

Henning Muszynski

Session 6

JavaScript for Web



Feedback



What did we learn last time

Re-examination

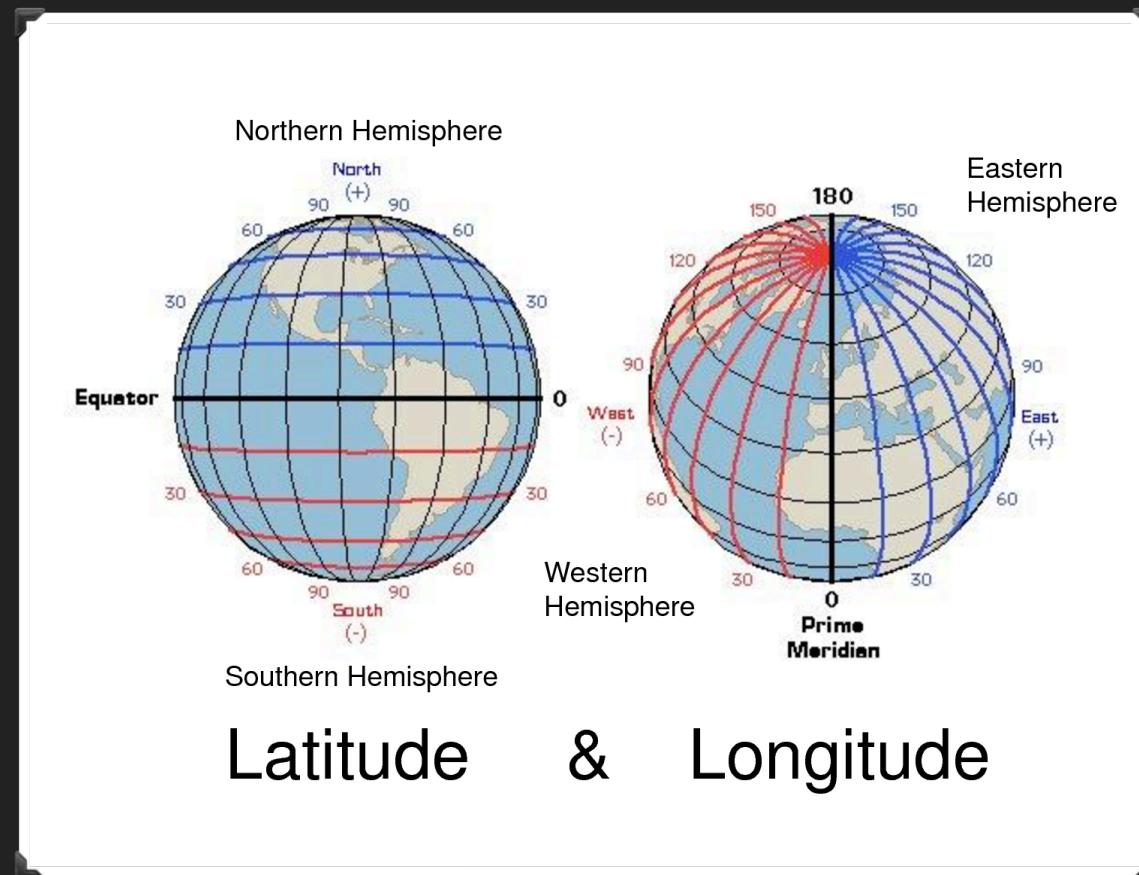
Web development in a nutshell

```
// create element
const linkElement = document.createElement('a')

// modify element
linkElement.href = 'https://google.com'
linkElement.innerText = 'Google'

// insert element
document.body.appendChild(linkElement)
```

Get the current location with the Geolocation API



Get the current location with the Geolocation API

Requires user permission - so we should always handle errors gracefully

```
if (navigator.geolocation) {  
    navigator.geolocation.getCurrentPosition(  
        (position) => {  
            const { latitude, longitude } = position.coords  
            console.log('Location:', latitude, longitude)  
        },  
        (error) => {  
            console.log('Oh no:', error)  
        }  
    )  
}
```

