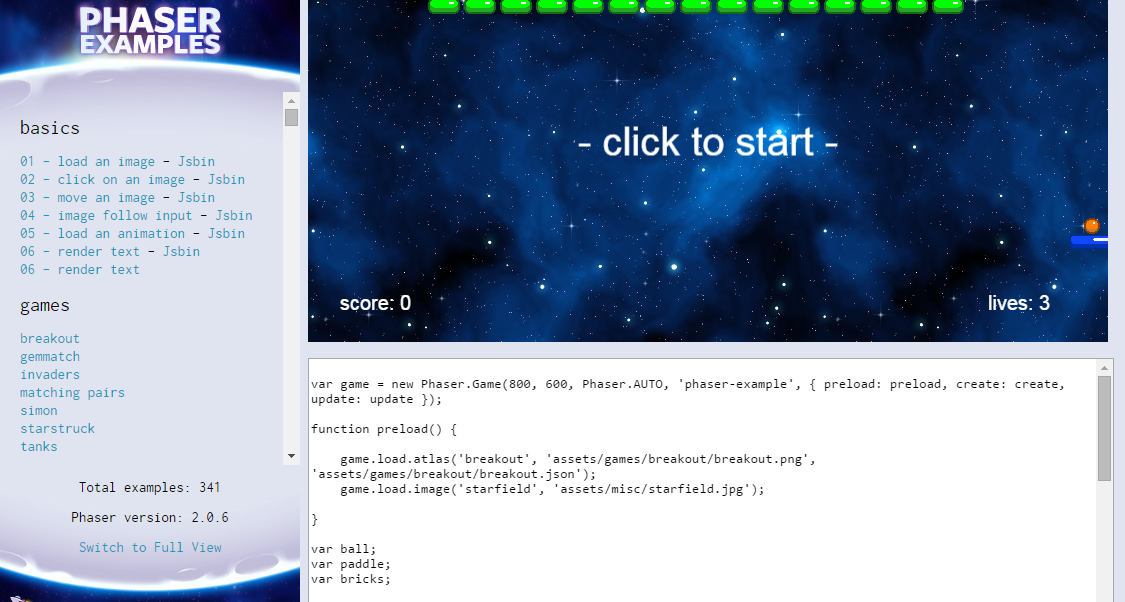
Advanced:

Beyond Scratch - writing games for (almost) any device using JavaScript with Phaser



Contents

[Getting started 2](#_Toc410464997)

[Connect to the internet 2](#_Toc410464998)

[Work with JSbin to begin with 2](#_Toc410464999)

[Open examples in JSbin 2](#_Toc410465000)

[[Optional] Log in to JSbin 2](#_Toc410465001)

[Your first edits 3](#_Toc410465002)

[Repeat the above for the other examples with JSbin links 4](#_Toc410465003)

[Get ready to create your own games outside JSbin 4](#_Toc410465004)

[Build your first game outside JSbin 4](#_Toc410465005)

[Deploying games on the web free, using Google Drive 5](#_Toc410465006)

[Deploy your first game on your own website 6](#_Toc410465007)

[Copy a game from the Phaser Examples and deploy it to your site 7](#_Toc410465008)

[Fancy copying another game? Try Flappy Bird 7](#_Toc410465009)

[Take your JavaScript and Phaser skills to game authoring level 7](#_Toc410465010)

# Getting started

If you are ready to move on from Scratch and you are also willing to invest some time to become a really good games coder, read on.

There are lots of places you could start on this: the following will give you a taster and you can check out additional resources when you want to go further. You can do most or all of this at home, or in the Dojo; you will need reliable internet access for much of it.

We suggest that you work through this document completing each part before moving to the next. But you can skip sections if you like and jump in where suits you. Don’t expect your mentors to be able to help you debug your code during the Dojo. Debugging will be mainly your responsibility and you will need to invest time in this at home.

Why write all your own code when someone else has already done a lot of the work, free? For that reason we will make use of the Phaser open source library for JavaScript game writing. Phaser has a lot of built-in ‘physics’ and game features. When you become really good at this, maybe you will even be able to contribute to and improve Phaser.

It is possible using the Brackets program (see later) to write and test JS games (largely) without an internet connection; you can also do this without Brackets but that requires some set-up work described [here](http://phaser.io/getting-started-js2.php): and we will not take the time to do that right now; you can try it later at home if you like.

