline

HTML5, CSS and JavaScript are the “classic three” for Web developers and designers. JavaScript lets you add interactive features to your Web sites, including dynamically updated content, controlled multimedia, animated images, and much more.

* In **Module 1**, we present the basics of JavaScript by the way of many examples. We also show how to use the browser devtools, a powerful, built-in set of tools that represent the Swiss army knife of any Web developer.
* In **Module 2**, you'll learn about conditional statements, logical operators, loops and how to detect events. We will introduce the DOM API and the selector API (for selecting elements in the DOM). At the end of this module, armed with this new knowledge, we'll write a small game together.
* **Module 3** is more "project oriented", and less focused on fundamental concepts. Discover the audio and video APIs, and other cool HTML5 APIs. You will play with sound samples and music, and this is going to be useful to put a music background and/or sound effects to the game you started to develop in Module 2!
* **Module 4** is about structuring data with a focus on JavaScript Object Oriented Programming. As with many other programming languages, JavaScript objects can be compared to real-life objects. Learn how to create your own classes and multiples objects from the same class. And experiment it all by doing one or more optional projects!
* **Module 5** is the last module of the course! Let's enhance interactivity of past projects by adding forms. To put into practice, let's develop a mini contact manager that load/save its data using the JSON format.

### Syllabus

1. Course information
   1. [Welcome to "JavaScript Introduction"](https://learning.edx.org/course/course-v1:W3Cx+JS.0x+2T2021/block-v1:W3Cx+JS.0x+2T2021+type@sequential+block@241b2b4bdcac485e90e727b3990c6600)
   2. [Course outline, grading and certificate](https://learning.edx.org/course/course-v1:W3Cx+JS.0x+2T2021/block-v1:W3Cx+JS.0x+2T2021+type@sequential+block@57f8bf03201047ab94c00a31ec455174)
   3. [Course forums](https://learning.edx.org/course/course-v1:W3Cx+JS.0x+2T2021/block-v1:W3Cx+JS.0x+2T2021+type@sequential+block@3a8cdd90c62f4062afc887562aa62037)
   4. [Course tools](https://learning.edx.org/course/course-v1:W3Cx+JS.0x+2T2021/block-v1:W3Cx+JS.0x+2T2021+type@sequential+block@24575cf0615e44a893953a5d19c4602e)
   5. [About W3C and the Web](https://learning.edx.org/course/course-v1:W3Cx+JS.0x+2T2021/block-v1:W3Cx+JS.0x+2T2021+type@sequential+block@b649399965e14ec3a9c5b988a4c57e9e)

## Module 1: Introduction to JavaScript

* 1. [1.1 Introduction - Module 1](https://learning.edx.org/course/course-v1:W3Cx+JS.0x+2T2021/block-v1:W3Cx+JS.0x+2T2021+type@sequential+block@2623bb73553449888f527253dfbbcb97)
  2. [1.2 JavaScript, HTML and CSS](https://learning.edx.org/course/course-v1:W3Cx+JS.0x+2T2021/block-v1:W3Cx+JS.0x+2T2021+type@sequential+block@19e689263977480bb5cbd6ac632e28c0)
  3. [1.3 JavaScript overview](https://learning.edx.org/course/course-v1:W3Cx+JS.0x+2T2021/block-v1:W3Cx+JS.0x+2T2021+type@sequential+block@070bf35233d745279eb58abfe3ded2d6)
  4. [1.4 Your first HTML/CSS/JS page](https://learning.edx.org/course/course-v1:W3Cx+JS.0x+2T2021/block-v1:W3Cx+JS.0x+2T2021+type@sequential+block@d160be87d1ed41299477cce607277c90)
  5. [1.5 Variables, values, functions, operators and expressions](https://learning.edx.org/course/course-v1:W3Cx+JS.0x+2T2021/block-v1:W3Cx+JS.0x+2T2021+type@sequential+block@1e5ef862fa80431e82ec1a17bc4e193b)
  6. [1.6 Simple JavaScript examples to play with](https://learning.edx.org/course/course-v1:W3Cx+JS.0x+2T2021/block-v1:W3Cx+JS.0x+2T2021+type@sequential+block@0c071e4b3e8040c5987afbe506249a24)

