



Put yourself on the map

For this activity you require access to a PC/Mac/Raspberry Pi (plus peripherals) with internet connection.

A calamity of epic proportions is underway, and you and your team are the only ones who can save us! Can you create something that will protect the world during the apocalypse? What about something to avert the apocalypse altogether? Time to get your survival kits out and your thinking hats on, as you #MakeYourIdeas and save us all!

We've just received this message:



rpf.io/pioneers

We can't just give up on trying to reach people out there who might need help. So that they can find their way to the safety of our base, you need to make a simple map which survivors can access easily if they manage get any WiFi signal. The best way to do that is create a custom Google map and embed it in a webpage with HTML.



#MakeYourIdeas

For more inspiration and help, head to rpf.io/pioneers

For this task you're going to need the longitude and latitude coordinates of the bunker, and a computer with an internet connection. To access the Google Maps API from your base, you'll also need to get a secure key – let's do that first.

An API (or Application Programming Interface) is a set of functions you can use to interact with another computer – in this case the Google Map server. You need an **API key** to access the server, so that Google knows who is utilising its services.

You will need to create a Google account by visiting <http://accounts.google.com/SignUp>, or use an existing Google account to obtain an API key. (There is a minimum age requirement for creating an account, which varies depending on the country in which you live. Make sure you check first whether you are old enough to create an account.)

Visit [Google services](#), and under the heading “Step 3: Get an API key”, follow steps 1-4 to obtain a Google Maps JavaScript API key.

You can now use your private Key whenever you want to access the Google Maps API. Save it in a file and keep it somewhere safe. You can also login to Google again if you need a reminder of what it is.

Now that you have your API key, you can begin to make the page that will display your location to survivors. In your browser, navigate to Trinket.io, and log in to your account (or create a new one for free). Open up a blank HTML file in your Trinket window.

Begin by building the basic structure of your page. Enter the following code into the blank file you created, then save the file when you are done.

```
<html>
<head>
  <title>My map</title>
</head>
<body>
  My map will go here
</body>
</html>
```

Setting latitude and longitude

Now that you have the frame in which to place your map, you need to tell Google Maps which area you want to show.

Go back to your HTML file and find the `<head>` tag in your code. Position your cursor on the line below this tag and add the following code:

```
<style>
#map {
  width: 100%;
  height: 400px;
  background-color: grey;
}
</style>
```

This is some CSS code which will tell the map to take up the whole width of your screen, and to be 400px high. You can change these values to make the map larger or smaller if you like.

You need to access the Google Maps API using the key we generated earlier. **To use your new API key in a HTML page**, position your cursor just above the `</body>` tag. Add the following code:

```
<script async defer
src="https://maps.googleapis.com/maps/api/js?key=YOUR_API_KEY&callback=initMap">
</script>
```

Locate the part in the code you just typed which says `YOUR_API_KEY`, and replace it with your API key. **Note:** make sure there are no spaces between the key and the `=` and `&` characters on either side of it.

Now locate the sentence `My map will go here` in your code. Delete the sentence — you're going to add the map in its place!

Add the following code to create a <div> (an invisible box) where your map will eventually appear:

```
My Google map
<div id="map"></div>
```

Immediately below the <div> code you just added, add the following code to create the map:

```
<script>

    function initMap() {

        var Bunker = {lat: #, lng: #};

        var map = new google.maps.Map(document.getElementById(
            'map'), {
            zoom: 10,
            center: Bunker
        });
    }
</script>
```

Look at the line of code which begins with var Bunker. Replace the # symbols with the latitude and longitude values for the location of the base: [51.547190](#) and [-0.074143](#). The first value is the latitude or **lat**, and the second one is the longitude or **lng**.

For example:

```
var Bunker = {lat: 51.547190, lng: -0.074143};
```

Where is your base located? You'll need to add a marker to your Google map so people can get directions to it.

You can find your latitude and longitude by opening Google Maps and clicking once on the blue dot that indicates your current position. (You could also try the [GPS challenge found here](#), if you have a Raspberry Pi and a GPS dongle!) At the bottom of the screen, you'll see

your `lat` and `lng` listed. You can copy and paste these values into your code. If your base isn't where you are sitting right now, you can search for the address, or find it on the map, and click on its position to find the `lat` and `lng` coordinates.

Save your code. Now click **Run** in the Trinket window. You should see a Google map with your location of choice at the centre (try refreshing if it doesn't look right).

Adding markers

Now you can add markers to your map to designate important places, such as the base.

