Sticky Header / Elements

StyleSheet und JavaScript Fallback

Seit einiger Zeit wird in den Browsern das Verhalten Sticky bereits durch CSS unterstützt, wo bei alten Browserern noch ein JavaScript erforderlich war. Wenn man alte Browser vernachlässigen kann reichen also nur ein paar Zeilen CSS, auch wenn man alte Browser unterstützen möchte reicht ein kleines JavaScript. Wo im JavaScript noch jeweils ein Eventlistener auf Scroll und Resize benötigt wird, hat man in der nativen CSS Variante natürlich in Sachen Performance einiges gewonnen.

Das Setzen des Attributes position: sticky und der Angabe des Wertes für top an dem das Element "einrasten" soll reich bereits aus, alternativ wird dem Element im Augenblick wenn es das Verhalten "Sticky" einnehmen soll, also wiederum an der in top angegeben Position, die Klasse sticky hinzugefügt, wird diese Position unterschritten, wird natürlich diese Klasse auch wieder entfernt.

StyleSheet:

```
@media screen {
  #toolbar {
    position: sticky;
    position: -webkit-sticky;
    top: 0;
  }
  #toolbar.sticky {
    position: fixed;
  }
}
@media screen and (min-width:768px) {
  #nav-container {
    position: sticky;
    position: -webkit-sticky;
    top: 40px;
  }
  #nav-container.sticky {
    position: fixed;
  }
}
```

Zur Identifizierung des Elementes oder der Elemente welches ich als sticky behandeln möchte, füge ich dem Tag das Attribut data-type="sticky" hinzu.

JavaScript:

sticky.js JavaScript (1,65 kByte) 12.12.2021 15:53

```
// coding: utf-8
/** Created by: Udo Schmal | https://www.gocher.me/ */
// sticky element
// css replacement
// #toolbar {position: sticky; position: -webkit-sticky; top: 0;}
// #toolbar.sticky {position: fixed;}
```

```
// #nav-container {position: sticky; position: -webkit-sticky; top: 40px;}
// #nav-container.sticky {position: fixed;}
(function() {
  'use strict';
  var el = document.querySelector("[data-type='sticky']");
  if (el) {
    var positionSticky =
window.getComputedStyle(el).getPropertyValue("position");
    if (positionSticky != "sticky") {
      var useCapture = false;
      try {
        var opts = Object.defineProperty({}, 'passive', {
          get: function() {
            useCapture = {capture: false, passive: true};
          }
        });
        window.addEventListener("testPassive", null, opts);
        window.removeEventListener("testPassive", null, opts);
      } catch (e) {}
      var scrollPos = 0;
      var marginTop = - window.getComputedStyle(el).getPropertyValue("top");
      function getScrollPos() {
        pos = 0;
        var i = el;
        do {
          pos += i.offsetTop;
          i = i.offsetParent;
        } while (i);
        pos += marginTop;
        if (pos != 0) {
          scrollPos = pos;
        }
      }
      function stickyElement(){
        if (window.scrollY > scrollPos) {
          el.classList.add('sticky');
        } else {
          el.classList.remove('sticky');
        }
      }
      window.addEventListener('scroll', stickyElement, useCapture);
      window.addEventListener('resize', function () { getScrollPos();
stickyElement();}, useCapture);
      getScrollPos();
      stickyElement();
    }
  }
})();
```

Autor: Udo Schmal, veröffentlicht: 29.06.2020, letzte Änderung: 29.08.2024