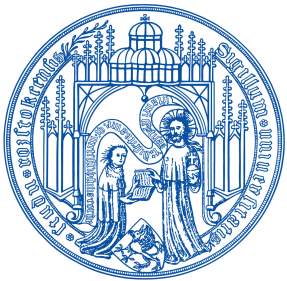
Web Speech API

Native Sprachverarbeitung im Browser

Clemens H. **Cap**
ORCID: 0000-0003-3958-6136

Department of Computer Science
University of **Rostock**
Rostock, Germany
clemens.cap@uni-rostock.de



2026-06-24 16:01

Vorlesung Web 2.0

Gliederung

1. Einführung

Sprachschnittstellen und ihre Bausteine.

Motivation: Warum Sprachschnittstellen?

Sprache: Natürlichste Form menschlicher Kommunikation.
Im Web: Zugang jenseits von Tastatur und Maus.

Zentrale Vorteile

- **Barrierefreiheit:** Bedienung ohne Hände für Menschen mit motorischen oder visuellen Einschränkungen.
- **Effizienz:** Diktat ist oft schneller als Tippen, gerade auf mobilen Geräten.
- **Kontext:** Freihändige Nutzung in Küche, Auto oder Werkstatt.

Die **Web Speech API** bringt diese Fähigkeiten direkt in den Browser
Keine Zusatzsoftware oder Plug-ins erforderlich.

Was ist Spracherkennung?

Spracherkennung (Speech Recognition, STT) wandelt gesprochenes Audio in Text um.

- Eingabe: Audiosignal vom Mikrofon oder einer Tonspur.
- Erkennungsdienst: Analysiert Signal, liefert eine oder mehrere Textvarianten.
- Jede Variante trägt einen Vertrauenswert (**confidence**).

Typische Anwendungsfälle

Diktat, Sprachbefehle, Suchabfragen per Stimme und Echtzeit-Untertitel.

Was ist Sprachsynthese?

Sprachsynthese (Speech Synthesis, TTS) erzeugt aus Text hörbare Sprache.

- Ein Synthesizer erzeugt aus Zeichenketten ein Audiosignal.
- Je nach System und Sprache verschiedene **Stimmen** verfügbar.
- Tonhöhe, Tempo und Lautstärke einstellbar.

Typische Anwendungsfälle

Vorlesefunktionen, akustische Rückmeldungen und Assistenzsysteme.

Erkennung und Synthese im Überblick

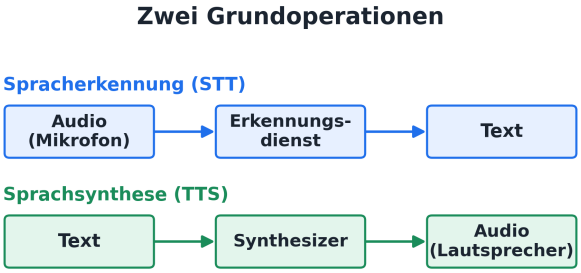


Abb. 1: Die zwei Grundoperationen: Erkennung wandelt Audio in Text (oben), Synthese wandelt Text in Audio (unten). Beide Vorgänge sind zueinander invers. Rechte s. Anhang.

2. Überblick

Aufbau und Einstieg in Web Speech API.

Was ist die Web Speech API?

JavaScript-Schnittstelle, die Spracherkennung und Sprachsynthese im Browser verfügbar macht.

- **Draft Community Group Report** der W3C Audio Community Group [W3C25].
- Keine eigene Erkennungs-Engine, sondern Einbindung der Dienste des Betriebssystems oder lokaler Modelle.
- Getrennte Teilbereiche: Erkennung und Synthese.

Wichtiger Hinweis

Das frühere Konzept der **Grammatik** (`SpeechGrammar`) wurde aus der Spezifikation entfernt und hat keine Wirkung mehr.

Architektur: Zwei Schnittstellen

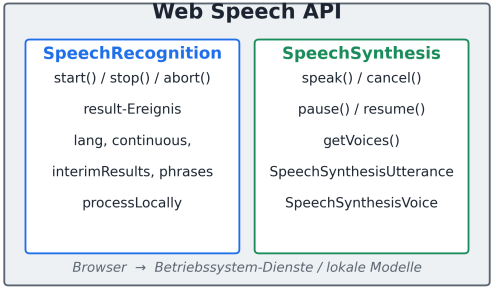


Abb. 2: Zwei unabhängige Controller-Objekte: `SpeechRecognition` für die Erkennung und `SpeechSynthesis` für die Synthese, jeweils mit ihren wichtigsten Mitgliedern. Rechte s. Anhang.

