

Introduction to
Babylon JS



Babylon JS Sandbox

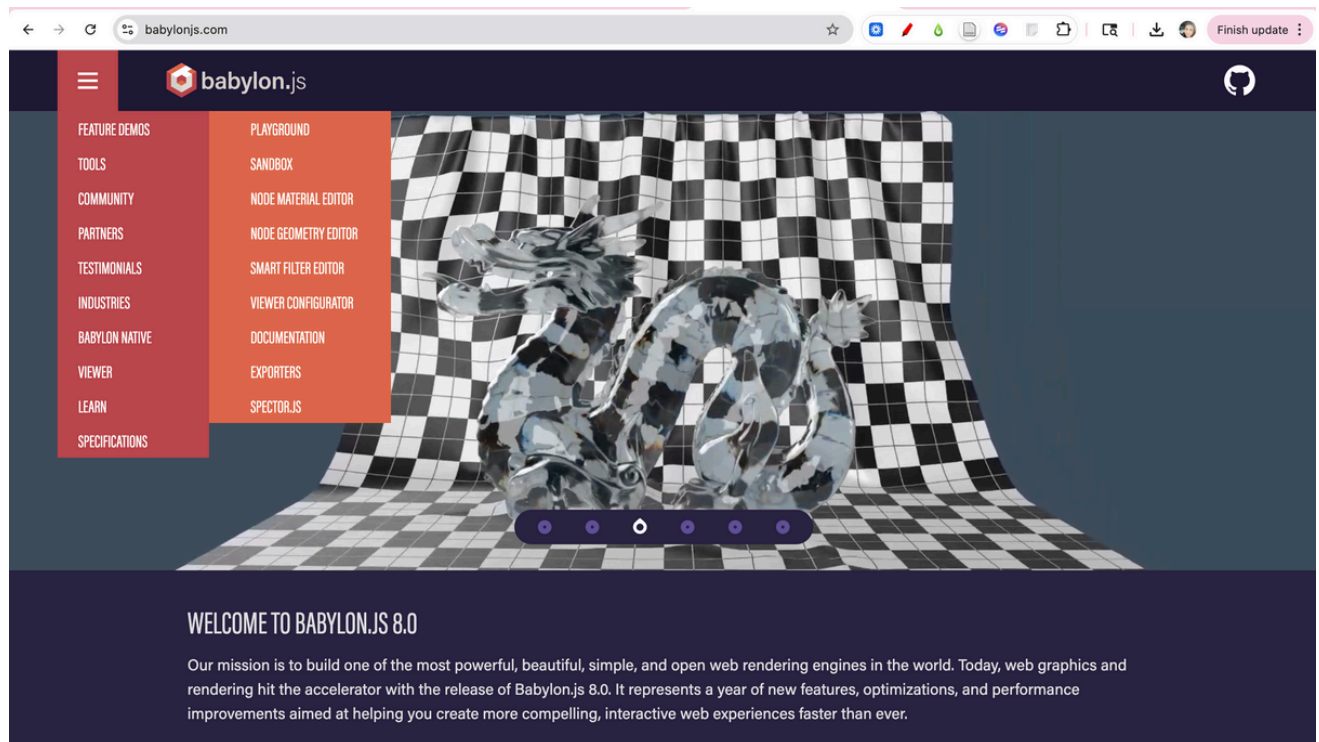
The Bridge: From Snapchat to the Open Web

Congratulations! You have mastered the "Walled Garden" of Snapchat. You've built interactive AR that millions can see inside an app. But what if you want your AR to live on a website, an email, or a digital portfolio that you control completely?

It's time to meet Babylon.js.

Same Logic, Different Playground

The "If/Then" logic you used for your "Mouth Open" trigger in Lens Studio is exactly how professional web developers build 3D games for the browser. The only difference is the "Home" where your code lives.



Introducing the Babylon.js Sandbox

Before we write a single line of web code, we need to see how our 3D models (like your "POW" stickers) look in a browser. For this, we use the Babylon.js Sandbox.