## Module 2: Adding interactivity to HTML documents

* 1. [2.1 Introduction - Module 2](https://learning.edx.org/course/course-v1:W3Cx+JS.0x+2T2021/block-v1:W3Cx+JS.0x+2T2021+type@sequential+block@24b431bc835147c8ab938e6305d4f62f)
  2. [2.2 Conditional statements, loops and logical operators](https://learning.edx.org/course/course-v1:W3Cx+JS.0x+2T2021/block-v1:W3Cx+JS.0x+2T2021+type@sequential+block@f6cf675cb0b04aa3a6d20ea5172c3ca7)
  3. [2.3 Functions (part2): callbacks](https://learning.edx.org/course/course-v1:W3Cx+JS.0x+2T2021/block-v1:W3Cx+JS.0x+2T2021+type@sequential+block@7382ad09ff3147e99bfc26ac3a871ad6)
  4. [2.4 Handling events](https://learning.edx.org/course/course-v1:W3Cx+JS.0x+2T2021/block-v1:W3Cx+JS.0x+2T2021+type@sequential+block@2b7536f2b69145d39b42668950ec5b05)
  5. [2.5 The DOM API](https://learning.edx.org/course/course-v1:W3Cx+JS.0x+2T2021/block-v1:W3Cx+JS.0x+2T2021+type@sequential+block@eb07da2208384bf393e3124d8806f347)
  6. [2.6 Let's write a small game](https://learning.edx.org/course/course-v1:W3Cx+JS.0x+2T2021/block-v1:W3Cx+JS.0x+2T2021+type@sequential+block@f9c0988a7a0846adbcd0b0e0578f0369)

## Module 3: Playing with HTML5 APIs

* 1. [3.1 Introduction - Module 3](https://learning.edx.org/course/course-v1:W3Cx+JS.0x+2T2021/block-v1:W3Cx+JS.0x+2T2021+type@sequential+block@f93740c2c2594149a62e6369fe68d28b)
  2. [3.2 Arrays (part 2): iterators](https://learning.edx.org/course/course-v1:W3Cx+JS.0x+2T2021/block-v1:W3Cx+JS.0x+2T2021+type@sequential+block@6fa3ab0ce4d34ae6b14f61cd28538de4)
  3. [3.3 HTML5 multimedia and JavaScript API](https://learning.edx.org/course/course-v1:W3Cx+JS.0x+2T2021/block-v1:W3Cx+JS.0x+2T2021+type@sequential+block@63293f0b87b24340b3b18667c89e644e)
  4. [3.4 Displaying a map with the geolocation API](https://learning.edx.org/course/course-v1:W3Cx+JS.0x+2T2021/block-v1:W3Cx+JS.0x+2T2021+type@sequential+block@2b58c0a5fc1543a3a7cca1d37099ef33)
  5. [3.5 Playing sound samples and music](https://learning.edx.org/course/course-v1:W3Cx+JS.0x+2T2021/block-v1:W3Cx+JS.0x+2T2021+type@sequential+block@aab15657b0d6482185c723dee89037c4)

## Module 4: Structuring data

* 1. [4.1 Introduction - Module 4](https://learning.edx.org/course/course-v1:W3Cx+JS.0x+2T2021/block-v1:W3Cx+JS.0x+2T2021+type@sequential+block@0cb22ada6bc74ecb8bb59c06ed5d5c74)
  2. [4.2 Objects (part 2): properties and methods](https://learning.edx.org/course/course-v1:W3Cx+JS.0x+2T2021/block-v1:W3Cx+JS.0x+2T2021+type@sequential+block@9c8aeb657b10443893ad9d08f37a0870)
  3. [4.3 Objects (part 3): creating multiple objects](https://learning.edx.org/course/course-v1:W3Cx+JS.0x+2T2021/block-v1:W3Cx+JS.0x+2T2021+type@sequential+block@c33324ad26584d6391466a8966b74abb)
  4. [4.4 Improving the game with classes](https://learning.edx.org/course/course-v1:W3Cx+JS.0x+2T2021/block-v1:W3Cx+JS.0x+2T2021+type@sequential+block@06a98acbdc884a89bad2791f4b90c6fb)

