

making a custom digital
snowglobe

what we **are** making

A custom snowglobe that
you can take anywhere

A memory you can treasure forever <3

Unlimited hours of entertainment

what we're **not** making

A catapult that will tear apart
the space-time continuum

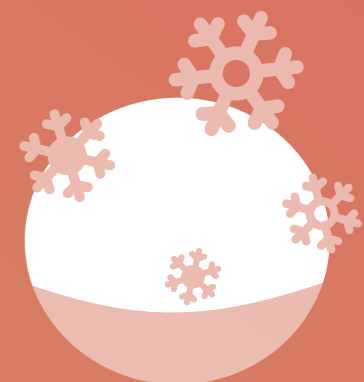
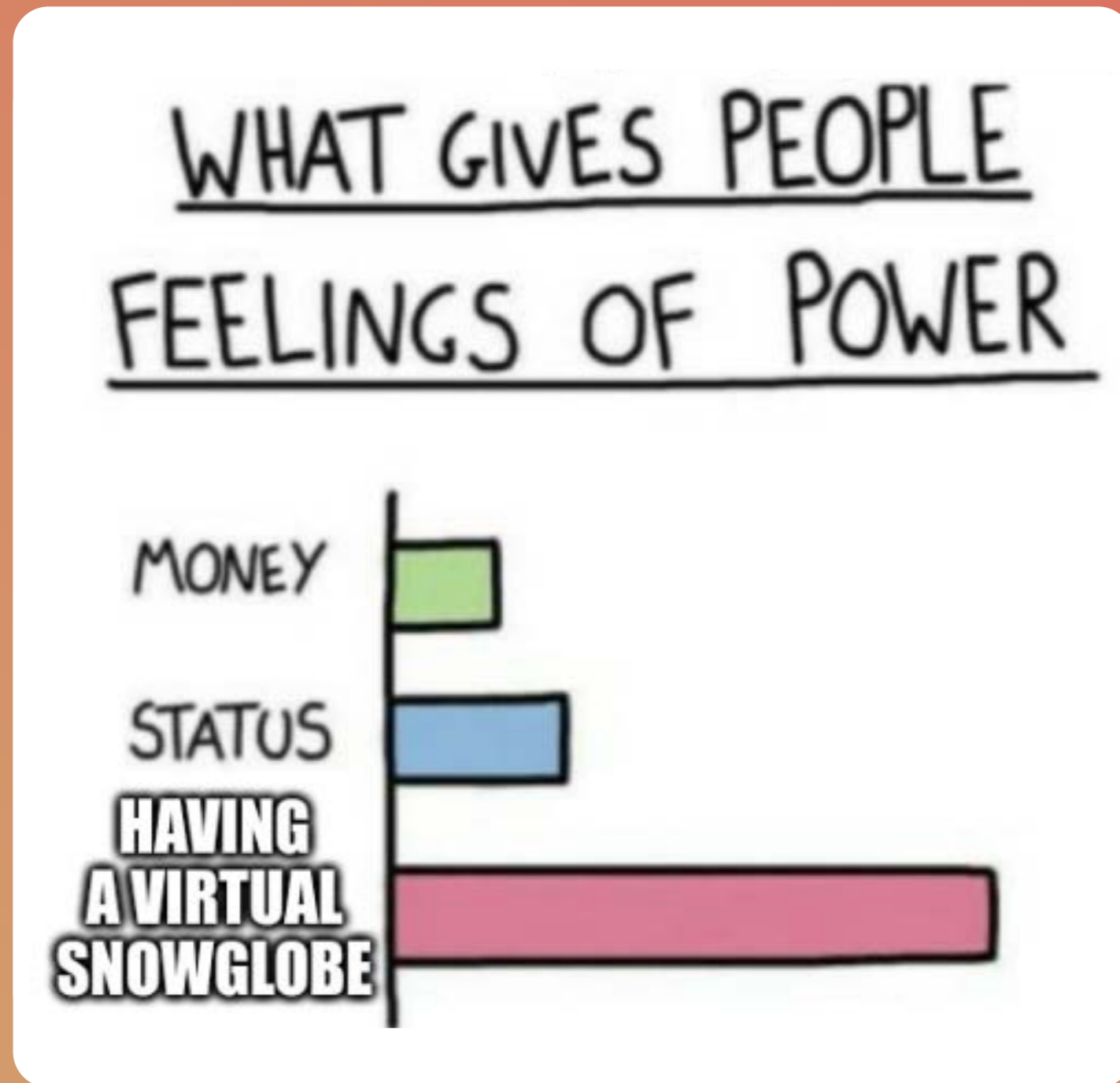
A high-yield hydroponic garden

A DIY Tiny House

it's not rocket science



why make a snowglobe ?



quickstart
quickstart

visit

hack.af/SGTemplate

to get started



quickstart
quickstart

visit

hack.af/SGNotes

to get the code snippets





making it stylish





1) reset the spacing

By default, web pages have preset margin and padding...
We'll fix that up.