Beispiel: Minimaler Einstieg

Vor der Nutzung empfiehlt sich **Feature-Detection**
Nicht jeder Browser unterstützt beide Teilbereiche.

```
// Erkennung: Konstruktor je nach Browser auswaehlen
const SR = window.SpeechRecognition
        || window.webkitSpeechRecognition;
if (SR) {
  const rec = new SR();
}

// Synthese: ueber das globale Objekt speechSynthesis
const u = new SpeechSynthesisUtterance("Hallo Welt");
speechSynthesis.speak(u);
```

Src. 1: Feature-Detection der Erkennung und einfache Sprachausgabe. Rechte s. Anhang.

3. Sprachsynthese

Text in Sprache umwandeln (TTS)

Die Schnittstelle SpeechSynthesis

Einstiegspunkt ist globales Objekt `speechSynthesis`
Verwaltet Warteschlange auszusprechender Äußerungen.

Wichtige Methoden

- `speak(utterance)` reiht eine Äußerung ein.
- `cancel()` leert die Warteschlange sofort.
- `pause()` & `resume()` halten an und setzen fort.
- `getVoices()` liefert verfügbaren Stimmen.

Zustands-Eigenschaften

- `speaking`, `pending` und `paused` beschreiben den Zustand der Warteschlange.

SpeechSynthesisUtterance

`SpeechSynthesisUtterance` kapselt vorzulesenden Text samt aller Parameter.

Steuerbare Eigenschaften	
text	Der auszusprechende Inhalt.
lang	Sprache als BCP-47-Kennung, etwa `de-DE`.
voice	Eine konkrete `SpeechSynthesisVoice`.
rate	Tempo (Standard 1, Bereich 0,1 bis 10).
pitch	Tonhöhe (Standard 1, Bereich 0 bis 2).
volume	Lautstärke (Bereich 0 bis 1).

Schnittstelle breit verfügbar (**Baseline**) und produktiv einsetzbar [MDN25b].

Stimmen verwalten

Liste der Stimmen oft **asynchron** geladen.
Auf Ereignis `voiceschanged` reagieren.

```
1  let voices = speechSynthesis.getVoices();
2
3  speechSynthesis.onvoiceschanged = () => {
4    voices = speechSynthesis.getVoices();
5    // z. B. nur deutsche Stimmen anbieten:
6    const deutsch = voices.filter(v => v.lang === "de-DE");
7  };
```

Src. 2: Nachladen der Stimmen über das Ereignis `voiceschanged`. Rechte s. Anhang.

Ereignisse der Sprachausgabe

Äußerung durchläuft mehrere Ereignisse.
erlaubt Synchronisation mit dem UI.

Ereignisse einer Utterance	
start / end	Beginn und Ende der Ausgabe.
pause / resume	Anhalten und Fortsetzen.
boundary	Erreichen einer Wort- oder Satzgrenze; ideal zum synchronen Hervorheben des Textes.
error	Ein Fehler verhindert die Ausgabe.

`boundary`-Ereignis nützlich zur mitlaufenden optischen Markierung einzelner Wörter.

Beispiel: Text vorlesen

Beispiel liest deutschen Satz mit angepaßtem Tempo und gewählter Stimme vor.

```
1  const u = new SpeechSynthesisUtterance(
2    "Guten Tag, willkommen zur Vorlesung.");
3  u.lang = "de-DE";
4  u.rate = 0.95;
5
6  const voices = speechSynthesis.getVoices();
7  u.voice = voices.find(v => v.lang === "de-DE");
8
9  u.onend = () => console.log("Ausgabe beendet");
10 speechSynthesis.speak(u);
```

Src. 3: Vollständige Sprachausgabe mit Sprache, Tempo und Stimme. Rechte s. Anhang.

4. Spracherkennung

Sprache in Text umwandeln (STT)

Die Schnittstelle SpeechRecognition

`SpeechRecognition` ist das Controller-Objekt der Erkennung.
Erzeugung über einen Konstruktor.

Methoden

- `start()` beginnt das Zuhören.
- `stop()` beendet es und liefert das letzte Ergebnis.
- `abort()` bricht ab, ohne ein Ergebnis zu liefern.

Voraussetzungen

Erkennung erfordert die Erlaubnis zur Mikrofonnutzung und einen sicheren Kontext (HTTPS).
Standardmäßig wird Audio an einen Dienst im OS übertragen [MDN25a].