## Module 5: Working with forms

* 1. [5.1 Introduction - Module 5](https://learning.edx.org/course/course-v1:W3Cx+JS.0x+2T2021/block-v1:W3Cx+JS.0x+2T2021+type@sequential+block@3eddff40b8db4408b65c81178a98a785)
  2. [5.2 Built-in JavaScript objects](https://learning.edx.org/course/course-v1:W3Cx+JS.0x+2T2021/block-v1:W3Cx+JS.0x+2T2021+type@sequential+block@a32962a1e62848d890bc9a0d1599824e)
  3. [5.3 HTML5 tables, forms and input fields](https://learning.edx.org/course/course-v1:W3Cx+JS.0x+2T2021/block-v1:W3Cx+JS.0x+2T2021+type@sequential+block@3fd025df64df4a5fa2818efe2f1c689d)
  4. [5.4 The JSON notation](https://learning.edx.org/course/course-v1:W3Cx+JS.0x+2T2021/block-v1:W3Cx+JS.0x+2T2021+type@sequential+block@ed3ab9994af649e6a1e40ba6d80a8ff5)
  5. [5.5 A small application](https://learning.edx.org/course/course-v1:W3Cx+JS.0x+2T2021/block-v1:W3Cx+JS.0x+2T2021+type@sequential+block@f075d51203fd492ca991a4cf9e944acb)

This course is designed for anyone who is comfortable with programming concepts. No prerequisites are required though we encourage you to follow these two other W3Cx courses: [HTML5 and CSS Fundamentals](https://www.edx.org/course/html5-and-css-fundamentals) and [CSS Basics](https://www.edx.org/course/css-basics).

#### During this course, you will learn:

* How to add JavaScript code in your Web site/Web app, and how to debug it.
* How to make interactive Web sites through the DOM API.
* How to change the CSS styles of HTML5 elements from JavaScript.
* How to deal with HTML5 forms.
* How to make basic graphics and animations using the HTML5 canvas.

You will make good use of your JavaScript skills in the other 2 W3Cx courses included in the ["Front-End Web Developer" W3Cx Professional Certificate](https://www.edx.org/professional-certificate/w3cx-front-end-web-developer) program: [HTML5 Coding Essentials and Best Practices](https://www.edx.org/course/html5-coding-essentials-and-best-practices) and [HTML5 Apps and Games](https://www.edx.org/course/html5-apps-and-games).

In the meantime, have fun in this course!

#### Web editors

**Foreword:**

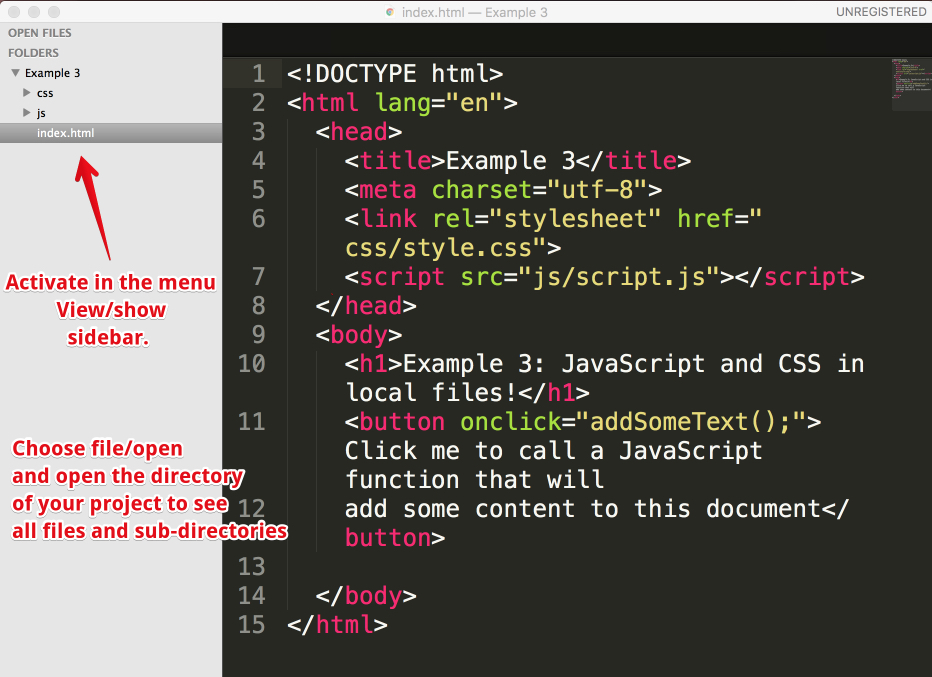
Michel Buffa, author of this course:  
  
"When I work locally, with files located on my computer hard drive, I use the Sublime Text source code editor, Visual Studio Code, WebStorm, NetBeans, depending on the size of the project I'm working on.  
- For testing simple examples, I mainly use the JsBin.com online code editor.  
- For embedding online examples in this course, I use the CodePen.io online code editor.  
- For choosing a CSS color, shadow, text-shadow, gradients, etc. I use the LiveWeave online code editor."