2) body basics

Next, we'll set up the body structure using flexbox, center the content, and fill the page.





3) stylish snow

Now, we'll set the size of the snowglobe, its background, spacing, and add a faint glow.





3) breaking in
Finally, we'll set the
area of the particles
and ground.



{ } script it up





1. declarations

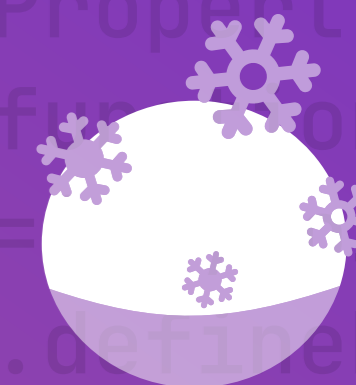
To start, we'll declare some simple variables and select some elements.

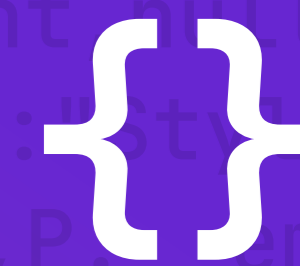




2. in motion (p1)

Next, we'll write a quick function to calculate magnitude... we'll use this later.





3. shake it up (p1)

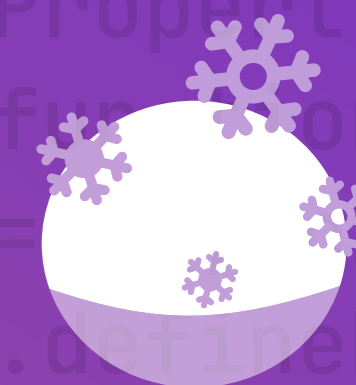
Now, we'll write the function that runs when our snowglobe is "shaken"





4. choices, choices

Next, we'll load the configuration options for tsParticles





5. shake it up (p2)

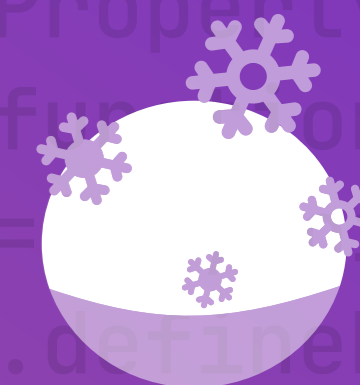
Next, we'll add the event listener that triggers snow when the globe is clicked.

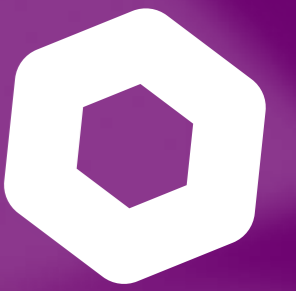




5. in motion (p2)

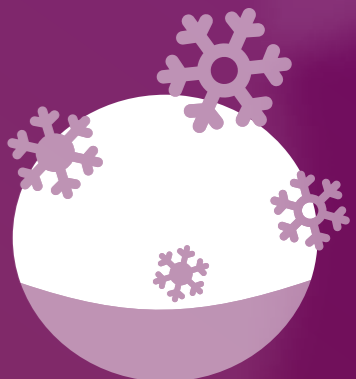
Finally, we'll add the event listener that triggers snow when the device is shaken.





1. set the scene

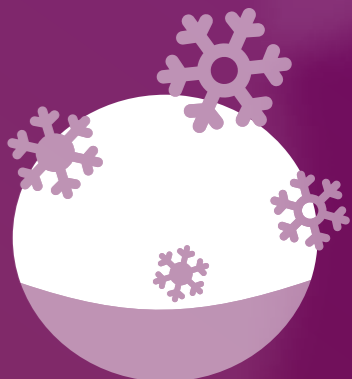
First, we'll cover how to set an image background





```
#snowglobe {  
  background: url('/background.jpg') center/cover;  
}
```

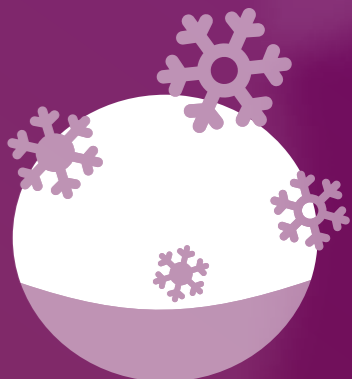
(center crops the center of the image,
cover fills the whole space with the image)





