

Instantly share code, notes, and snippets.

Haprog / deepQuerySelectorAll.js



Created 4 years ago

[Code](#) [Revisions](#) 1 [Stars](#) 7 [Forks](#) 1

A version of querySelectorAll() that also recursively looks into all shadow roots

deepQuerySelectorAll.js

```
1  /**
2   * A version of querySelectorAll() that also recursively looks into all shadow roots.
3   * @param selector Selector
4   * @param root (Optional) Scope of the query (Element or Document). Defaults to the document.
5   * @returns
6   */
7  function deepQuerySelectorAll(selector, root) {
8      root = root || document;
9      const results = Array.from(root.querySelectorAll(selector));
10     const pushNestedResults = function (root) {
11         deepQuerySelectorAll(selector, root)
12             .forEach(elem => {
13                 if (!results.includes(elem)) {
14                     results.push(elem);
15                 }
16             });
17     };
18     if (root.shadowRoot) {
19         pushNestedResults(root.shadowRoot);
20     }
21     for (const elem of root.querySelectorAll('*')) {
22         if (elem.shadowRoot) {
23             pushNestedResults(elem.shadowRoot);
24         }
25     }
26     return results;
27 }
```

andrescevp commented on Jan 7, 2024

some enhaces

```
const DeepQuerySelectorAll = (
  selector,
  root,
  maxLevel = 3,
  current = 1,
) => {
  root = root || document;
  const results = Array.from(root.querySelectorAll(selector));
  if (current >= maxLevel) {
    return results;
  }
  const pushNestedResults = (parent) => {
    DeepQuerySelectorAll(selector, parent, maxLevel, current + 1).forEach(
      (elem) => {
        if (!results.includes(elem)) {
          results.push(elem);
        }
      },
    );
  };
  if (root.shadowRoot) {
    pushNestedResults(root.shadowRoot);
  }
  const allElements = root.getElementsByTagName("*");
  Object.values(allElements).forEach((elem) => {
    if (elem.shadowRoot) {
      pushNestedResults(elem.shadowRoot);
    }
  });
  return results;
};
```