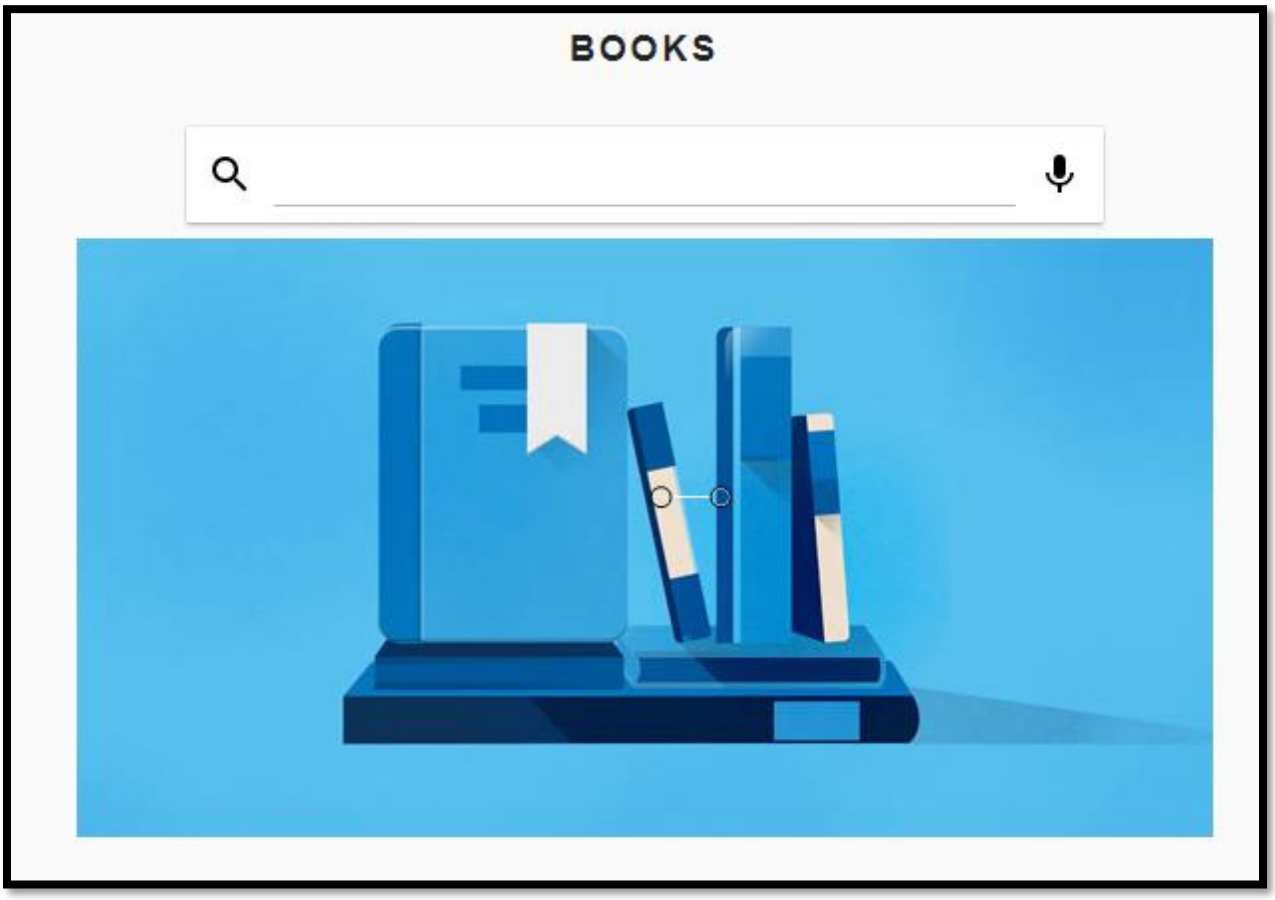
Rex Jones II

Intro To Shadow DOM For Selenium

Introduction

In this session, I am going to introduce Shadow DOM. We can think of a Shadow DOM like a DOM within another DOM. That's why it challenging to locate an element inside of a Shadow DOM. DOM stands for Document Object Model which represents the HTML Nodes in a tree format. The purpose is to define properties, define a structure, and define contents of a web page. When it comes to a Shadow DOM, it allows a developer to hide the DOM by attaching it to a tree within the original DOM.