### Traditional source code editors

You can use any source code editor that has good support for HTML, CSS and JavaScript files. For this course, you are free to use whichever you prefer. However, there are some in particular that we recommend.

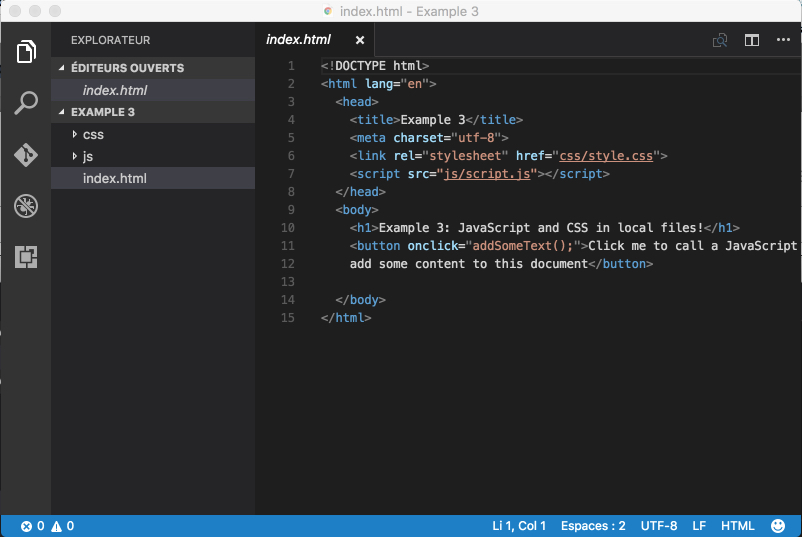
#### Sublime Text logo

[Sublime Text](https://www.sublimetext.com/) is a very powerful, multi-platform source code editor, it's semi-free (you can use it without paying, it will pop up a dialog asking you to buy it once in a while, but not very often). Sublime text supports hundreds of plugins to enhance its features.





[Visual Studio Code](https://code.visualstudio.com/) is a free, open source, multi-platform editor by Microsoft.



#### Other tools

#### Free of charge:

* + [Atom source](https://atom.io/) code editor (note that Visual Studio code is based on Atom).
  + [Brackets source](http://brackets.io/) code editor.
  + [NotePad++](https://notepad-plus-plus.org/) (Windows only)
  + [NetBeans](https://netbeans.org/) and [Eclipse](https://eclipse.org/): very powerful IDEs (integrated development environments), but heavier than all the "lightweight" source editors that we've talked about so far. More dedicated to "mid-size/large-size projects", more for pro developers that are also looking for good support for server-side languages such as Java, Python, PHP, etc.

#### Not free of charge:

* + [IDEs by JetBrains.com](https://www.jetbrains.com/) have a very good reputation and can be obtained for free if you are an academic customer (student or teacher). The [WebStorm](https://www.jetbrains.com/webstorm) IDE is a very good mid-weight tool for developing HTML/CSS/JS/NodeJS code.
  + [BBedit](https://www.barebones.com/products/bbedit/) (for Macs): source code editor for mac with support for Web languages.

#### Online editors/IDEs

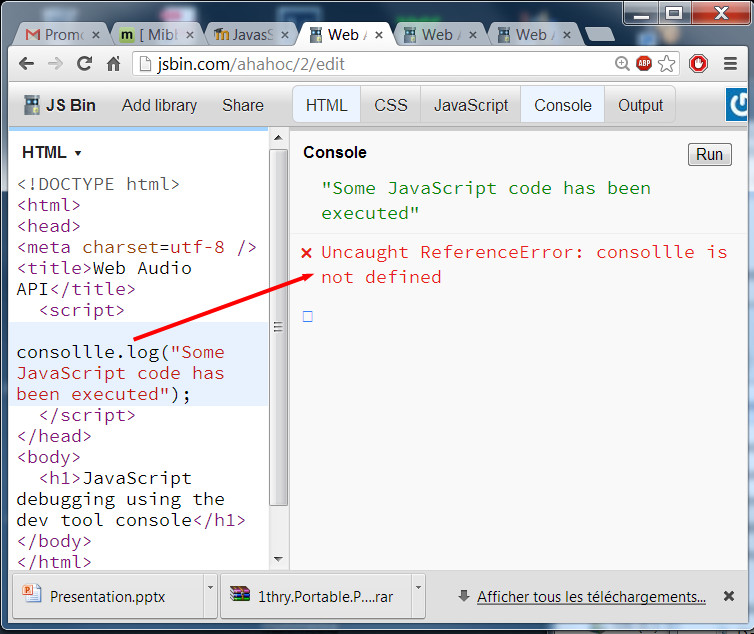
To help you practice for the duration of the course, you will use the following tools:

