Selection API

Design the Composed Range

November 13, 2024
dizhangg@google.com

Define Composed Range

It is a range where the start and end share the same shadow-inclusive root.

Define association between a selection and its range

Current:

Selection <-> Live Range

Option 1: Each selection can be associated with one composed range and each composed range can be associated with one range.

Selection <-> Composed Range <-> Live Range

Option 2: Each selection can be associated with one composed range and one range.

Selection <-> Live Range

Selection <-> Composed Range

getComposedRanges/Anchor/Focus should use Composed Range

getRangeAt(0)/startContainer/startOffset/endContainer/endOffset should use Live Range

Example: getRangeAt(0)

Option 1: Selection <-> Composed Range <-> Live Range

1. If index is not 0 or if this is empty or either focus or anchor is not in the document tree, throw an IndexSizeError exception.
2. If this is associated with a composed range.
    a. If the composed range is associated with a live range, return it.
3. Let newRange be a new range.
4. If the start of this's composed range is not in the document tree, set newRange's start to the first ancestor of start that is in the document tree.
5. If the end of this's composed range is not in the document tree, set newRange's end to the first ancestor of end that is in the document tree.
6. Set this's range to newRange.
7. Return newRange.

Option 2: **Selection <-> Live Range, Selection <-> Composed Range**

1. If index is not 0 or if this is empty or either focus or anchor is not in the document tree, throw an IndexSizeError exception.
2. If this is associated with a range, return it.
3. Let newRange be a new range.
4. If the start of this's composed range is not in the document tree, set newRange's start to the first ancestor of start that is in the document tree.
5. If the end of this's composed range is not in the document tree, set newRange's end to the first ancestor of end that is in the document tree.
6. Set this's range to newRange.
7. Return newRange.

startContainer/startOffset/endContainer/endOffset should all call getRangeAt(0) to find start/end to return.

Example: setBaseAndExtent()

Option 1: Selection <-> Composed Range <-> Live Range

1. Set this composed range's start/end to base/extent.
2. Let newRange be a new range.
   a. SetStart(base), SetEnd(extent), etc.
3. Set this's composed range to newRange.

Option 2: **Selection <-> Live Range, Selection <-> Composed Range**

1. Set this composed range's start/end to base/extent.
2. Let newRange be a new range.
   a. SetStart(base), SetEnd(extent), etc.
3. Set this's composed range to newRange.
4. Set this's range to newRange.

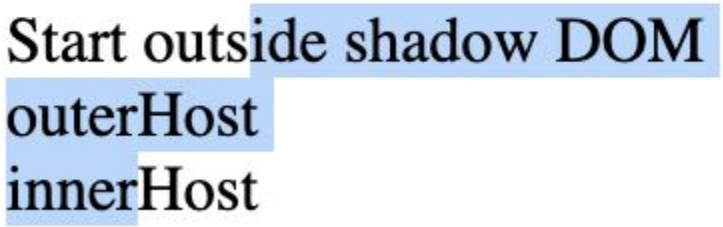
Define effects of Mutations

Example

```
<html>
<body>
<div id="light">Start outside shadow DOM</div>
<div id="outerHost">outerHost
 <template shadowrootmode="open">
   <slot></slot>
   <div id="innerHost">innerHost
    <template shadowrootmode="open">
      <slot></slot>
    </template>
   </div>
 </template>
</div>

<script>
selection = getSelection();
outerRoot = outerHost.shadowRoot;
innerHost = outerRoot.getElementById('innerHost');

selection.setBaseAndExtent(light.firstChild, 10, innerHost.firstChild, 5);
</script>
```



Start outside shadow DOM
outerHost
innerHost

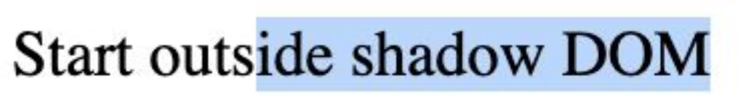
Call outerHost.remove();

start is in light DOM and not affected by mutation.

end is in innerHost. It is a shadow-inclusive descendant, but is not a descendant of outerHost.

Live Range is collapsed because endpoints are in different trees (calling setStart/setEnd).

Composed Range is start {light.firstChild, 10} and end {document, 3} because that’s the position of light element in <body>.



Start outside shadow DOM

Redefine mutations in DOM spec

Update DOM spec by adding a line for composed range

To remove a node *node*, with an optional *suppress observers flag*, run these steps:

4. For each live range whose start node is an inclusive descendant of *node*, set its start to (*parent*, *index*).

   For each composed range whose start node is a shadow-inclusive descendant of node, set its start to (parent, index).

We could also refactor the DOM specification with a new “Update Selection” algorithm.