# Praktikum 3

## Aufgabe 5

1. (wie intersection) – Mengenoperationen

|  |  |  |  |
| --- | --- | --- | --- |
|  | JavaScriptCore | V8 | SpiderMonkey |
| Set.prototype.\* | 16.4 (April 2023) | 10.9 (Dezember 2022) | 106 (September 2022) |
| Static Blocks | 16.0 (Oktober 2022) | 9.8 (Juni 2021) | Version 91 (August 2021) |
| Array.prototype.flat(depth) | 2018 (Safari 12) | 2018 (Chrome 69) | 2019 (Firefox 62) |
| Array.prototype.findLast | 15.4 (Dezember 2021) | 9.6 (Oktober 2021) | 91 (August 2021) |
| Array.prototype.group | 16.4 (April 2023) | 10.8 (Oktober 2022) | 106 (September 2022) |
| Temporal | 17.0 (November 2023) | 9.4 (Mai 2021) | 91 (August 2021) |

Alle drei JavaScript-Engines (JavaScriptCore, V8 und SpiderMonkey) unterstützen die genannten Sprachkonstrukte, wenn sie auf den neuesten Versionen laufen. Da diese Engines die Grundlage für die Browser Safari, Chrome und Firefox bilden, können wir davon ausgehen, dass diese Features in modernen Browsern gut unterstützt werden.

Entscheidung:

* Verbesserung der Codequalität
* Modernere und effizientere Techniken
* (Sicherheit: Strengere Gesetze, höherer Wissensstand)

Nachteile:

* Ältere Versionen nicht mehr unterstützt
* (Sicherheit: Neuere Versionen = weniger Nutzung)

Quellen:

* JavaScriptCore Release Notes
* V8 Release Notes
* [SpiderMonkey Release Notes](https://developer.mozilla.org/en-US/docs/Mozilla/Projects/SpiderMonkey/Releases/91)
* [MDN Web Docs](https://developer.mozilla.org/)
* [Can I use](https://caniuse.com/)

### Set.prototype.\*

Mechanismus für das Setzen von Eigenschaften

**Bsp**.: Object.defineProperty, Reflect.set

Die Wahl, wie man Eigenschaften setzt, hängt oft von den Anforderungen an Flexibilität, Sicherheit und Kontrolle ab:

**Direkte Zuweisung**:

* einfach und unkompliziert
* wenig Kontrolle

**Object**.**defineProperty** und **Reflect**.**set**

* mehr Kontrolle
* spezifische Konfigurationsoptionen
  + Enumerierbarkeit
  + Beschreibbarkeit
  + Konfigurierbarkeit

**Proxies**

* ermöglichen, Verhalten beim Setzen von Eigenschaften zu ändern
  + Logginh
  + Validierung

### Array.prototype.flat(depth)

Zur Gruppierung von Elementen eines Arrays

### Temporal

Weiterentwicklung von Date

### Array.prototype.findLast

Diese Methode durchläuft das Array rückwärts und gibt das erste Element zurück

## Aufgabe 6

### Twitter API:

**Zweck**: Die Twitter API ermöglicht Entwicklern den Zugriff auf verschiedene Funktionen von Twitter, wie das Lesen und Schreiben von Tweets, das Abrufen von Benutzerinformationen, das Suchen nach Tweets und Benutzern, das Verwalten von Listen und das Streamen von Echtzeit-Tweets.

**Daten**: Mit der Twitter API können verschiedene Arten von Daten ausgetauscht werden, darunter Tweets, Benutzerprofile, Listen, Trends und Suchergebnisse.

**REST-Prinzipien**: Die Twitter API implementiert die REST-Prinzipien, indem sie HTTP-Methoden wie GET, POST, PUT und DELETE verwendet, um auf Ressourcen zuzugreifen und diese zu manipulieren. Die API verwendet auch einheitliche URIs für Ressourcen und gibt Daten im JSON-Format zurück.

**Richardson Maturity Model**: Die Twitter API liegt auf Stufe 3 des Richardson Maturity Model (RMM). Sie verwendet einheitliche URIs, HTTP-Methoden und Hypermedia-Links, um Ressourcen darzustellen und zu verknüpfen. Die API ist zustandslos und unterstützt die Verwendung von Caching.

**Versionierung**: *pip install requests redis requests\_oauthlib flask*

* imports

### Google Maps API:

**Zweck**: Die Google Maps API ermöglicht Entwicklern den Zugriff auf verschiedene Funktionen von Google Maps, wie das Anzeigen von Karten, das Suchen nach Orten, das Abrufen von Routen, das Anzeigen von Verkehrsinformationen und das Erstellen interaktiver Karten.

**Daten**: Mit der Google Maps API können verschiedene Arten von Geodaten ausgetauscht werden, darunter Kartenbilder, Ortsinformationen, Routeninformationen und Geocodierungsergebnisse.

**REST-Prinzipien**: Die Google Maps API implementiert die REST-Prinzipien, indem sie HTTP-Methoden wie GET verwendet, um Ressourcen abzurufen, und die Daten im JSON-Format zurückgibt. Die API verwendet auch einheitliche URIs für Ressourcen.

**Richardson Maturity Model**: Die Google Maps API liegt ebenfalls auf Stufe 3 des Richardson Maturity Model (RMM). Sie verwendet einheitliche URIs, HTTP-Methoden und Hypermedia-Links, um Ressourcen darzustellen und zu verknüpfen. Die API ist zustandslos und unterstützt die Verwendung von Caching.

**Versionierung**:

Js nicht erforderlich:

<iframe  
  width="600"  
  height="450"  
  style="border:0"  
  loading="lazy"  
  allowfullscreen  
  referrerpolicy="no-referrer-when-downgrade"  
  src="https://www.google.com/maps/embed/v1/place?key=API\_KEY  
    &q=Space+Needle,Seattle+WA">  
</iframe>

Hypermedia   
As   
The   
Engine   
Of   
Application   
State   
Einordnung von Webservices Ein Bild, das Text, Screenshot, Schrift, Reihe enthält.

Automatisch generierte Beschreibung