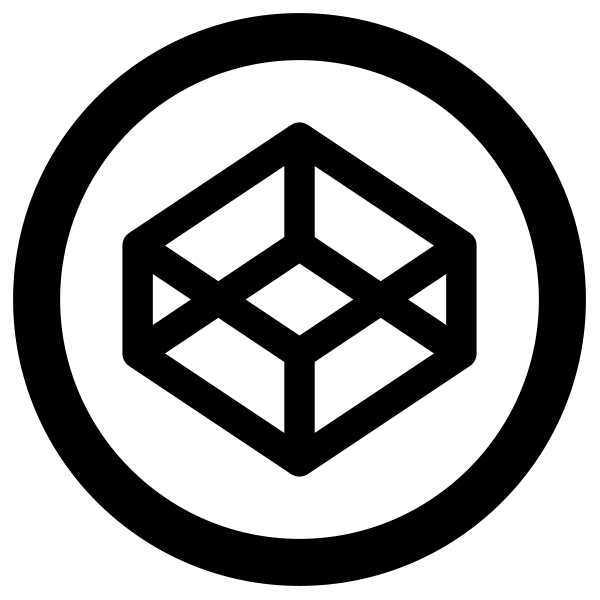


[JS Bin](https://jsbin.com/) is an open source collaborative Web development debugging tool. Most of the examples you will find in this course are either on JsBin or on CodePen.

Tutorials can be found on the Web (such as [these ones](https://www.youtube.com/playlist?list=PLXmT1r4krsTooRDWOrIu23P3SEZ3luIUq)) or on YouTube. The tool is really simple: just open the link to the provided examples, look at the code, look at the result, etc. And you can modify the examples as you like, you can also modify / clone / save / share them. Keep in mind that it's always better to be logged in (it's free) if you do not want to lose your work.

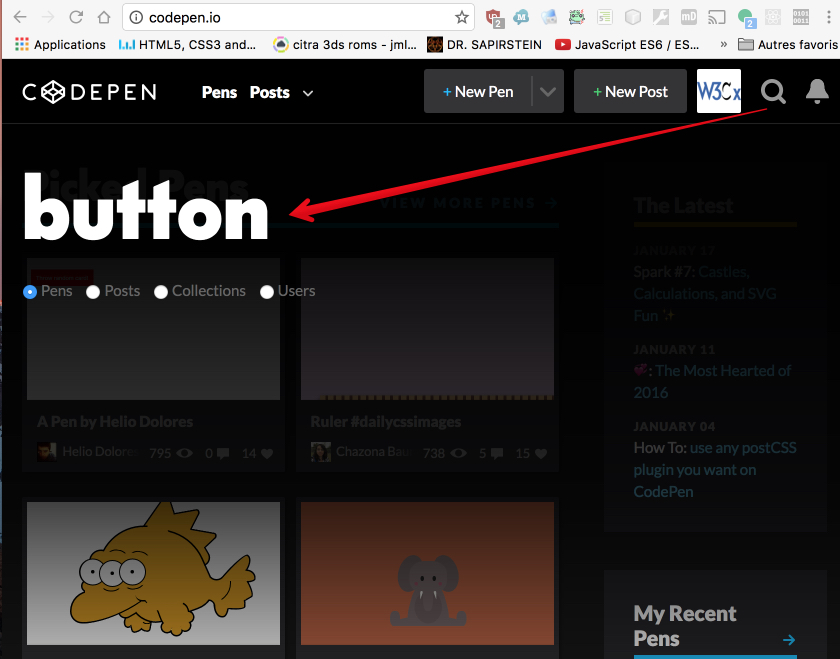
In our opinion, JsBin is the best online IDE for "live coding": typing and seeing what you are doing in real time, monitoring error messages in the console tab, and debugging your code. We will mainly use this tool for the live coding videos.