JavaScript games ultimately require several components, including i) a web page (e.g. somethimg.html), that refers to a JavaScript file (e.g. something.js) and with some styling from a css file (e.g. styles.css).

# Connect to the internet

This may require persistence in the Dojo and should be easy at home.

Take a look at some (Phaser) examples: <http://examples.phaser.io/sideview.html> to see what is possible with JavaScript and Phaser.

Click any link that you like on this page. The game will show at the top of the page and the JS code underneath.

# Work with JSbin to begin with

## Open examples in JSbin

JSbin is a website that makes it relatively easy to make a game and then share it with everyone, without having to install anything or set anything up. However the JS code for JSbin will require a few edits to make it work.

The simple examples have JSbin links: click these to see what JSbin looks like and also how the code looks a little different.

## [Optional] Log in to JSbin

To work well in JSbin, it is better to log in (then it is easier to save and share your work).

The login details are:

Username: [bray@coderdojo.com](mailto:bray@coderdojo.com) Password: brayhead

## Your first edits

Some lines of code need to be edited to work in JSbin; a few lines need to be added and the first line needs moving to the end (and editing).

|  |  |
| --- | --- |
| Code shown in Phaser website | Code needed in JSbin |
| var game = new Phaser.Game(800, 600, Phaser.CANVAS, 'phaser-example', { preload: preload, create: create });  function preload() {  // You can fill the preloader with as many assets as your game requires  // Here we are loading an image. The first parameter is the unique  // string by which we'll identify the image later in our code.  // The second parameter is the URL of the image (relative)  game.load.image('einstein', 'assets/pics/ra\_einstein.png');  }  function create() {  // This creates a simple sprite that is using our loaded image and  // displays it on-screen  game.add.sprite(0, 0, 'einstein');  } | var MainState={  preload:function() {    // You can fill the preloader with as many assets as your game requires  // Here we are loading an image. The first parameter is the unique  // string by which we'll identify the image later in our code.  // The second parameter is the URL of the image (relative)  this.load.image('einstein', 'http://examples.phaser.io/assets/pics/ra\_einstein.png');  },  create:function() {  // This creates a simple sprite that is using our loaded image and  // displays it on-screen  this.add.sprite(0, 0, 'einstein');    }  };  var game = new Phaser.Game(800, 600, Phaser.CANVAS, 'phaser-example', MainState); |

Open the JSbin version for this first example in a Chrome browser tab. Then open another tab and Click File>New in JSbin to start your own ‘bin’.

Copy and paste the JSbin code from their tab to yours, starting with the HTML section, then JS.

You should soon have the Einstein picture in your Output tab in JSbin.

Now experiment with your bin; e.g. find another image on the web that you’d like to use instead of Einstein; change the code so that your image shows up.

Use File>Save snapshot any time to save the current version of your work.

## Repeat the above for the other examples with JSbin links

Open example 2 using the Phaser JSbin link and then copy that into your own code project (tab) in Chrome. Overwrite what you had before.

Repeat this for each of the Phaser examples up to Example 6.

# Get ready to create your own games outside JSbin

If the above was interesting for you, you may be ready to build your own games for the web, outside JSbin.

To get started on this, download Brackets, a free, powerful and easy to use code editor: <http://brackets.io/>. Brackets has a nice Live Preview feature that lets you see how your game will look before you deploy it on the web. You can use this feature with files stored locally on your computer and without connecting to the web.

## Build your first game outside JSbin

Now build your first game using the instructions here:

