

Die Audio bzw. Video Events können als Attribut im `<audio>` und `<video>` Element vorkommen oder als Eventlistener in Form einer Objekt-Vereinbarung. Selbstverständlich feuern auch die meisten anderen Events (z. B. Maus, Keyboard, Touch usw.) mit den AV-Elementen. Hier ein kleiner Auszug der Audio-Video-Events:

loadstart Zu Beginn des Ladeprozess, wenn der Browser nach dem AV-Element sucht!

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

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

progress Sobald der Download beginnt

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

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

canplay Sobald das Video geladen ist und genug Buffer hat, um es zu starten!

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

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

waiting Feuert, wenn das Video-Frame gebuffert werden muss.

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

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

playing Wenn nach einer Pause (bzw. gestoppt wegen Bufferung) wieder abgespielt wird.

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

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

seeked Wenn der User die Position des Videos verändert hat.

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

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



Beispiel seeked: Sobald der User die Zeit-Position im Video verändert hat, wird entweder 1. Abschnitt oder 2. Abschnitt im `<h1>` Element ausgegeben.

```
<h1 id="status">Croissants backen!</h1>
<video controls src="croissants.mp4" id="meinVideo"></video>

<script>
  var dasVideoElement = document.getElementById("meinVideo");
  dasVideoElement.addEventListener("seeked", meineFunktion);

  function meineFunktion() {
    if (dasVideoElement.currentTime <= 30) {
      var ausgabeText = "1. Abschnitt";
    } else {var ausgabeText = "2. Abschnitt";}
    document.getElementById("status").innerHTML = ausgabeText;
  }
</script>
```