1. set the scene

Now, we'll cover how to set a gradient background

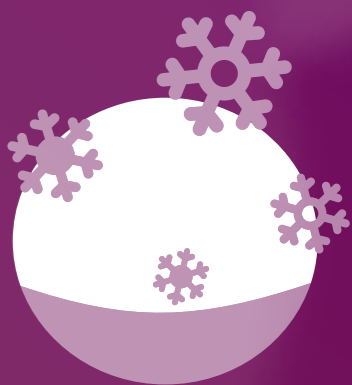




The image displays a web-based color gradient generator interface. At the top, a horizontal gradient bar transitions from dark blue on the left to light blue on the right. Below this bar, a control panel allows for precise adjustments. On the left, a vertical color picker shows a gradient from black to blue. The central 'COLOR CODE' section includes a hex code input field containing '#090979', and individual input fields for Red (9), Green (9), Blue (121), and Alpha (100). Below these is a rainbow color spectrum bar with a slider. On the right, a table lists the gradient stops:

Color	HEX	STOP	Remove
Dark Blue	#020024	0	X
Dark Blue	#090979	35	X
Light Blue	#00d4ff	100	X

At the bottom, the 'Linear' mode is selected, and the angle is set to 90°. An 'Upload Image' button and a row of five color swatches are also visible.





HEX R G B A

#790909

#040046

Linear Radial

4°

Upload Image

```
css
```

```
1 background: rgb(134,100,0);
```

```
2 background: linear-gradient(4deg, rgba(134,100,0,1) 0%, rgba(121,9,9,1) 37%, rgba(4,0,70,1) 100%);
```

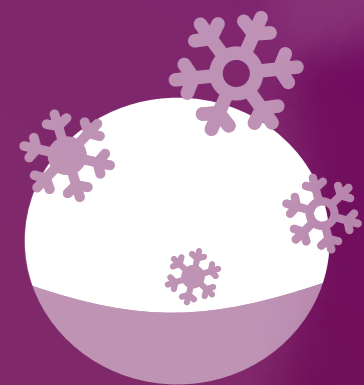
```
3
```

```
4
```

```
5
```

```
6
```

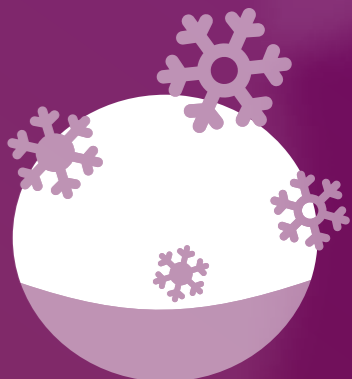
Copy to Clipboard





```
#snowglobe {  
  background: linear-gradient(4deg, #000, #111 100%);  
}
```

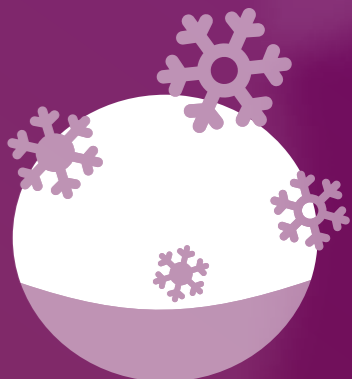
Replace this with your custom linear gradient code from cssgradient.io





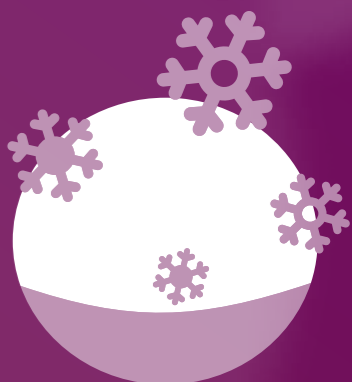
2. raining, pouring

Next, we'll cover how to change the colour, shape, and density of what's falling





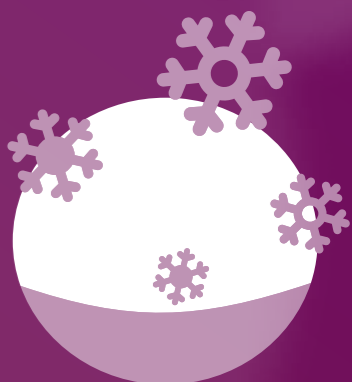
```
{
  particles: {
    colors: {
      value: '#000000' ← Change this to the
                       color you want
    }
  }
}
```





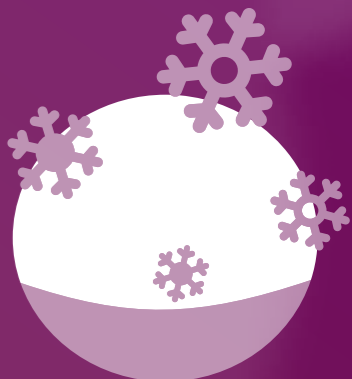
```
{  
  particles: {  
    shape: {  
      type: 'circle'  
    }  
  }  
}
```

← Change this to square,
circle, or triangle.





