

JSON / XML

JavaScript Object Notation und Extensible Markup Language

- beides sind Vereinbarungen (Notationen) die dem Austausch von Daten zwischen Anwendungen und Computersystemen
 - Webseiten bekommen z.B. die Informationen die an bestimmten Stellen über JSON oder XML, damit nicht immer der Code der Webseite angefasst werden muss, sondern einfach die erhaltene Übertragung ausgelesen werden muss und die "Webseite" dann weiß wo sie was eintragen muss
- Vorteil dabei ist, dass es sich um ein Datenformat handelt, das sowohl vom Computer als auch Menschen lesbar ist
- für beide Formate gibt es Parser in den üblichen Programmiersprachen
 - Parser sind Programme oder Programmteile die eine Datei auslesen und interpretieren können
- XML arbeitet mit Feldern die geöffnet werden `<feldöffnung>` und geschlossen werden `</feldöffnung>`
 - zwischen solchen Feldern können weitere Felder geöffnet werden, dabei entwickelt sich eine Hierarchie der Daten. Felder die zwischen zwei anderen Feldern liegen, sind diesem untergeordnet
- JSON erinnert an die Dictionary-Notation aus Python. Dabei werden Bereiche mit geschweiften Klammern geöffnet und Feldnamen durch "Feldname" gekennzeichnet. Mehrere solcher Felder werden durch Komma getrennt

Exkurs: Notation

- Notation: Art und Weise Informationen darzustellen. Dabei werden bestimmten Zeichen bestimmte Eigenschaften zu geordnet. Eine Notation die wir alle kennen ist die Infix Notation. Also die Art und Weise wie wir Rechnungen darstellen:
 - Infix Notation: $3 + 4$
- Alternativ dazu existieren auch folgende Notationen:
 - Prefix Notation: $+ 3 4$
 - Postfix Notation: $3 4 +$
- Wer sich wundert warum es so was gibt, die Postfix Notation kommt gerne bei Berechnungen die mit Hilfe des Computers gemacht werden zum Einsatz, da sie schneller zu verarbeiten ist und damit die Rechenzeit signifikant verkürzt.

- Beispiel für XML:

```
<?xml version="1.0" standalone="yes" ?>
- <shop location="Birmingham" size="Large">
- <food>
  <Name>Apple</Name>
  <type>fruit</type>
  <cost>15</cost>
</food>
- <food>
  <Name>Carrot</Name>
  <type>vegetable</type>
  <cost>10</cost>
</food>
</shop>
```

- Beispiel für JSON:

```
{
  "@context": "http://schema.org/",
  "@type": "SearchAction",
  "object": {
    "@type": "Event",
    "location": {
      "@type": "Place",
      "geo": {
        "@type": "GeoCoordinates",
        "latitude-input": "optional",
        "longitude-input": "optional"
      }
    },
    "organizer": {
      "@type": "Organization",
      "identifier-input": "optional"
    },
    "isAccessibleForFree-input": "optional"
  },
  "result": {
    "@type": "Event",
    "name-output": "required"
    ...
  },
  "query-input": "optional"
}
```

Revision #4

Created 31 March 2022 17:47:13 by Joshua Lieder

Updated 1 April 2022 09:57:55 by Christoph Wirth