Konfiguration der Erkennung

Vor dem Start Steuerung des Verhaltens der Erkennung über mehrere Eigenschaften.

Wichtige Eigenschaften	
lang	Erkennungssprache, etwa `de-DE`.
continuous	Fortlaufende Erkennung statt eines Einzelergebnisses.
interimResults	Liefert vorläufige Zwischenergebnisse, bevor Endergebnis feststeht.
maxAlternatives	Maximale Anzahl alternativer Erkennungsvarianten je Ergebnis.

Vorläufige Ergebnisse verbessern die wahrgenommene Reaktionszeit
da Text schon während des Sprechens erscheint.

Der Ereignis-Lebenszyklus

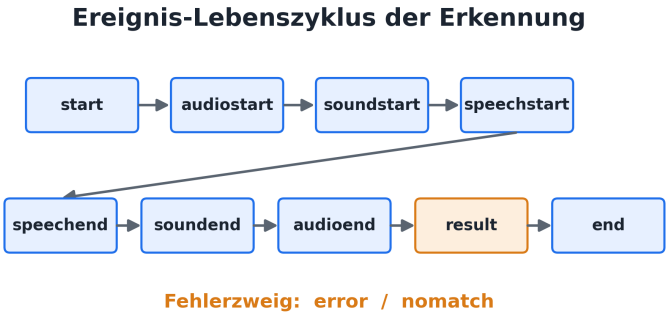


Abb. 3: Geordnete Ereignisfolge einer Erkennung von `start` bis `end`; `result` trägt das eigentliche Ergebnis, `error` und `nomatch` bilden den Fehlerzweig. Rechte s. Anhang.

Ergebnisse auswerten

`result`-Ereignis enthält eine verschachtelte Struktur.
Jedes Ergebnis bietet Alternativen mit Text und Vertrauenswert.

```
1  rec.onresult = (event) => {
2    const result = event.results[0];  // erstes Ergebnis
3    const best   = result[0];         // beste Alternative
4    console.log(best.transcript);     // erkannter Text
5    console.log(best.confidence);     // Vertrauenswert
6    console.log(result.isFinal);      // endgueltig?
7  };
```

Src. 4: Zugriff auf Transkript, Vertrauenswert und `isFinal`. Rechte s. Anhang.

Contextual Biasing mit phrases

Mit der Eigenschaft `phrases` lassen sich domänenspezifische Begriffe gezielt bevorzugen. Beispiel: Eigennamen oder Fachwörter.

```
rec.phrases = [
  new SpeechRecognitionPhrase("Anthropic", 2.0),
  new SpeechRecognitionPhrase("Beamer", 1.5)
];
```

Src. 5: Der zweite Parameter ist ein Boost-Faktor, der die Erkennungswahrscheinlichkeit erhöht und das entfernte Grammatik-Konzept ablöst. Rechte s. Anhang.

Fehlerbehandlung

Robuste Anwendung reagiert auf Fehler und auf fehlende Treffer.
Ziel: Nutzer nicht im Unklaren zu lassen.

Relevante Ereignisse

error	Liefert einen Fehlercode, etwa `no-speech`, `not-allowed` oder `network`.
nomatch	Es wurde gesprochen, aber kein Treffer erreichte den nötigen Vertrauenswert.
end	Wird stets gefeuert und eignet sich zum Aufräumen der Oberfläche.

Code `not-allowed` weist auf eine verweigerte Mikrofon-Berechtigung hin. Muß klar an Nutzer kommuniziert werden.

Beispiel: Sprachbefehl erkennen

Das Beispiel erkennt einen einzelnen deutschen Befehl und gibt das Ergebnis aus.

```
const SR = window.SpeechRecognition
        || window.webkitSpeechRecognition;
const rec = new SR();
rec.lang = "de-DE";
rec.continuous = false;
rec.maxAlternatives = 1;

rec.onresult = (e) => {
  const text = e.results[0][0].transcript;
  console.log("Erkannt:", text);
};
rec.onerror = (e) => console.warn("Fehler:", e.error);
rec.start();
```

Src. 6: Erkennung eines einzelnen Befehls mit Fehlerbehandlung. Rechte s. Anhang.