Display a map with Google Maps API

- ▶ Interactive map of any location
- ▶ Zoom levels (continents to houses)
- ▶ Markers and Lines



```
<html lang="en">
  <head>
    <script src="googleapis.com?callback=initMap" async defer></script>
    <script>
      let map

      function initMap() {
        const mapElement = document.querySelector('#map')
        map = new google.maps.Map(mapElement, {
          center: { lat: -33.9249, lng: 18.4241 },
          zoom: 12
        })
      }
    </script>
  </head>
  <body>
    <div id="map"></div>
  </body>
</html>
```

Showing a marker

```
const marker = new google.maps.Marker({  
  position: { lat: -33.0249, lng: 18.4241 },  
  map: map  
})  
  
// read position of marker  
const position = marker.getPosition()
```

Handling bounds of the map

```
const bounds = new google.maps.LatLngBounds()
cities.forEach(city => {
  const marker = new google.maps.Marker(/* ... */)
  bounds.extend(marker.getPosition())
})
map.fitBounds(bounds)
```

Handling events

```
map.addListener('click', event => {  
  /* ... */  
})
```

- ▶ Possible events include:
 - ▶ click, dblclick, rightclick, mouseout, dragend
 - ▶ center_changed, bounds_changed

Drawing a line

```
const line = new google.maps.Polyline({  
  map: map,  
  path: [  
    { lat: -33.0249, lng: 18.4241 },  
    { lat: -32.0249, lng: 19.4241 }  
  ]  
})
```



