JavaScript Day - 17

Event Delegation

Event Delegation

Capturing and bubbling allow us to implement one of the most powerful event handling patterns called event delegation.

The idea is that if we have a lot of elements handled in a similar way, then instead of assigning a handler to each of them – we put a single handler on their common ancestor.

In the handler we get event.target to see where the event actually happened and handle it.

Event Delegation

```
<table>
  <tr>
    <th colspan="3"><em>Bagua</em> Chart: Direction, Element, Color, Meaning</th>
  </tr>
  <tr>
    <td class="nw"><strong>Northwest</strong><br>Metal<br>Silver<br>Elders</td>
    <td class="n">...</td>
    <td class="ne">...</td>
  </tr>
  <tr>...2 more lines of this kind...</tr>
  <tr>...2 more lines of this kind...</tr>
</table>
```

Event Delegation

```
let selectedTd;

table.onclick = function(event) {
  let target = event.target; // where was the click?

  if (target.tagName != 'TD') return; // not on TD? Then we're not interested

  highlight(target); // highlight it
};

function highlight(td) {
  if (selectedTd) { // remove the existing highlight if any
    selectedTd.classList.remove('highlight');
  }
  selectedTd = td;
  selectedTd.classList.add('highlight'); // highlight the new td
}
```

Event Delegation

There is one drawback of the previous code

The click may occur not on the <td>, but inside it.

Event Delegation

In the handler table.onclick we should take such event.target and find out whether the click was inside <td> or not.

Here's the improved code:

```
table.onclick = function(event) {
  let td = event.target.closest('td'); // (1)

  if (!td) return; // (2)

  if (!table.contains(td)) return; // (3)

  highlight(td); // (4)
};
```

Event Delegation

- The method elem.closest(selector) returns the nearest ancestor that matches the selector. In our case we look for <td> on the way up from the source element.
- If event.target is not inside any <td>, then the call returns immediately, as there's nothing to do.
- In case of nested tables, event.target may be a <td>, but lying outside of the current table. So we check if that's actually our table's <td>.
- And, if it's so, then highlight it.