5. On-Device-Verarbeitung

Lokal, privat und offline

Cloud- versus On-Device-Erkennung

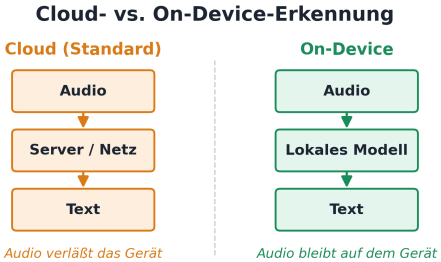


Abb. 4: Bei der Standard-Erkennung wird Audio an einen Dienst gesendet; bei On-Device-Erkennung verlassen weder Audio noch Transkript das Gerät. Vorteile: Datenschutz, geringere Latenz und Offline-Fähigkeit [Web25].
Rechte s. Anhang.

processLocally aktivieren

Eigenschaft `processLocally` erzwingt die lokale Verarbeitung.
Wenn nicht möglich: Fehlermeldung, keine weitere Verarbeitung.

```
1 const rec = new SpeechRecognition();
2 rec.lang = "de-DE";
3 rec.processLocally = true;    // Audio bleibt auf dem Geraet
4 rec.start();
```

Src. 7: Standardwert ist `false`; lokale Erkennung benötigt ein installiertes Sprachpaket. Rechte s. Anhang.

Verfügbarkeit prüfen und installieren

Klären, ob ein Sprachpaket vorhanden ist.
Bei Bedarf Installation

```
const status = await SpeechRecognition.available({
  langs: ["de-DE"], processLocally: true
});
// "available"|"downloadable"|"downloading"|"unavailable"

if (status === "downloadable") {
  await SpeechRecognition.install({ langs: ["de-DE"] });
}
```

Src. 8: `available()` liefert einen Statuswert, `install()` ein `Promise<boolean>` über den Erfolg. Rechte s. Anhang.

Permissions-Policy und Datenschutz

Zugriff auf die On-Device-Funktionen ist über die **Permissions-Policy** abgesichert.

Sicherheitsmodell

- Die Methoden `available()` und `install()` unterliegen der Richtlinie `on-device-speech-recognition`.
- Standard-Erlaubnisliste ist `'self'`, also nur das eigene Dokument.
- Nachladen großer Sprachpakete kann Zustimmung des Nutzers erfordern.

Stellt sicher, daß lokale Modelle nicht unbemerkt von fremden Einbettungen ausgelöst werden [Web25].

6. Praxis

Browser-Support, Sicherheit und Anwendungen

Browser-Unterstützung

Synthese und Erkennung sind unterschiedlich weit verbreitet.
Synthese gilt als **Baseline**, Erkennung nicht.

Funktion	Chrome	Edge	Safari	Firefox
Sprachsynthese (TTS)	Ja	Ja	Ja	Ja
Spracherkennung (STT)	Ja	Ja	Teilweise	Nein
On-Device (`processLocally`)	Ja (ab 139)	Experimentell	Nein	Nein

Tab. 1: Browser-Unterstützung der Web Speech API (qualitativer Überblick, Stand Juni 2026). Rechte s. Anhang.

On-Device-Erkennung: **Chrome** seit Version 139, **Edge** in Erprobung [Mic26].
Feature-Detection und sinnvoller Rückfall notwendig.

Sicherheit und Berechtigungen

Sprachfunktionen berühren sensible Daten und unterliegen daher strengen Voraussetzungen im Browser.

Regeln für den produktiven Einsatz

- **HTTPS:** Die Erkennung läuft nur in einem sicheren Kontext.
- **Erlaubnis:** Der Nutzer muß die Mikrofonnutzung ausdrücklich gestatten.
- **Transparenz:** Ein sichtbarer Hinweis sollte zeigen, wann zugehört wird.
- **Datensparsamkeit:** Für vertrauliche Inhalte empfiehlt sich `processLocally`.

Anwendungsfälle und Barrierefreiheit

Web Speech API entfaltet ihren Nutzen vor allem dort, wo Sprache der natürlichere Kanal ist.

Typische Einsatzgebiete

- Freihändige Bedienung und Sprachbefehle.
- Vorlesefunktionen für längere Texte und Artikel.
- Diktat in Formulare und Notizen.
- Sprachlern-Anwendungen mit Aussprache-Rückmeldung.

Barrierefreiheit als Leitgedanke

Sprachausgabe und -eingabe ergänzen unterstützende Technologien,
ersetzen sie aber nicht.
Alternative Eingabewege müssen erhalten bleiben.

Fallstudie: Speech color changer

Das offizielle MDN-Beispiel **Speech color changer** zeigt die Erkennung in einer kompakten, lauffähigen Anwendung.