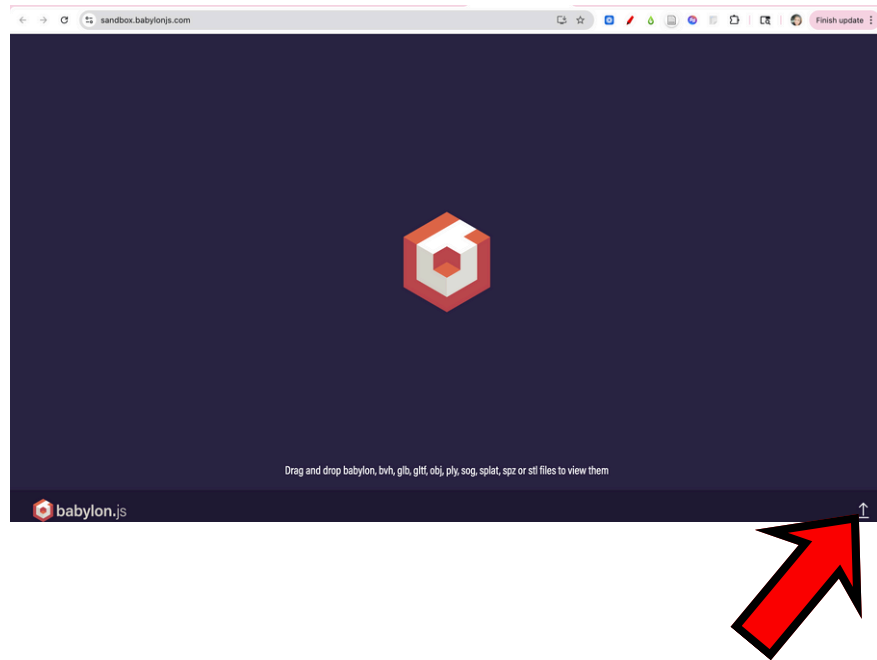
Think of the Sandbox as a "Universal Viewer."

To access Babylon.js Sandbox go to babylonjs.com. In the menu search Tools > Sandbox.

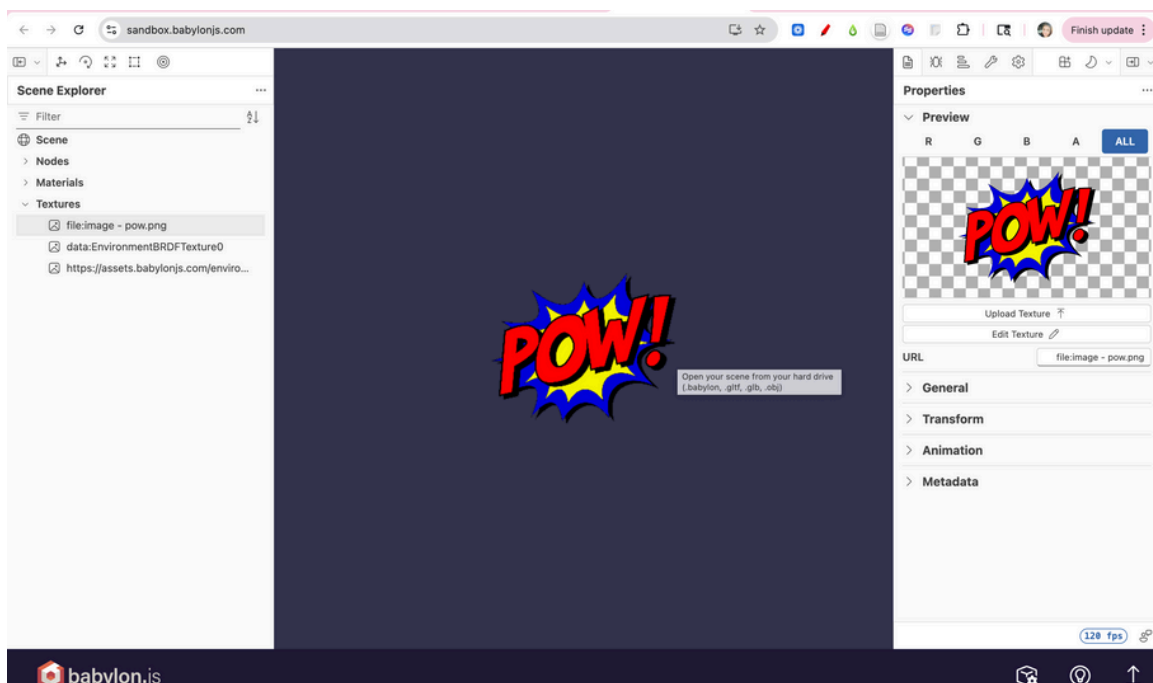
Babylon JS Sandbox

When we go to the sandbox, the page prompts us with different types of files to load. Even though not listed, Babylon can also use the same .png files we used for Lens Studio.

Select the Upload button in the bottom right corner and upload one of the image files.



And we are now in the Sandbox!