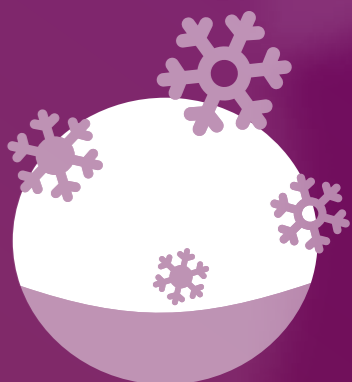
```
{  
  particles: {  
    number: {  
      value: '400' ← Change this number to  
                  the number of your choice.  
    }  
  }  
}
```

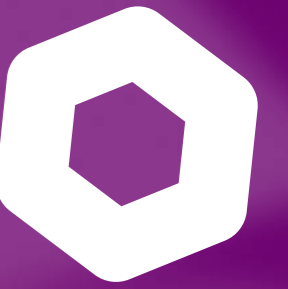




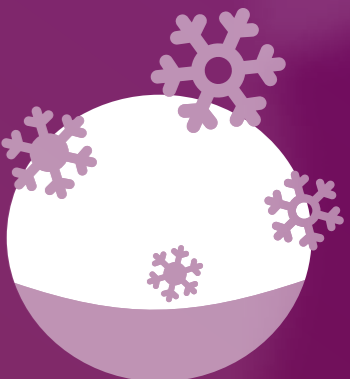
3. Twinkle, twinkle

Let's spice it up and add some music



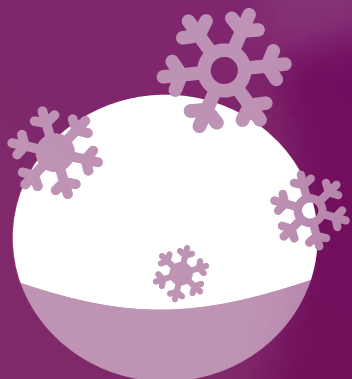


```
<audio id="music">  
  <source src="audio.ogg" type="audio/ogg"></source>  
</audio>
```





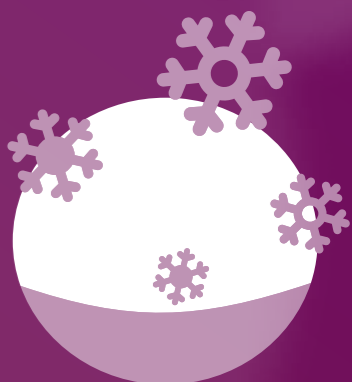
```
let music = document.getElementById('music')
...
function shakeItUp() {
  ...
  audio.play()
}
```





4. it's an app!

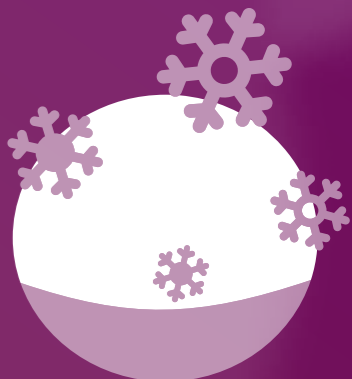
Finally, let's take this site to the next level and turn it into an app!





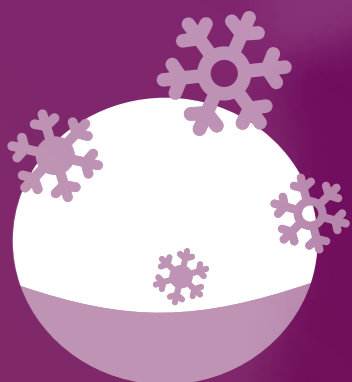
What is a PWA?

- Uses smaller technologies to make one big web app
- Allows your app to run in the background with service workers
- Ties your app into a neat little package that's installable





```
←!— PWA Stuff —→  
<link rel="apple-touch-icon" sizes="180x180" href="/apple-  
touch-icon.png">  
<link rel="icon" type="image/png" sizes="32x32" href="/  
favicon-32x32.png">  
<link rel="icon" type="image/png" sizes="16x16" href="/  
favicon-16x16.png">  
<link rel="manifest" href="/manifest.json">  
<link rel="mask-icon" href="/safari-pinned-tab.svg"  
color="#3063ac">  
<meta name="apple-mobile-web-app-title" content="Snowglobe">  
<meta name="application-name" content="Snowglobe">  
<meta name="msapplication-TileColor" content="#2b5797">  
<meta name="theme-color" content="#2e2e2e">
```



making a custom digital
snowglobe