Was die Demo zeigt

- Ein Klick startet die Erkennung über `start()`.
- Im `result`-Ereignis wird der gesprochene Farbname entnommen.
- Der Hintergrund der Seite wechselt auf die genannte Farbe.

Quelltext und Live-Demo: mdn.github.io/dom-examples.
Gut geeignet als Ausgangspunkt für eigene Experimente.

7. Zusammenfassung

Kernpunkte und Ausblick

Zusammenfassung

Die wichtigsten Punkte

- Web Speech API umfaßt: `SpeechSynthesis` und `SpeechRecognition`.
- Synthese steuert Text, Stimme, Tempo und Tonhöhe. ist breit verfügbar.
- Erkennung liefert Texte mit Vertrauenswerten und Alternativen.
- `phrases` bevorzugt gezielt Fachbegriffe.
- `processLocally` ermöglicht private, offline-fähige Erkennung auf dem Gerät.

Ausblick

API entwickelt sich aktiv weiter.
Schwerpunkt liegt auf lokaler Verarbeitung und besserer Erkennungsqualität.

- **On-Device:** Ausweitung auf mehr Browser, Sprachen und Plattformen.
- **Contextual Biasing:** Feinere Steuerung der Erkennung über gewichtete Begriffe.
- **Standardisierung:** Annäherung der Implementierungen an die gemeinsame Spezifikation [W3C25].

Empfehlung

Heute produktiv nutzbar ist vor allem die Synthese.
Erkennung sollte stets mit Feature-Detection und Rückfallweg umgesetzt werden.

Weiterführende Ressourcen

Spezifikation	W3C Web Speech API [W3C25], webaudio.github.io.
Referenz	MDN Web Docs [MDN25b], developer.mozilla.org.
Leitfaden	MDN: Using the Web Speech API [MDN25a].
On-Device	Explainer der WebAudio Community Group [Web25].
Kompatibilität	Can I use, caniuse.com.

Appendix

Übersicht

Literaturverzeichnis

[MDN25a] MDN Web Docs. *Using the Web Speech API*. Abgerufen am 10. Juni 2026. 2025. URL: https://developer.mozilla.org/en-US/docs/Web/API/Web_Speech_API/Using_the_Web_Speech_API (cit. on pp. 19, 39).

[MDN25b] MDN Web Docs. *Web Speech API*. Abgerufen am 10. Juni 2026. 2025. URL: https://developer.mozilla.org/en-US/docs/Web/API/Web_Speech_API (cit. on pp. 14, 39).

[Mic26] Microsoft Edge Team. *Expanding on-device AI in Microsoft Edge: New models and APIs for the web*. Microsoft Edge Blog. Abgerufen am 10. Juni 2026. 2026. URL: https://blogs.windows.com/msedgedev/2026/06/02/expanding-on-device-ai-in-microsoft-edge-new-models-and-apis-for-the-web/ (cit. on p. 32).

[W3C25] W3C Audio Community Group. *Web Speech API*. Draft Community Group Report. Abgerufen am 10. Juni 2026. 2025. URL: https://webaudio.github.io/web-speech-api/ (cit. on pp. 9, 38, 39).

[Web25] WebAudio Community Group. *On-device speech recognition (Explainer)*. Abgerufen am 10. Juni 2026. 2025. URL: https://github.com/WebAudio/web-speech-api/blob/main/explainers/on-device-speech-recognition.md (cit. on pp. 27, 30, 39).

Programmquellenverzeichnis

Verzeichnis aller Abbildungen

Verzeichnis aller Tabellen

Rechtsnachweise

Rechtliche Hinweise (1)