<https://github.com/henningmu/js-workshop>

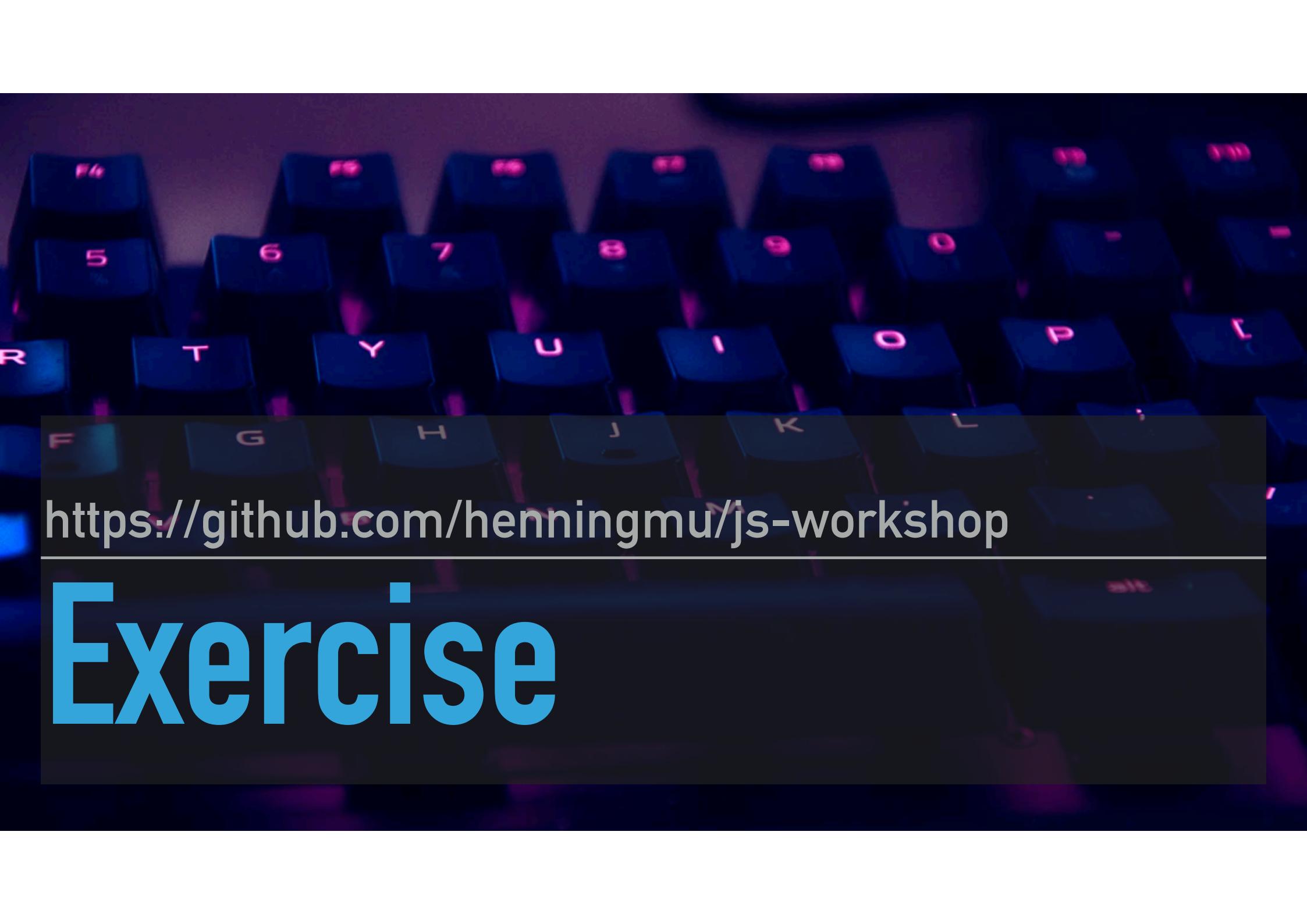
Exercise

Reverse Geocoding

- ▶ New API that needs to be enabled in the Google Cloud Console
- ▶ Allows to look up addresses based on coordinates

Reverse Geocoding

```
const geocoder = new google.maps.Geocoder()
geocoder.geocode(
  { location: { lat: -33, lng: 18 } },
  (results, status) => {
    if (status === 'OK') {
      if (results) {
        console.log(results)
      } else {
        console.error('No results found')
      }
    } else {
      console.error('Geocoder failed due to: ' + status)
    }
  }
)
```



<https://github.com/henningmu/js-workshop>

Exercise

Firebase: A comprehensive app development platform



Build better apps

- ▶ Databases
- ▶ Authentication
- ▶ Machine Learning
- ▶ Backends
- ▶ Hosting



Improve app quality

- ▶ Crash / Error reporting
- ▶ Performance monitoring
- ▶ App distribution



Grow your business

- ▶ Analytics
- ▶ A/B Testing
- ▶ Notifications

Cloud FireStore

- ▶ NoSQL Database
- ▶ Create, read, update & delete operations (CRUD)
- ▶ Automatic scaling, data syncing, user-based security

