

Aufgabenstellung ü. UTF, XML, JSON

3.Sew

Kodierungsarten (z. B. UTF-8)

Was ist UTF-8?

- Ein Standard zur Umwandlung von Zeichen in digitale Form (Bytes).
- Teil des Unicode-Standards, der nahezu alle Schriftzeichen und Symbole weltweit abdeckt.

Warum verwendet man UTF-8?

- **Kompatibilität:** Abwärtskompatibel mit ASCII (z. B. englische Zeichen).
 - **Speicherplatz:** Häufige Zeichen (z. B. lateinische Buchstaben) benötigen nur 1 Byte.
 - **Universell:** Unterstützt viele Sprachen, Emojis und Sonderzeichen.
 - **Webstandard:** Wird von fast allen Webseiten und APIs verwendet.
-

Was ist XML?

XML (*Extensible Markup Language*) ist ein menschen- und maschinenlesbares Textformat, um strukturierte Daten zu speichern und zu übertragen.

Wie funktioniert XML?

- **Datenstruktur:** Hierarchische Organisation in einer Baumstruktur.
- **Tags:** Daten werden mit benutzerdefinierten Tags gekapselt, z. B. `<name>John</name>`.
- **Parser:** Software kann XML-Daten lesen und in maschinenverständliche Form umwandeln.

Beispiel für XML:

```
<person>
  <name>John Doe</name>
```

```
<age>30</age>
<email>johndoe@example.com</email>
</person>
```

Was ist JSON?

Definition:

JSON (*JavaScript Object Notation*) ist ein leichtgewichtiges Datenformat für den Austausch von Informationen, besonders in Web-APIs.

Wie funktioniert JSON?

- **Datenstruktur:** Organisiert als Schlüssel-Wert-Paare.
- **Flexibilität:** Werte können Arrays oder verschachtelte Objekte sein.
- **Parser:** JSON kann direkt in Objekte vieler Programmiersprachen umgewandelt werden.

Beispiel für JSON:

```
{
  "person": {
    "name": "John Doe",
    "age": 30,
    "email": "johndoe@example.com"
  }
}
```

XML vs. JSON (Kurzvergleich)

- **XML:** Hierarchisch, benötigt Tags, ideal für Dokumentzentrierte Daten.
 - **JSON:** Kompakter, einfacher für APIs und moderne Anwendungen.
-