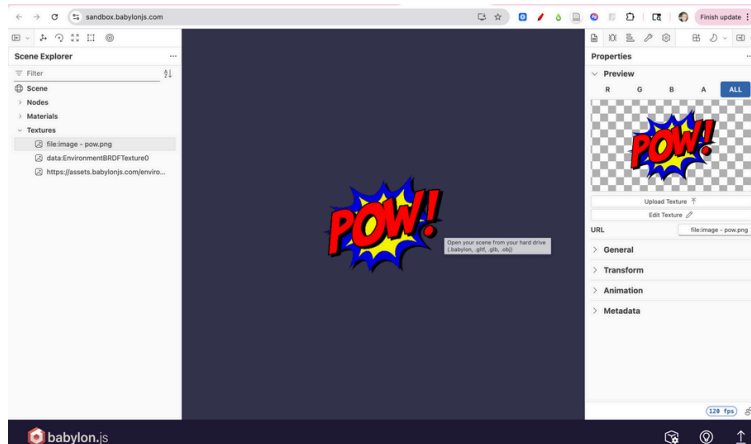
Babylon JS Sandbox

The Sandbox: Bringing 2D Art to the 3D Web

When we move your 2D stickers from Lens Studio to the Babylon.js Sandbox, we are essentially placing your drawing on a digital piece of glass that can float in a web browser.

How it Works with 2D Images

1. *The Transparent Sticker:* You take your .png file (the one with the transparent background).
2. *The Plane:* In the Sandbox, we place that image onto a flat surface called a **Plane**.
3. *The View:* Just like in Lens Studio, we can tell the web browser to make that plane always face the user.



Think of the Sandbox like a Digital Mirror. When you look in a mirror, you can see yourself and even adjust your hair or clothes (the Inspector), but the mirror doesn't "save" a photo of you. Once you walk away, the mirror is blank for the next person.

The Sandbox "Save" Rules

Because the Sandbox is a Web-Based Viewer, it doesn't have a "Hard Drive" to store your work. Here is what happens when that tab closes:

- *Temporary Workspace:* Any changes you made to the colors, the lighting, or the position of your stickers in the Inspector will vanish.
- *Safety of your Files:* The good news is that your original image asset files on your computer are safe.
- *No URL Sharing:* Unlike a Snapchat Lens which has a permanent link, you cannot send someone a link to your Sandbox session. If they want to see your art, they have to drop the files into the Sandbox on their computer.

Babylon JS

Babylon.js is a professional-grade 3D engine that allows your interactive art to live anywhere—on your own website, inside a mobile browser, or even in a VR headset. If Lens Studio is a "walled garden," Babylon.js is the "open frontier."

What is Babylon.js?

Babylon.js is a powerful JavaScript library used to display 3D graphics in a web browser using HTML5. In the context of STEAM, it represents the "Universal Language" of 3D on the web.

- **No App Required:** Your users don't need to download Snapchat or Instagram. They just click a URL link, and your experience starts.
- **WebXR Support:** This is the industry standard for "Web-Based Augmented Reality." It allows your phone's browser to use its camera to "track" the floor just like Lens Studio does.
- **Total Ownership:** Because you host the code yourself, you aren't subject to another company's review process or design rules.

The Technical Bridge: "Same Logic, New Syntax"

The most important thing to remember is that we are not starting over. The Computational Thinking you practiced in Lens Studio (If/Then, Events, Scales) translates directly into Babylon.js. We are just using slightly different words to say the same thing.

The "Big Three" of a Babylon Scene

To build a world in Babylon.js, you always need three core components. Think of this like setting up a physical Film Set:

1. *The Engine & Scene:* This is the "Stage." It's the digital space where everything exists.
2. *The Camera:* This is the "Audience's Eyes." In AR, your phone's camera becomes the Babylon camera.
3. *The Light:* Just like in real life, if there's no light, your 3D art will be pitch black!

Unplugged Activity: The 2D Sticker in a 3D Room

The Setup:

- A 2D drawing of a "POW" sticker on a piece of paper.
- A pair of scissors and a small piece of tape.
- A pencil.

The Exercise:

1. *The 2D Asset:* Cut out your "POW" sticker and tape it to a pencil so it can stand upright.
2. *The "Fixed" Orientation:* Place the pencil on your desk so the "POW" is facing the front of the classroom. Do not rotate the pencil.
3. *The 3D Walk:* Move your head (the camera) in a circle around the desk.
 - a. What happens when you are in front? You see the full art.
 - b. What happens when you move to the side? The sticker looks thinner and thinner until it almost disappears!

