

Die Geolocation API liefert die geographische Position eines Users bzw. Device. Dabei greift die API auf IP-Basis, WLAN-Netzwerkdaten, Funk-Signale oder GPS zu und ermittelt so die Position des Device (z. B. Tablett, Smartphone).

Wegen dem Datenschutz wird er User oft aufgefordert, den Zugriff auf den Standort zu erlauben. Darüber hinaus erlauben moderne Browser den Zugriff auf den Standort nur in einem "sicheren Kontext" (d. H. HTTPs, lokale Dateien, localhost).

Geolocation ist ein Objekt von `window.navigator`. → `navigator.geolocation`.

JS

`.getCurrentPosition(zeigePositionFunktion, zeigeFehlerFunktion);`



Im Beispiel wird zuerst geprüft, ob der Browser das `.geolocation` Objekt unterstützt. Ist diese Prüfung `true`, dann ruft die `.getCurrentPosition()` Methode eine Funktion auf welche die Positionsdaten übergibt und eine Funktion die etwaige Fehler auswertet.

```
function ermittlePosition() {
    if (navigator.geolocation) {
        navigator.geolocation.getCurrentPosition(zeigePosition, zeigeFehler);
    } else {alert("Ihr Browser unterstützt keine Geolocation.");}
```

JS

`.coords`.



Die `zeigePosition` Funktion übernimmt die Positionsdaten welche über die `.coords` Eigenschaft ausgelesen werden können. Folgende Eigenschaften können ausgelesen werden:

| | |
|---------------------------------------|--|
| <code>.coords.latitude</code> | Breitenangabe als Dezimalzahl |
| <code>.coords.longitude</code> | Längenangabe als Dezimalzahl |
| <code>.coords.accuracy</code> | Genauigkeit der Koordinaten (in Meter) |
| <code>.coords.altitude</code> | Höhenangabe (über dem Meeresspiegel) |
| <code>.coords.altitudeAccuracy</code> | Genauigkeit der Höhenangabe |
| <code>.coords.heading</code> | Richtung |
| <code>.coords.speed</code> | Geschwindigkeit (in m/s) |
| <code>.timestamp</code> | Zeit der Positionsangabe |

```
function zeigePosition(position) {
    alert("Breite: " + position.coords.latitude);
    alert("Länge: " + position.coords.longitude);
    alert("Höhe: " + position.coords.altitude); }
```



Etwaige Fehler werden in einer Funktion abgefangen und über eine Switch-Verzweigung ausgewertet.

```
function zeigeFehler(fehler) {
    switch(fehler.code) {
        case fehler.PERMISSION_DENIED:
            alert("Benutzer lehnte Standortabfrage ab."); break;
        case fehler.POSITION_UNAVAILABLE:
            alert("Standortdaten sind nicht verfügbar."); break;
        case fehler.TIMEOUT:
            alert("Die Standortabfrage dauerte zu lange."); break;
        case fehler.UNKNOWN_ERROR:
            alert("unbekannter Fehler."); break;}}
```