Die hier angebotenen Inhalte unterliegen deutschem Urheberrecht. Inhalte Dritter werden unter Nennung der Rechtsgrundlage ihrer Nutzung und der geltenden Lizenzbestimmungen hier angeführt. Auf das Literaturverzeichnis wird verwiesen. Das Zitatrecht in dem für wissenschaftliche Werke üblichen Ausmaß wird beansprucht. Wenn Sie eine Urheberrechtsverletzung erkennen, so bitten wir um Hinweis an den auf der Titelseite genannten Autor und werden entsprechende Inhalte sofort entfernen oder fehlende Rechtsnennungen nachholen. Bei Produkt- und Firmennamen können Markenrechte Dritter bestehen. Verweise und Verlinkungen wurden zum Zeitpunkt des Setzens der Verweise überprüft; sie dienen der Information des Lesers. Der Autor macht sich die Inhalte, auch in der Form, wie sie zum Zeitpunkt des Setzens des Verweises vorlagen, nicht zu eigen und kann diese nicht laufend auf Veränderungen überprüfen.
Alle sonstigen, hier nicht angeführten Inhalte unterliegen dem Copyright des Autors, Prof. Dr. Clemens Cap, ©2020. Wenn Sie diese Inhalte nützlich finden, können Sie darauf verlinken oder sie zitieren. Jede weitere Verbreitung, Speicherung, Vervielfältigung oder sonstige Verwertung außerhalb der Grenzen des Urheberrechts bedarf der schriftlichen Zustimmung des Rechteinhabers. Dieses dient der Sicherung der Aktualität der Inhalte und soll dem Autor auch die Einhaltung urheberrechtlicher Einschränkungen wie beispielsweise Par 60a UrhG ermöglichen.
Die Bereitstellung der Inhalte erfolgt hier zur persönlichen Information des Lesers. Eine Haftung für mittelbare oder unmittelbare Schäden wird im maximal rechtlich zulässigen Ausmaß ausgeschlossen, mit Ausnahme von Vorsatz und grober Fahrlässigkeit. Eine Garantie für den Fortbestand dieses Informationsangebots wird nicht gegeben.
Die Anfertigung einer persönlichen Sicherungskopie für die private, nicht gewerbliche und nicht öffentliche Nutzung ist zulässig, sofern sie nicht von einer offensichtlich rechtswidrig hergestellten oder zugänglich gemachten Vorlage stammt.
Use of Logos and Trademark Symbols: The logos and trademark symbols used here are the property of their respective owners. The YouTube logo is used according to brand request 2-9753000030769 granted on November 30, 2020. The GitHub logo is property of GitHub Inc. and is used in accordance to the GitHub logo usage conditions https://github.com/logos to link to a GitHub account. The Tweedback logo is property of Tweedback GmbH and here is used in accordance to a cooperation contract.

Rechtliche Hinweise (2)

Disclaimer: Die sich immer wieder ändernde Rechtslage für digitale Urheberrechte erzeugt ein nicht unerhebliches Risiko bei der Einbindung von Materialien, deren Status nicht oder nur mit unverhältnismäßig hohem Aufwand abzuklären ist. Ebenso kann den Rechteinhabern nicht auf sinnvolle oder einfache Weise ein Honorar zukommen, obwohl deren Leistungen genutzt werden.
Daher binde ich gelegentlich Inhalte nur als Link und nicht durch Framing ein. Lt EuGH Urteil 13.02.2014, C-466/12 (Pressemitteilung, Blog-Beitrag, Urteilstext). ist das unbedenklich, da die benutzten Links ohne Umgehung technischer Sperren auf im Internet frei verfügbare Inhalte verweisen.
Wenn Sie diese Rechtslage stört, dann setzen Sie sich für eine Modernisierung des völlig veralteten Vergütungs- und Anreizsystems für urheberrechtliche Leistungen ein. Bis dahin klicken Sie bitte auf die angegebenen Links und denken Sie darüber nach, warum wir keine für das digitale Zeitalter sinnvoll angepaßte Vergütungs- und Anreizsysteme digital erbrachter Leistungen haben.
Zu Risiken und Nebenwirkungen fragen Sie Ihren Rechtsanwalt oder Gesetzgeber.
Weitere Hinweise finden Sie im Netz hier und hier oder hier.

Zitierweise dieses Dokuments

Wenn Sie Inhalte aus diesem Werk nutzen oder darauf verweisen wollen,
zitieren Sie es bitte wie folgt:

Clemens H. Cap: Web Speech API. Electronic document. https://iuk.one/203-0802 24. 6. 2026.

Bibtex Information: https://iuk.one/203-0802.bib

```
@misc{doc:203-0802,
    author        = {Clemens H. Cap},
    title         = {Web Speech API},
    year          = {2026},
    month         = {6},
    howpublished  = {Electronic document},
    url           = {https://iuk.one/203-0802}
}
```

Typographic Information:

Typeset on 2026-06-24 at 16:01
This is pdfTeX, Version 3.141592653-2.6-1.40.29 (TeX Live 2026) kpathsea version 6.4.2
This is pgf in version 3.1.11a
This is preamble-slides.tex ©C.H.Cap
Jobtype flag=foreground
Active modes (if any): index, biblioKeywords: keywords
Subject: subject

Abkürzungsverzeichnis

Verzeichnis aller Folien

Legende:
Fortsetzungsseite
Seite ohne Überschrift
Bildseite