

Natürlich sollen unsere Webseiten auch auf mobilen Geräten gut funktionieren. Dafür gibt es Touch Events für berührungssensitive Touchscreens. Sie ähneln den Mouse-Events.

touchstart Beginn der Berührung ⇔ mousedown

JS `objekt.addEventListener("touchstart", meineFunktion);`

HTML `<div ontouchstart="meineFunktion()">`

touchmove Wischen ⇔ mousemove, pointermove

JS `objekt.addEventListener("touchmove", meineFunktion);`

HTML `<div ontouchmove="meineFunktion()">`

touchend Ende der Berührung ⇔ mouseup, pointerup

JS `objekt.addEventListener("touchend", meineFunktion);`

HTML `<div ontouchend="meineFunktion()">`

touchcancel Abbruch der Berührung ⇔ mouseup, pointerup

JS `objekt.addEventListener("touchcancel", meineFunktion);`

HTML `<div ontouchcancel="meineFunktion()">`

JS `.targetTouches`



Die `.targetTouches` Eigenschaft liefert eine TouchList mit Touch Objekten für jeden Finger der das aktuelle Element berührt.

z. B. `.targetTouches[0]` ← für den ersten Finger

Das Beispiel ermittelt, wieviel Finger ein `<div>` berühren.

```
<div ontouchstart="meinTouch(event)">
<p id="ausgabe" style="height: 500px; background: green;"></p></div>

<script>
  function meinTouch(event) {
    var wieViele = "Anzahl der Finger: ";
    wieViele += event.targetTouches.length;
    document.getElementById("ausgabe").innerHTML = wieViele;
  }
</script>
```

JS `.touches` ← Liefert eine TouchList mit aktuellen Berührungen am Screen



Das Beispiel ermittelt die Koordinaten des Touch Events.

```
<body ontouchstart="zeigeXY(event)" ontouchmove="zeigeXY(event)">
<div id="anzeige" style="height:400px; background:red"></div>

<script>
  function zeigeXY(event) {
    var x = event.touches[0].clientX;
    var y = event.touches[0].clientY;
    document.getElementById("anzeige").innerHTML = x + ", " + y;
  }
</script>
```