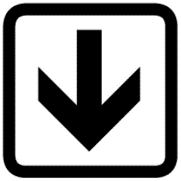




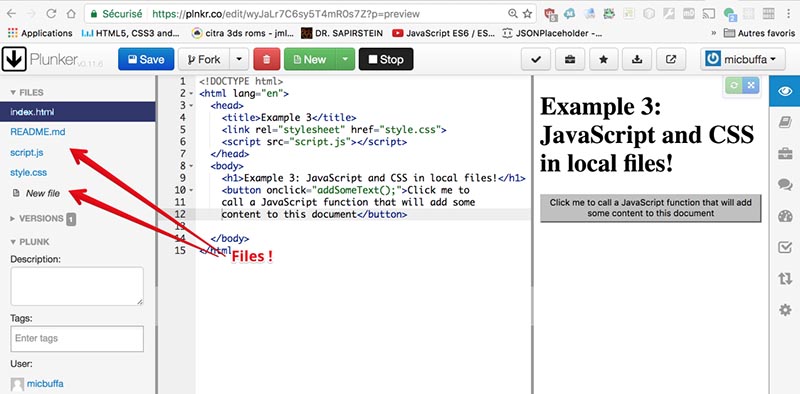
[**CodePen**](https://codepen.io/) is similar to JSBin except that its Web site includes a search engine, which is very useful for finding out what others  have developed. Looking for a nice HTML5/CSS button style? Just search for "button", etc. It's also easier for us to embed HTML/CSS/JS examples in this course with CodePen than with other online IDEs; this is why so many "pens" are embedded in the course pages.

This is a great service to get you started quickly as it doesn't require you to download anything and you can access it, along with your saved projects from any Web browser. Here's an article which will be of-interest if you use CodePen: [Things you can do with CodePen](https://codepen.io/brentmiller/post/things-you-can-do-with-codepen) [Brent Miller, February 6, 2019].



