Emulate Mobile-Web using Chrome

In this lesson, we will learn how to emulate the mobile-web using chrome.

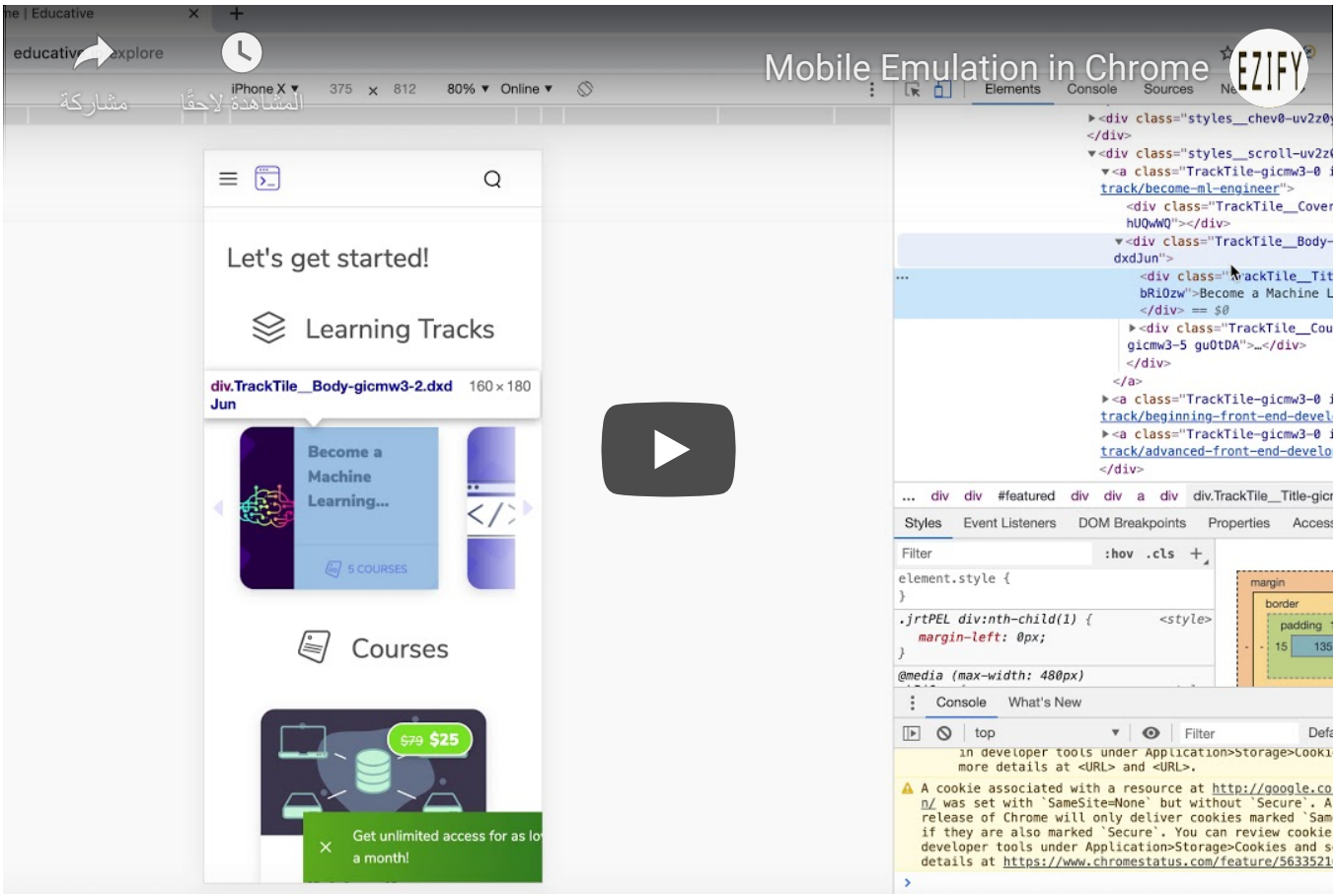
WE'LL COVER THE FOLLOWING

What is mobile emulation in Chrome?

Chrome allows users to emulate Chrome on a mobile device (e.g. an **iPad**, or an **iPhone X** or a **Pixel 2**, etc.) from the desktop version of Chrome; enable this using Chrome DevTools.

This feature allows a test engineer to quickly test how a website will render in a mobile device without requiring a real device.

How to emulate mobile in Chrome using Chrome DevTools?



Setting desired capabilities for mobile emulation in WebDriver

```
Map<String, String> mobileEmulation = new HashMap<>();

// adding name of the device to emulate
mobileEmulation.put("deviceName", "iPhone X");

ChromeOptions chromeOptions = new ChromeOptions();

// Experimental Option needs to be set for emulation
chromeOptions.setExperimentalOption("mobileEmulation", mobileEmulation);

WebDriver driver = new ChromeDriver(chromeOptions);
```

List of supported devices for mobile emulation

Any of the below devices can be emulated in Chrome by setting the value for the key `deviceName` in the above code snippet.

```
  "BlackBerry Z30",
  "Google Nexus 6",
  "Google Nexus 7",
```

```
  "Google Nexus 4",
  "Google Nexus 5",
  "Apple iPad",
  "Apple iPad Mini",
  "Samsung Galaxy Note II",
  "Nokia N9",
  "Samsung Galaxy S4",
  "Nokia Lumia 520",
  "BlackBerry PlayBook",
  "Apple iPhone 5",
  "Apple iPhone 4",
  "Apple iPhone 6",
  "Apple iPhone 6 Plus",
  "Amazon Kindle Fire HDX",
  "Samsung Galaxy S III",
  "Samsung Galaxy Note 3",
  "Google Nexus 10",
  "Pixel 2",
  "Nexus 6 P",
  "iPhone 8 Plus",
  "iPhone 7 Plus",
  "iPhone 7",
  "iPhone 8",
  "iPhone SE",
  "iPhone X",
  "Pixel 2 XL"
```

Adding custom devices for mobile emulation

We can emulate devices that are not available as a part of Chrome emulation by setting the device metrics.

```
Map<String, Object> deviceMetrics = new HashMap<>();

// device with custom width, height and pixel ratio
deviceMetrics.put("width", 380);
deviceMetrics.put("height", 670);
deviceMetrics.put("pixelRatio", 3.0);

Map<String, Object> mobileEmulation = new HashMap<>();
mobileEmulation.put("deviceMetrics", deviceMetrics);

// device having custom user agent
mobileEmulation.put("userAgent": "Mozilla/5.0 (Linux; Android 8.0.0; Pixe
l 2 XL Build/OPD1.170816.004) AppleWebKit/537.36 (KHTML, like Gecko) Chrom
```

```
e / 66.0.3329.0 Mobile Safari / 537.36");


ChromeOptions chromeOptions = new ChromeOptions();

// ExperimentalOption needs to be set for emulation
chromeOptions.setExperimentalOption("mobileEmulation", mobileEmulation);
WebDriver driver = new ChromeDriver(chromeOptions);
```

That's all about mobile emulation. In the next lesson, you'll learn a new concept, how to find elements in a web page using Selenium.