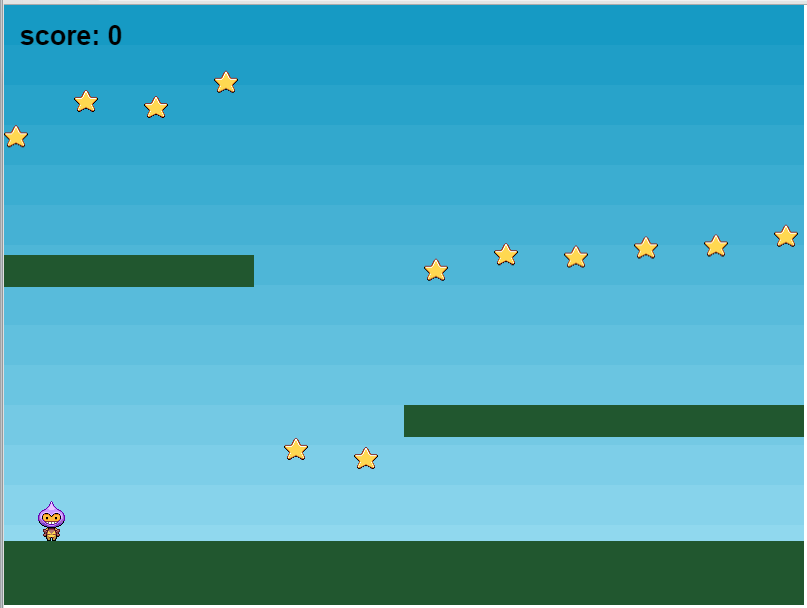
<http://www.photonstorm.com/phaser/tutorial-making-your-first-phaser-game>

Where the Phaser instructions say “Download the resources for this tutorial from here and unzip it into your web root.”, that means for you: Download the resources for this tutorial from here and unzip into the folder where you are building the game”.

(If you’re doing this in the Dojo and have trouble getting internet access, ask a mentor – they may have the files you need on a memory stick.

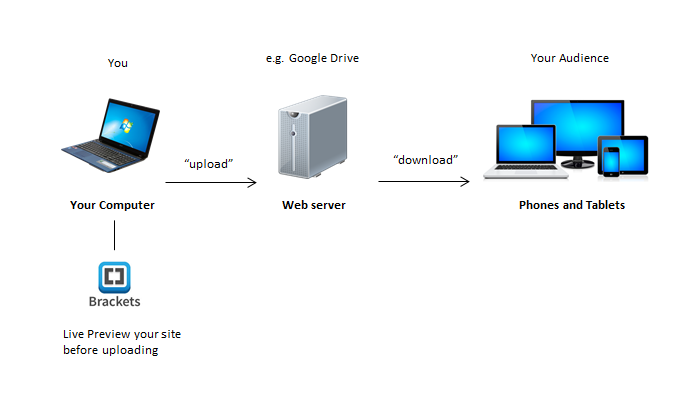
Copies of many of the files are also in the Dropbox folder you can reach from <http://scratch.coderdojobray.com/>)

You will know this is all working well when you open page9.html in Brackets and use Live Preview to see:



# Deploying games on the web free, using Google Drive

When building a JS game you will work with files stored locally on your own PC (in a folder) and you will later copy those files to a ‘web server’ so that you and others can then visit them in a web browser.



For this second part, you need access to a web server, that is, a place on the Internet where you can save copies of the same files you have in a folder on your PC. Once they are on a web server, connected to the Internet, the whole world can visit your pages using web browsers.

You can use Google’s webservers for this purpose, quite easily and free of charge by following the notes below. (There are other ways to get a web server but they will either cost some money or be more complex to set up).

Sign up for a Google account if you don’t already have one

You will (see below) upload your game(s) to folders in your Google Drive (you get 15Gb free of charge). To access your Drive, click the button indicated below – this usually appears near the top of the page when you visit Google:

First click: then click

Once you have a folder that you want to share (see below), follow Google’s instructions (taken from this [link](https://support.google.com/drive/answer/2881970?hl=en)), with a couple of changes we have found to work better: To host a webpage with Google Drive:

1. Open Drive at [drive.google.com](https://drive.google.com/) and select a **folder**.
2. Click the **Share** button near the top of the page.
3. Click **Advanced** in the bottom right corner of the sharing box.
4. Click **Change...**.
5. Choose **On - Public on the web** and click **Save**.
6. Before closing the sharing box, copy the folder’s **document ID** from the URL in the field below "Link to share". The document ID is a string of *uppercase and lowercase letters and numbers* between slashes in the URL. E.g.