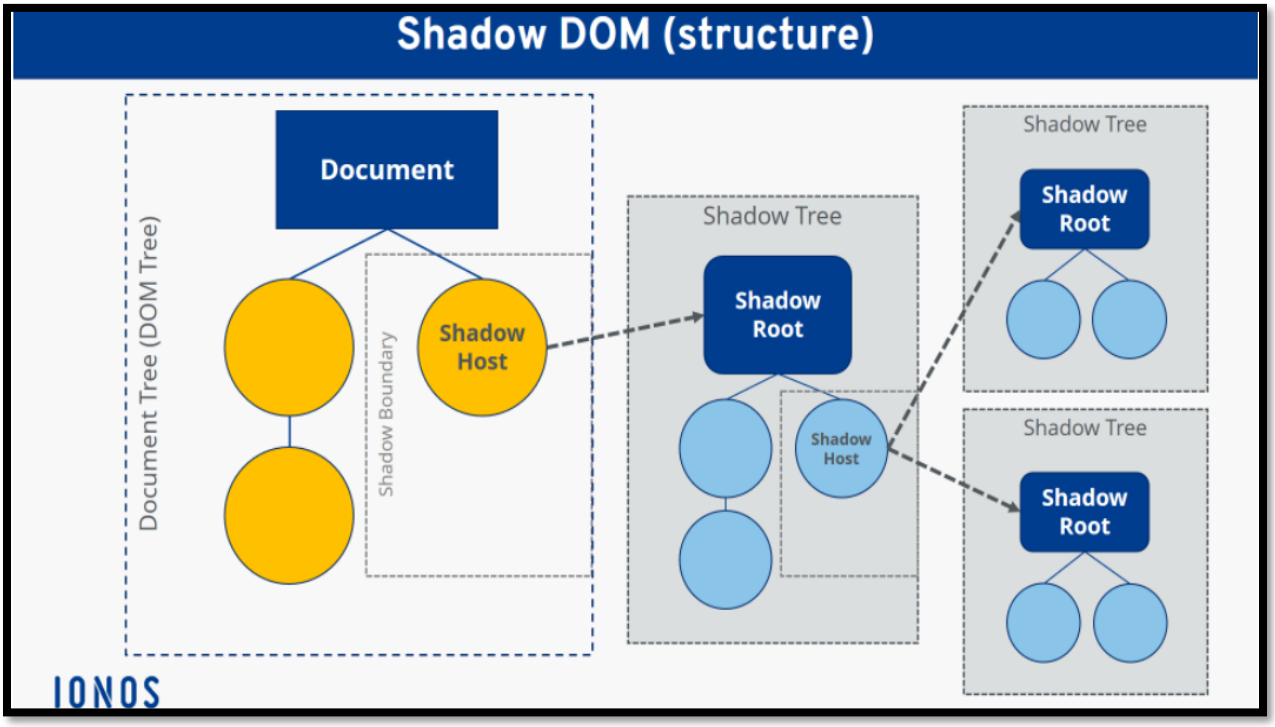
Let me show you how a developer can hide the DOM. On this popular Books page, we can inspect the search field.



Notice above this highlighted section, we see shadow-root 4 times. However, this bread crumb gives us a clue that the search field element is in a Shadow DOM but we only need 1 of these shadow roots. Do