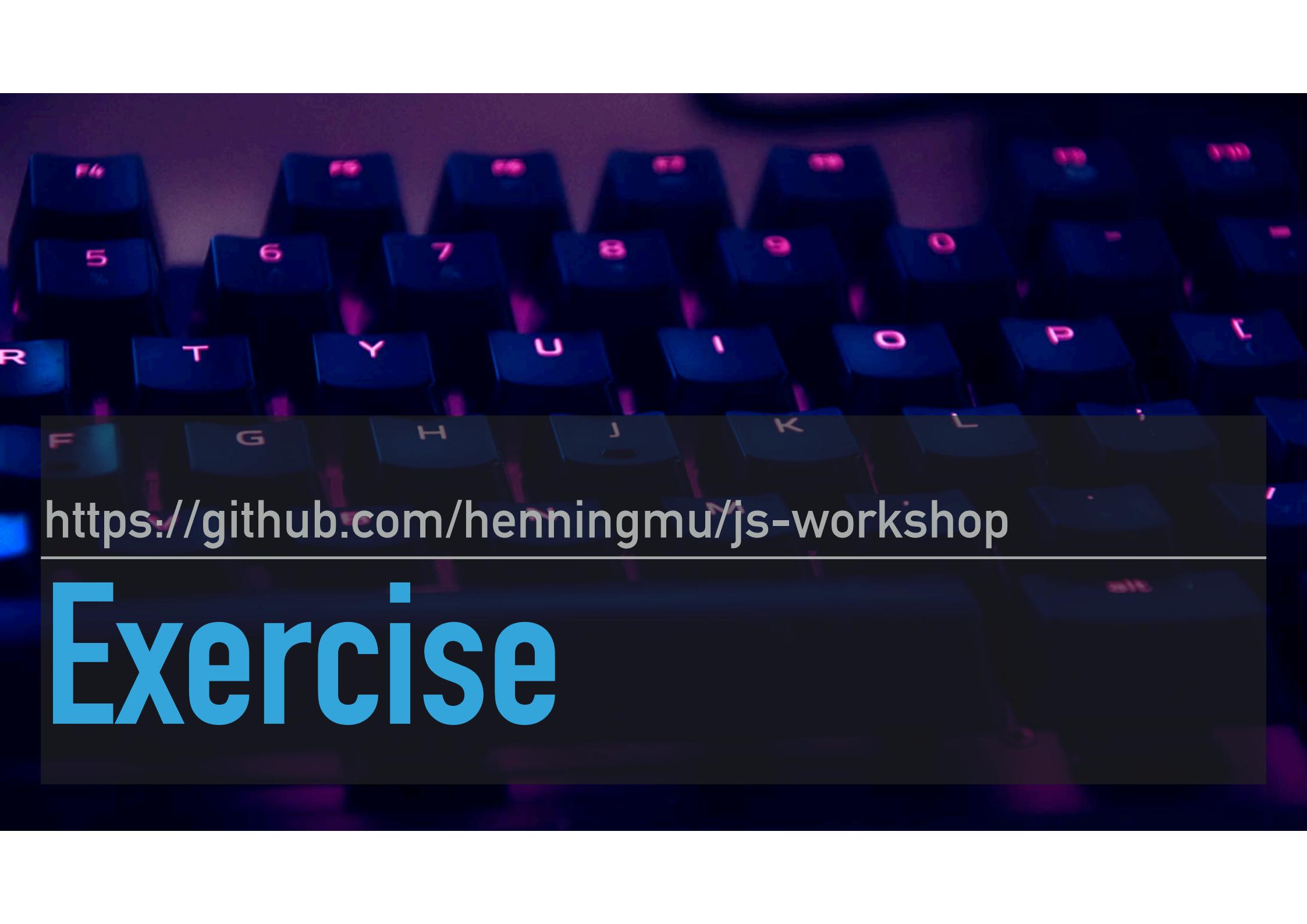


```
<script src="gstatic.com.firebaseiojs/7.firebaseio-app.js"></script>
<script src="gstatic.com.firebaseiojs/7.firebaseio-firebase.js"></script>
<script>
  const firebaseConfig = { /* ... */}
  // Initialize Firebase
  firebase.initializeApp(firebaseConfig)
  const database = firebase.firestore()

database
  .collection('posts')
  .add({
    title: 'Title of the post',
    description: 'Description of the post',
    postedAt: new Date()
  })
  .then(docRef => console.log('Created data:', docRef.id))
  .catch(error => console.error('Failed to create data', error))
</script>
```

Cloud FireStore: Reading data

```
database
  .collection('posts')
  .get()
  .then(snapshot => {
    snapshot.forEach(doc => {
      const data = doc.data()
      console.log("Data", data)
    })
  })
```



<https://github.com/henningmu/js-workshop>

Exercise



Quick Feedback