The Lesson

In the Babylon.js Sandbox, your 2D stickers are flat planes. By default, they stay facing one way. If you want your user to see the "POW" from every angle, you have to tell the computer to turn that sticker into a Billboard that always turns to follow the camera.

Review of the Sandbox

Navigating the Sandbox Interface

The Sandbox is divided into three main zones that help you debug your 2D stickers in a 3D environment.

1. The Scene Explorer (Left Side)

Think of this as the List of Ingredients for your web scene.

- *Nodes & Scene*: Just like the Objects Panel in Lens Studio, this shows every object currently "living" in your browser.
- *Textures*: This is where your 2D .png files live. In your screenshot, you can see file:image - pow.png listed here. This confirms the web browser has successfully "unpacked" your sticker.

2. The Viewport (Center)

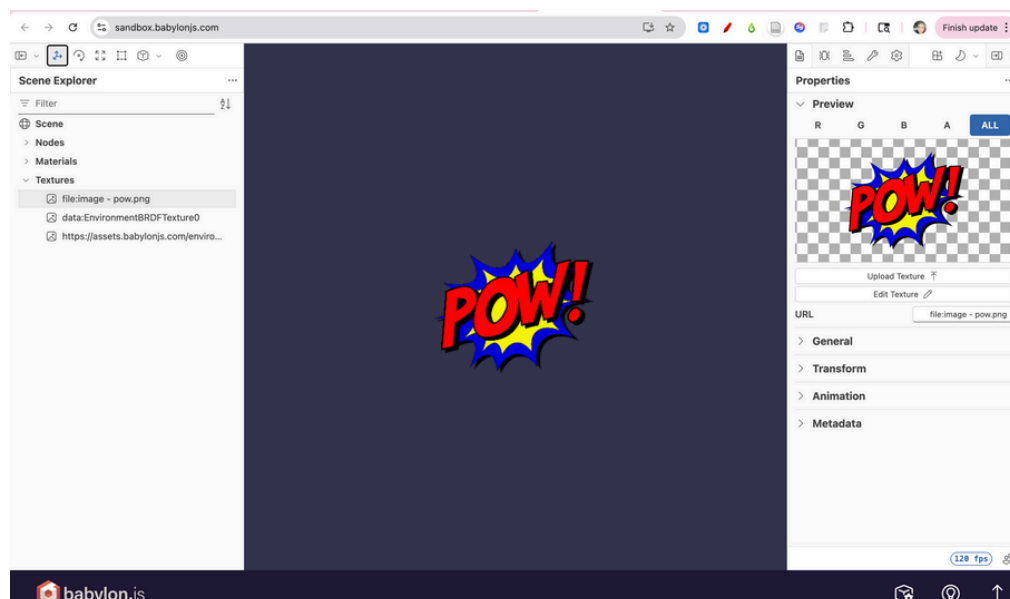
This is your "Digital Stage."

- *The Sticker*: Your "POW" graphic is shown here as it appears in 3D space.
- *Interaction*: You can use your mouse or touch screen to rotate the camera around the sticker to test the "Billboard" effect we practiced earlier.

3. The Properties Panel (Right Side)

This is similar to the Inspector in Lens Studio.

- *Preview*: Shows a thumbnail of your original sticker file.
- *Transform*: This is where you can manually change the Scale (size) or Rotation (angle) of your "POW" graphic.
- *Animations*: If your sticker has a "bounce" or "spin" effect, you can play and pause it here.



Coming Soon!

You've built the foundation. You are no longer just a "user" of technology; you are a creator.

What You've Achieved:

- *Spatial Engineering*: You mastered the 3D grid and learned how to place objects in a digital room.
- *Computational Logic*: You used "If/Then" triggers to make your art respond to the real world.
- *Web Readiness*: You transitioned your skills from a single app (Snapchat) to the open internet (Babylon.js).

Coming Soon

The Sandbox was our "Testing Lab," but in the next workbook, we are building a permanent home for your work with Babylon. We will move beyond just viewing art and start programming full interactive experiences.

In the next volume, get ready to:

- *Launch a Live Website*: Use GitHub to host your AR projects so anyone with a phone can see them.
- *Write Web Code*: Learn the "Magic Words" of JavaScript to build your own custom 3D buttons.
- *Master WebXR*: Turn any smartphone into an AR viewer without needing a single app download.

The "POW" on your screen is just a flat image, but the skills you used to put it there are three-dimensional. Keep your .png files safe—your journey into the world of Web3D is just beginning!