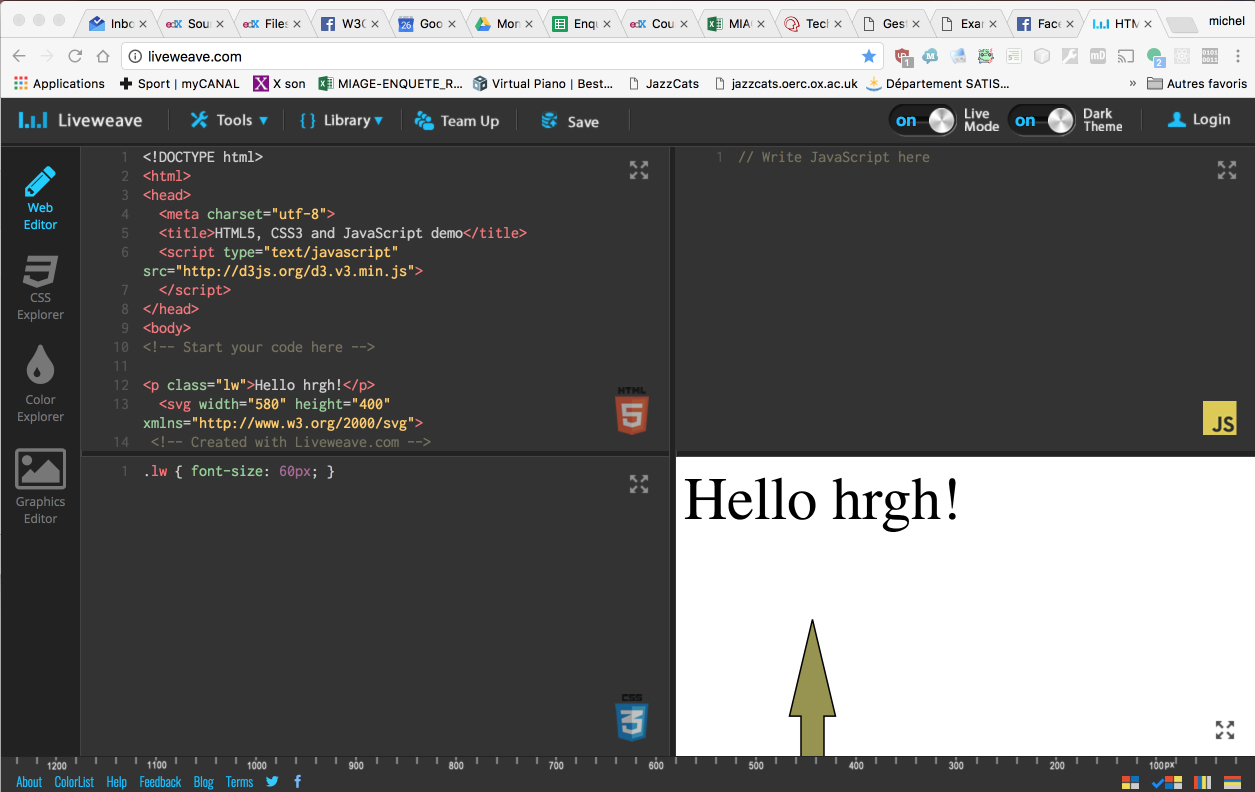


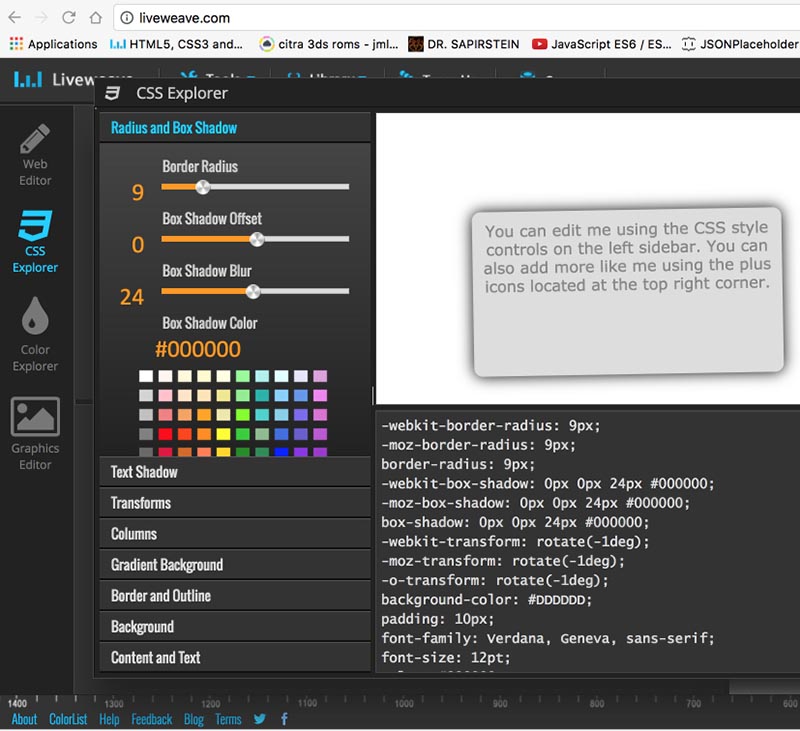
[Plunker](https://plnkr.co/) allows us to work online with separate files. So when we have no choice but to use separate files, we might use this tool.





[LiveWeave](https://liveweave.com/) is great for writing CSS code or for embedding SVG Graphics in an HTML document, as it includes online wizards and interactive editors. We use it when we have problems with CSS shadows, CSS colors or gradients, or when we want to include an SVG arrow in a document.







[JsFiddle](https://jsfiddle.net/) is very similar to JsBin and CodePen in terms of features.

#### Other tools

There are many other online IDEs and new ones appear each year. If you want a real, heavyweight online IDE that has nearly all the features offered by "big IDEs" such as Eclipse, NetBeans and WebStorm, take a look at the [Cloud9 IDE](https://c9.io/). It's free and will enable you to develop huge projects, that can include many files, it supports uploaded assets such as images, videos and sound files.  Furthermore, like Google Docs, it will support multiple users working at the same time on the same project, even on the same file. It's a real collaborative environment.

Michel Buffa, author of this course, developed a whole multitrack audio player this way. This application is available [online](https://mainline.i3s.unice.fr/). See also screenshots below:



100% of the development was done in a Web browser, by Michel Buffa and two friends, using the c9.io (Cloud 9) IDE (to see if online IDEs were a valuable approach):

#### JavaScript debuggers

Here is a selection of tools to help debug JavaScript code. The instructor will indicate other tools in module 1 of the course.

* [Firefox JS debugger](https://developer.mozilla.org/en-US/docs/Tools/Debugger) (debugger shipped inside Firefox) - all other browsers have integrated debuggers as well.
* [JS Lint](https://www.jslint.com/)  - The JavaScript Code Quality Tool
* [CodeBeautify](https://codebeautify.org/jsvalidate)  - JavaScript Validator
* Check also other[JavaScript debugging tools](https://jqueryhouse.com/best-javascript-debugging-tools/)

#### Browser compatibility

The term browser compatibility refers to the ability of a given Web site to appear fully functional on the browsers available in the market.