Link looks like: https://drive.google.com/folderview?id=0B-Q4NqewvNasZTFyajh4NzU2Nk0&usp=sharing

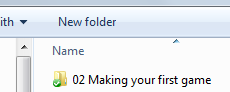
Document id is 0B-Q4NqewvNasZTFyajh4NzU2Nk0

1. Share the URL that looks like "www.googledrive.com/host/[doc id] where [doc id] is replaced by the folder ID you copied in step 6. E.g. link to share is:

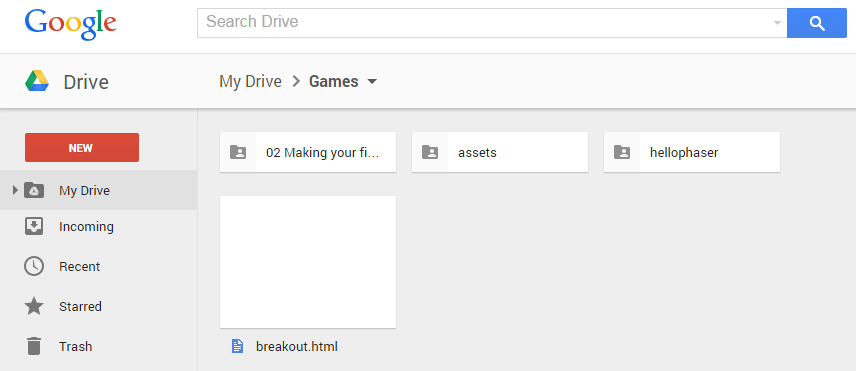
www.googledrive.com/host/0B-Q4NqewvNasZTFyajh4NzU2Nk0

## Deploy your first game on your own website

Now deploy your first game on your own website (this section assumes you will use Google Drive for this).

Try copying all the files and folders you have been using locally on your PC for your first game to a shared folder in your Google Drive.

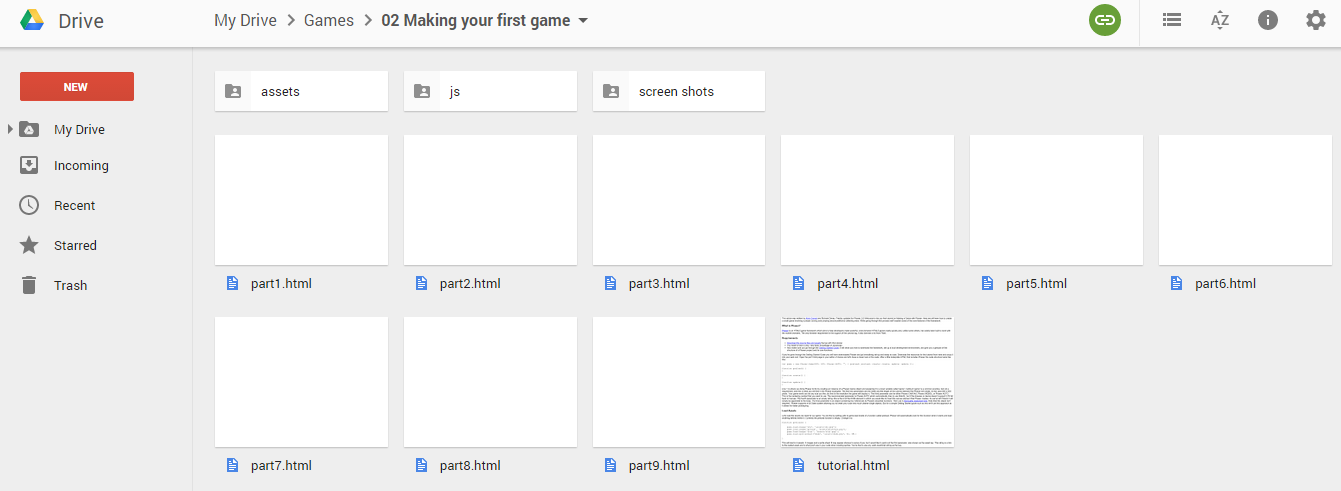
For example, my local folder on my PC looks like:



And the games area of my Google Drive looks like:

The folder on the top left above is where I put the unzipped files downloaded from Phaser for ‘making your first game’.

That folder contains:



To view the contents of your folder as a website, make a link to the whole folder as described in steps 6 and 7 above and open the address in your web browser.

