



Mapbox Training Session #2


- Export style from Mapbox studio
- iframes & Github hosting
- Custom data

--

- Interactivity
- Functions Intro
- Events
 - a. Popup -> Custom display of info

Quick Review

Mapbox Studio custom style

- Publish!
-  Copy style URL and bring into the `map({style: xxx})` initialization in the JavaScript

Hosting on Github



Activity:

**Hosting Upload to
Github**



Another good option:

Netlify

iframes


```
<iframe width="300"height="200"src="http://google.com">  
</iframe>
```

```
<iframe width="300"height="200"src="http://google.com">  
</iframe>
```

```
<iframe width="300"height="200"src="http://google.com">  
</iframe>
```



Custom Data and GeoJson



Activity:

**Create Geojson at
geojson.io**



*[geojson.io and create data,
download, upload to map]*

map.addSource()

map.addLayer()

**Add to github + custom data
map?**



Intro to Interactivity

- **Functions + Events**
- **Mouseover/click events**
- **Popups**



Important Concepts



Functions + Events



Functions



Functions

A block of code that does something.

- Can be repeated many times.
- Return the same result or do the same thing.


```
function writeName (name) {  
  console.log(name)  
}
```

FUNCTION DECLARATION



```
function writeName (name) {  
  console.log(name)  
}
```


PARAMETER



```
function writeName (name) {  
  console.log(name)  
}
```



PARAMETER being used

```
function writeName (name) {  
  console.log(name)  
}
```



BODY

```
function writeName (name) {  
  console.log(name)  
}
```



Call the function

```
writeName( "Ben" )
```

Call the function

```
writeName( "Ben" )
```

```
writeName( "Ben Matheson" )
```



Events





Events

The user does something -
like click, scroll, hover, or
type. You code listens for
those. 👂 👂 👂 👂




Event Listener



Set up the map to  wait for
specific events and then 
do...something





The concept is that the
map  **LISTENS** for an



EVENT.

Then runs the  **FUNCTION**
every time that **EVENT**
happens.

#1 - set up event listener on  mouseenter for the volcanos layer.

#2 - create popup  (but don't do anything with it)

#3 position popup on map  in listener

#4 configure  data to show in popup

#5 mouseout set up  to get rid of popup



Activity:

Mouse Event



Simple Workflow

- upload data into studio
- configure a few key labels
- data driven styling
- Publish and use



volcanos.geojson