1. What We're Making
2. Why make a Snowglobe?
    1. A snowglobe is a nice, fun, easy, and customizable project that will help us master some basic concepts of HTML and CSS, along with exploring some more advanced features of JavaScript.
3. Quick Start
    1. To get started, visit hack.af/SGTemplate. You'll need an account on repl.it to get started. Hit the "Use Template" button at the top of the page. Fill out the options, and then create the new project!
    2. For the code snippets, visit hack.af/SGNotes. (Note: You can also print these out before you do the presentation and hand them out physically.)
4. Making it Stylish ("Now, let's dive right into it and get started with making it stylish!")
    1. Reset the Spacing
        1. First, we'll want to go into the `<style>` tag. If it's not already there, we'll reset the spacing. Look at the first code snippet on your page. To start, we'll select all elements, then after that, we'll set both the padding and margin to zero.
    2. Body Basics
        1. Next, we’ll set up the body structure using flexbox, center the content, and fill the page.
        2. Look at the code snippet. To start, we'll select the `body` element.
        3. Then, we'll quickly set the `display` mode to `flex`. Flexbox is a cool display mode that automatically and evenly spaces out content into boxes that can shrink, grow, and flow different ways. We're using it to easily center our content.
        4. Next, we'll do just that: center the content, using the `justify-content` and `align-items` properties... both will be set to `center` in order to center our content on both axis.
        5. Finally, we'll set the length and width to fill the whole screen using `100vh` and `100vw`. These will calculate 100% of the viewport for each respective measure.
    3. Stylish Snow
        1. Now, we’ll set the size of the snowglobe, its background, spacing, and add a faint glow.
        2. Like we did with the `body`, we'll use `v` units to set the size. We'll use a new unit called `vmin`, which uses the smaller of the height and width axis. Set both the height and the width to `85vmin`.
        3. Next, we'll set a nice, temporary background. Don't worry, though, you'll get the opportunity to change this later. We'll set `background` to a sky blue with the hex code `#45b3e0`.

        4. Finally, we'll add a cool glow using box-shadow. Go ahead and copy that code from your code snippets while I explain. The first two values specify the offset of the shadow, the second two specify the spread and blur radius, while the final value specifies the color of the shadow. All in all, this just makes a cool blur effect.
    4. Breaking In
        1. Finally, we’ll set the area of the particles and ground.
        2. First, we'll select both the `#particles` and `#ground` elements. These should both have a `width` of `100%`.
        3. For the particles, we'll want it to fill the majority of the snowglobe, taking up a `height` of `75%`
        4. For the ground, we'll want to take up the rest of the space available, therefore we'll use a `height` of `25%`.
        5. Now, we'll add a background to the `ground` in order to contrast it from the background. For a nice blackish-grey, you can use a `background` of `#bebebe`.
5. Script it Up
    1. Declarations
        1. To start, we’ll declare some simple variables and select some elements.
        2. We'll use `document.getElementById()` to select the `snowglobe` as the `globe` variable.
        3. We'll predeclare the `particles` variable.
    2. In Motion (Part 1)
        1. Go ahead and look at your code snippets again. We'll write a quick function called `calculateAcceleration()` which will help us out later. Go ahead and copy this function into your code while I explain what it does. This function will take the x, y, and z from a motion event and use a simple equation to calculate the magnitude of the event. (Pause for them to finish)
    3. Shake it Up (Part 1)
        1. Now, we’ll write the function that runs when our snowglobe is “shaken”
        2. First, we'll run the `particles.play()` function that tsParticles exposes to us. This will unpause the particles and allow them to freely float.
        3. Next, we'll add the animation class "shake" to the globe.
        4. Finally, we'll set this animation to be removed after about 1.1 seconds.
    4. Choices, Choices
        1. Now, we'll load the configuration options from tsParticles. we'll use tsParticles' loadJSON function to load the particles configuration file `particles.json`. Next, we'll start writing a callback function using `.then()`
    5. Shake it Up (Part 2)

        1. Next, we’ll add the event listener that triggers snow when the globe is clicked. We'll start inside that callback function we just made.
        2. First, we'll select the `particles` element using tsParticles' `domItem()` function.
        3. Then, we'll immediately pause these.
        4. Now, let's use `globe.addEventListener()` to add a `click` event listener. Inside this, we'll call the `shakeItUp()` function we wrote earlier. At this point, it should be working!

    6. In Motion (Part 2)
        1. Finally, we’ll add the event listener that triggers snow when the device is shaken.
        2. Inside the same callback function, we'll add a new eventListener to the built-in `window` element of the type `devicemotion.`
        3. We'll then calculate the acceleration using the `calculationAcceleration` function we made earlier.
        4. Finally, if this value is less than a predefined threshold value (I used 23), we'll call the function `shakeItUp()`. Otherwise, we won't do anything.

6. Customize it!
   Now, we'll customize our snowglobes!
    1. Set the Scene (Image)
        1. First, we’ll cover how to set an image background
        2. Find an image you have rights to (for example, from unsplash.com) and upload it to your project as background.jpg (or .png, .gif, etc.)
        3. Go back into the css for your snowglobe and change the background to, for example, `url('/background.jpg') center/cover;`
            1. The center/cover sets the background to fill the background and to crop at the center in order to do this.
        4. Tada, you have a background!
    2. Set the Scene (Gradient)
        1. Now, we’ll cover how to set a gradient background.
        2. We'll use the site cssgradient.io to set this up. First, select your colors on the Right side of the screen (Point to the right side of the screen.)
        3. Then, scroll down and copy the part of the gradient that starts with "linear-gradient."
        4. Then, inside of the CSS for the Snowglobe element, replace the background color with your custom gradient. In the end, your background should be filled with a beautiful color combination. Congrats!
    3. It's raining, It's Pouring
        1. Next, we’ll cover how to change the colour, shape, and density of what’s falling.

    2. To start, open the file `particles.json` and scroll down to the section that says "particles" near line 30.
    3. If you'd like to change the color, just change the "value" inside of the color object near line 30.
    4. If you'd like to change the shape, just change the value near line 129 to square, circle, or triangle. By default, it's circle.
    5. If you'd like more snow to fall, you can change the `value` inside of the `number` object near line 106. Increase it, or decrease it. It's your choice!
4. Twinkle, Twinkle
    1. Finally, Let’s spice it up and add some music!
    2. You can find some music on the internet archive for free if you'd like, but make sure you have the rights to use the music!
    3. Download that audio and upload it to your project as audio.mp3 (or ogg, m4a, etc.)
    4. Add a new `<audio>` element under the snowglobe div with the ID `music`. Then, add a new `<source>` element with the src of `audio.mp3` and a type of `audio/mp3` (or ogg, m4a, etc.)
    5. Then, next to the other element selections, select the music element with `getElementByID()`. Later, in the `shakeItUp()` function, add `music.play()`. This will cause the first source to start playing when the snowglobe is shaken.
    6. Congratulations!
5. It's an App!
    1. Finally, let’s take this site to the next level and turn it into an app!
    2. What is a PWA?
In order to do this, we'll use a web technology called a Progressive Web App (Or, PWA for short.) PWA uses a combination of smaller web technologies, like web manifests that declare your site and its content and service workers that allow your site to cache content and run in the background. All in all, it creates a neat little package for a web app.
    3. So, copy this code from the screen to turn this web into a PWA and turn it up to "10!"

> At the end of this time, go around for a show and tell, take questions, or help people who are stuck.