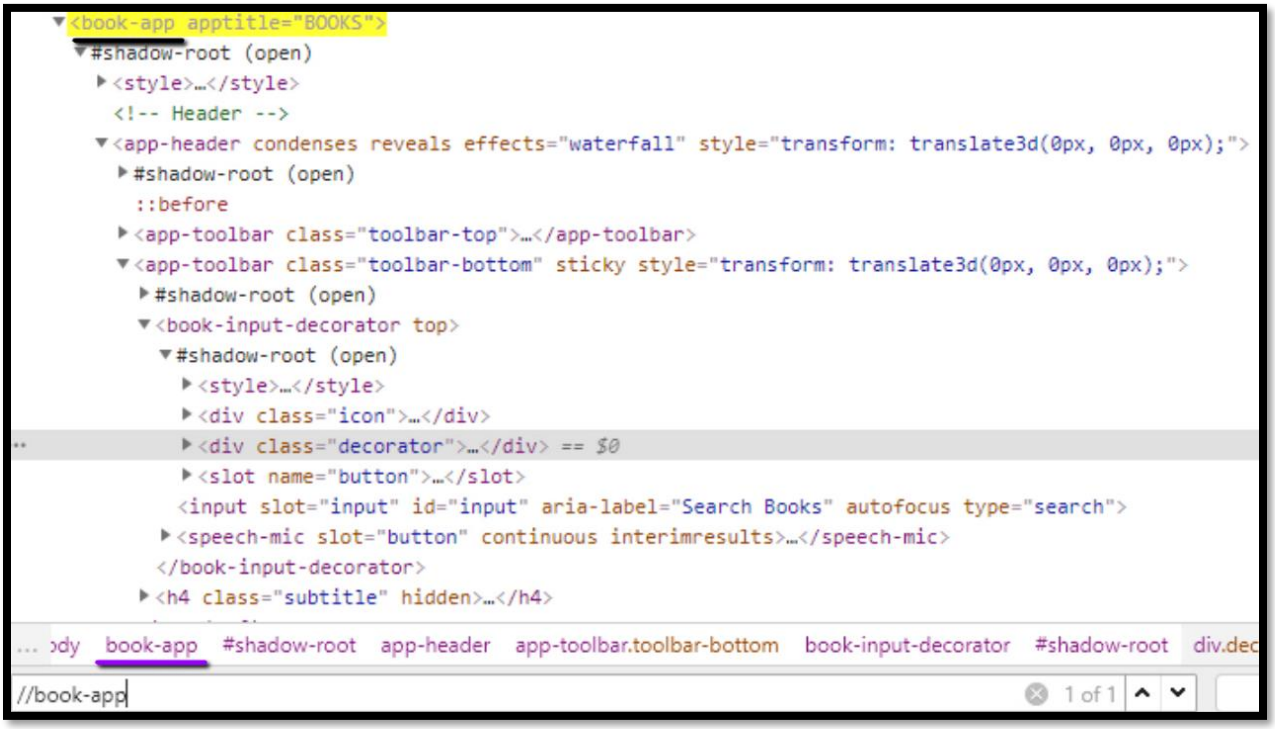
you see hashtag #shadow-root? That is the only shadow-root with an expanded node. The other #shadow-roots are not relevant for locating the search field.

I found this diagram on the internet and it shows the Shadow DOM structure.



We have 1 original DOM and 3 Shadow DOM's. The original DOM has an orange Shadow Host. A Shadow Host is an element with a Shadow Tree. The Shadow Tree is a DOM Tree but that Shadow Tree always begin with a Shadow Root which is similar to the original DOM always begin with html. It has its own DOM tree with elements and a possible Shadow Host. Do you see how the 2nd box which is the 1st Shadow DOM has an extension from the Shadow Host? It has 2 more Shadow Roots. We are not able to locate no elements in the gray boxes because the developers chose to hide those elements within the Shadow DOM. However, we can locate the Shadow Host in the Original DOM.

For example, if I go back to our AUT, we see book-app is the Shadow Host in the Original DOM. Therefore, it's no problem finding this Shadow Host. Search by writing //book-app and we see the element highlighted yellow in the DOM.



Whenever we see Shadow Root that means it is the start of a new Shadow DOM. Watch what happens when I search for app-header by writing //app-header. It's no good; It's no good because the Shadow DOM hid the app-header element. It also hides the app-toolbar element and the book-input-decorator element. I can show you that also. //app-toolbar – did not find it and also look for //book-input-decorator and it did not find it.

From this diagram, we see how an element can be in 1 Shadow DOM or within a nested Shadow DOM. A nested Shadow DOM is when a Shadow DOM is located in another Shadow DOM. Something like the last 2 boxes. There are different way to find an element inside a Shadow DOM. We can find an element without using the Selenium findElement Method. Also, locate an element when using the Selenium findElement Method. I'm going to show you both ways and demo how to locate an element within a nested Shadow DOM.

Rex Jones II

Contact Info

- ✔ Email Rex.Jones@Test4Success.org
- ✔ YouTube https://www.youtube.com/c/RexJonesII/videos
- ✔ Facebook https://facebook.com/JonesRexII
- ✔ Twitter https://twitter.com/RexJonesII
- ✔ GitHub https://github.com/RexJonesII/Free-Videos
- ✔ LinkedIn https://www.linkedin.com/in/rexjones34/