

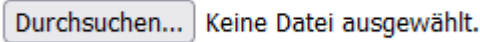
# JavaScript – File API

Ingo Köster

Diplom-Informatiker (FH)

# Die File-API

---

- › Zugriff auf Dateien, die der Anwender angeben muss (nur lesenden)
- › Auswahl der Dateien erfolgt über:
  - › Formularelement (type="file")
  - › Drag-n-Drop
- › Objekttypen der File-API:
  - › **Blob** – Objekt für Binärdaten
  - › **File** - Informationsobjekt für Dateien
  - › **FileList** – Array von File-Objekten
  - › **FileReader** – Objekt zum Auslesen von Daten aus Dateien

# Blob

---

- › Ein Blob-Objekt erlaubt es, reine Binärdaten als Objekt im Browser zu handhaben
- › Ein Blob hat zwei Eigenschaften:
  - › `size` – Länge in Byte
  - › `type` – Mime-Type der Daten
- › Ein Blob hat nur eine Methode:
  - › `slice()` – erlaubt es einen Blob mit einem beliebigen Teilausschnitt des Originals zu erhalten

# File

---

- › Ein File-Objekt erbt das Interface des Blob und erlaubt ganz allgemein Zugriff auf die wichtigsten Meta-Daten und Daten einer Datei
- › Es gibt folgende Eigenschaften:
  - › `size` – Länge in Byte
  - › `type` – Mime-Type der Daten
  - › `lastModifiedDate` – Datum der letzten Änderung
  - › `name` – Name der Datei

# FileReader-Objekt

---

- › FileReader bietet u.a. diese Methoden:
  - › `readAsText()`
    - › Dateiinhalt als Text auslesen
  - › `readAsDataURL()`
    - › Dateiinhalt als Blob auslesen
  - › `readAsArrayBuffer()`
    - › Dateiinhalt als Blob auslesen und in ein Array schreiben

# Events des FileReader-Objektes

---

- › FileReader arbeitet asynchron und hat u.a. folgende Events:
  - › progress, load, abort, error, loadend, loadstart
- › Die gelesenen Daten stehen anschließend im Event-Objekt zur Verfügung
- › Unterschied zwischen load und loadend
  - › load
    - › wenn der Lesevorgang erfolgreich beendet wird
  - › loadend
    - › wenn der Lesevorgang beendet wird (erfolgreich oder fehlerhaft)

# Demo

---

```
let file = ... // Objekt vom Typ File aus input-Element oder  
D'n'D
```

```
let reader = new FileReader();
```

```
reader.readAsText(file);
```

```
reader.onload = function (event)  
{  
    let dateiInhalt = event.target.result;  
}
```