The most powerful aspect of the Web is what makes it so challenging to build for: its universality. When you create a Web site, you’re writing code that needs to be understood by many different browsers on different devices and operating systems!

To make the Web evolve in a sane and sustainable way for both users and developers, browser vendors work together to standardize new features, whether it’s a new [HTML element](https://developer.mozilla.org/en-US/docs/Web/HTML/Element), [CSS property](https://developer.mozilla.org/en-US/docs/Web/CSS/Reference#Keyword_index), or [JavaScript API](https://developer.mozilla.org/en-US/docs/Web/API). But different vendors have different priorities, resources, and release cycles — so it’s very unlikely that a new feature will land on all the major browsers at once. As a Web developer, this is something you must consider if you’re relying on a feature to build your site.

We are then providing references to the browser support of HTML5 features presented in this course using 2 resources: [Can I Use](https://caniuse.com/) and [Mozilla Developer Network (MDN) Web Docs](https://developer.mozilla.org/en-US/).

#### Can I use

[Can I Use](https://caniuse.com/) provides up-to-date tables for support of front-end Web technologies on desktop and mobile Web browsers. Below is a snapshot of what information is given by CanIUse when searching for "CSS3 colors".



#### MDN Web Docs



To help developers make these decisions consciously rather than accidentally, [MDN Web Docs](https://developer.mozilla.org/) provides browser compatibility tables in its documentation pages, so that when looking up a feature you’re considering for your project, you know exactly which browsers will support it.

#### External resources

* [MDN browser compatibility data: Taking the guesswork out of web compatibility](https://hacks.mozilla.org/2018/02/mdn-browser-compatibility-data/)
* [Caniuse and MDN compatibility data collaboration](https://hacks.mozilla.org/2019/09/caniuse-and-mdn-compat-data-collaboration/)

#### W3C validators

For over 20 years, the W3C has been developing and hosting [**free** and **open source** tools](https://w3c.github.io/developers/) used every day by **millions of Web developers and Web designers**. All the tools listed below are Web-based, and are available as downloadable sources or as free services on the [W3C Developers tools](https://w3c.github.io/developers/tools/)site.

#### W3C Validator

The [W3C validator](https://validator.w3.org/) checks the [markup validity](https://validator.w3.org/docs/help.html#validation_basics) of various Web document formats, such as HTML.

#### CSS Validator

The [CSS validator](https://jigsaw.w3.org/css-validator/) checks Cascading Style Sheets (CSS) and (X)HTML documents that use CSS stylesheets.



#### Unicorn

[Unicorn](https://validator.w3.org/unicorn/) is W3C's unified validator, which helps people improve the quality of their Web pages by performing a variety of checks. Unicorn gathers the results of the popular HTML and CSS validators, as well as other useful services, such as RSS/Atom feeds and http headers.

#### Link Checker

The [W3C Link Checker](https://validator.w3.org/checklink) looks for issues in links, anchors and referenced objects in a Web page, CSS style sheet, or recursively on a whole Web site. For best results, it is recommended to first ensure that the documents checked use valid [HTML Markup](https://validator.w3.org/) and [CSS](https://jigsaw.w3.org/css-validator/).

#### Internationalization Checker

The [W3C Internationalization Checker](https://validator.w3.org/i18n-checker/) provides information about various internationalization-related aspects of your page, including the HTTP headers that affect it. It also reports a number of issues and offers advice about how to resolve them.

#### W3C cheatsheet

<https://dev.w3.org/2009/cheatsheet/doc/>

The [W3C cheatsheet](https://www.w3.org/2009/cheatsheet/) provides quick access to useful information from a variety of specifications published by W3C. It aims at giving in a very compact and mobile-friendly format a compilation of useful knowledge extracted from W3C specifications, completed by summaries of guidelines developed at W3C, in particular Web accessibility guidelines, the Mobile Web Best Practices, and a number of internationalization tips.

[](http://www.w3.org/2009/cheatsheet/)

Its main feature is a lookup search box, where one can start typing a keyword and get a list of matching properties/elements/attributes/functions in the above-mentioned specifications, and further details on those when selecting the one of interest.

The W3C cheatsheet is only available as a [pure Web application](https://dev.w3.org/2009/cheatsheet/doc/).

## First steps in Web accessibility