Below the map code you just entered, you need to add the script to import a marker. It's important to get the following code inside the `<script>` tags, otherwise it won't work.

Above the first `</script>` tag at the end of the `initMap` function, there is a single curly bracket. Add the following code inside that final bracket to make a marker called **Bunker** appear in the centre of your map:

```
var marker = new google.maps.Marker({
    position: Bunker,
    map: map,
    title: 'Bunker'
});
```

You can edit the position of your marker easily by entering different coordinates in the format `{lat: xx.xxx, lng: yy.yyy}`, like this:

```
var marker = new google.maps.Marker({
    position: {lat: 51.547190, lng: -0.074143},
    map: map,
    title: 'Bunker'
});
```

Another way to do this is to simply create new variables for other locations below the `var Bunker` line, like this (don't forget to change the variable each time!):

```
var Bunker = {lat: 51.547190, lng: -0.074143};
var Nottingham = {lat: 52.961034, lng: -1.158733};
```



#MakeYourIdeas

For more inspiration and help, head to rpf.io/pioneers

Then you can use your new variable instead of the one called **Bunker**.

You can also change the `title` to say anything you like!

Your final script should look something like this:

```
<html>
<style>
#map {
  width: 100%;
  height: 400px;
  background-color: grey;
}
</style>
<head>
  <title>My map</title>
</head>
<body>
  <div id="map">
<script>

  function initMap() {

    var Bunker= {lat: 51.547190, lng: -0.074143};

    var map = new google.maps.Map(document.
getElementById('map'), {
      zoom: 10,
      center: Bunker
    });

    var marker = new google.maps.Marker({
      position: {lat: 51.547190, lng: -0.074143},
      map: map,
      title: 'Bunker'
    });
  }
</script>
</div>
<script async defer
```



#MakeYourIdeas

For more inspiration and help, head to rpf.io/pioneers

```
src="https://maps.googleapis.com/maps/api/js?key=YOUR_API_KEY&callback=initMap">

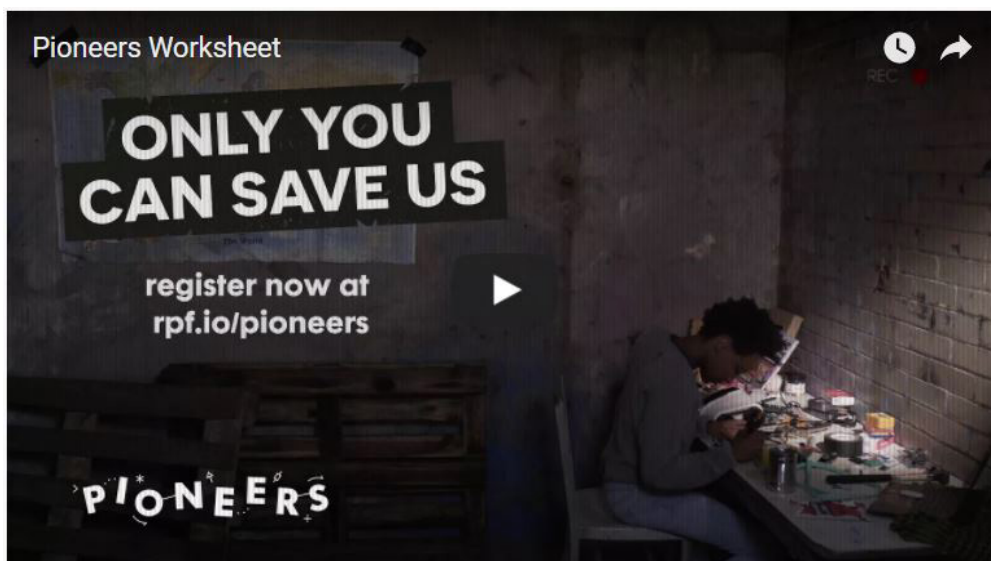
</script>
</body>
</html>
```

To add extra markers, simply repeat these steps, changing the values for `position` and `title` each time.

Debug and save your code. Click **Run** in the Trinket window, and you should see a Google map with your locations of choice displayed as markers.

You have just learnt to **use HTML to create an embedded Google map**.

You've done it! This message has just come through:



Visit rpf.io/pioneers to find out more about how only **you** can save us!

Take it further

For some more advanced ideas about how to make things with the Google Maps API, try these projects on the Raspberry Pi website:

[Poo Near you](#)

[Zombie Apocalypse map](#)

[Where's Zombie?](#)



#MakeYourIdeas

For more inspiration and help, head to rpf.io/pioneers