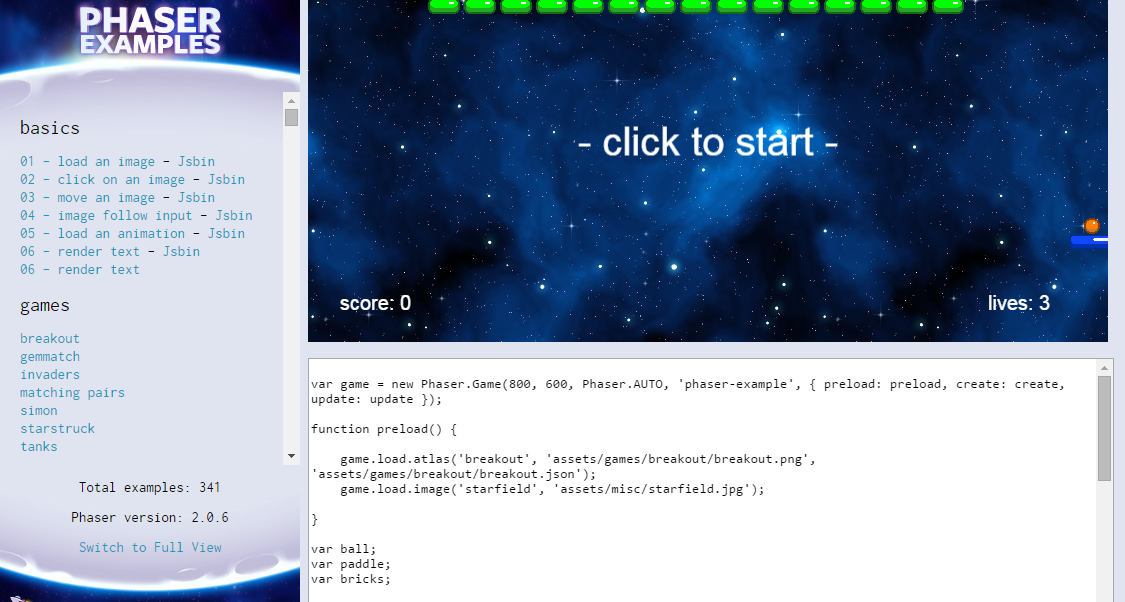
The long links generated by Google to display your game can be shortened using URL shortening websites like goo.gl or bit.ly, to something more like this: <http://bit.ly/dojogame>. (This is a link to a ‘breakout’ game, that you can add to your own website below).

**Tip**: You will find it easier to get Phaser examples working on your own website if you download their [Examples](https://github.com/photonstorm/phaser-examples/archive/master.zip) folder, unzip it and upload the **phaser-examples-master\examples\assets** folder to your own site. It is about 70 Mb and will take a few minutes to upload. Later you can add your own assets (mostly images you want to use for sprites) in this folder.

You can store the Phaser JS in a JS folder for each game or you can reference JS from the web using a source like <https://cdnjs.cloudflare.com/ajax/libs/phaser/2.2.1/phaser.min.js>.

## Copy a game from the Phaser Examples and deploy it to your site

To take your learning a bit further, copy a game (e.g. breakout) from the Phaser examples site to your own website. To do this, use the part 9.html page from the above example as a starting point. Then replace the JS code with code copied from here.



Having your own assets folder on your site will really help with this. And don’t forget to link to phaser.min.js.

Use Brackets Live Preview with your local files to get everything working. Then upload to your website and make sure it all works online.

## Fancy copying another game? Try Flappy Bird

If you want to keep practising by copying other games, you could try Flappy Bird here: <http://blog.lessmilk.com/how-to-make-flappy-bird-in-html5-1/>.

Test your work using Brackets and then deploy to your own website. Make it work on your phone too.

# Take your JavaScript and Phaser skills to game authoring level

You can learn lots more about what you can do with JavaScript and Phaser here: <http://phaser.io/examples/sideview.html>. We recommend that you spend a few hours during the week (between Dojo sessions) debugging and exploring how to build these games.

When you run into trouble, use Google to search for solutions to the specific problem that you have. Use the Phaser documentation too: <http://docs.phaser.io/>

To be good at this you will need to know JavaScript fairly well. There are good learning materials here: <http://www.w3schools.com/js/>.

There’s a pretty good set of videos teaching Phaser, here, on a fairly advanced project: <https://www.youtube.com/playlist?list=PLuPKgtwSH_obQV2rJF6TU8SR8KI9wINoi>

Soon, when you have working games on your site, send the links to us at [bray@coderdojo.com](mailto:bray@coderdojo.com).