

9.1 OSI-Schicht 6 - Darstellungsschicht

Die Darstellungsschicht ist Schicht 6 des OSI-Modells.

Sie liegt zwischen:

Schicht 5: Sitzungsschicht
und
Schicht 7: Anwendungsschicht

Die Hauptaufgabe von Schicht 6 ist:

Daten so darzustellen, umzuwandeln oder zu schützen,
dass Anwendungen sie korrekt verstehen können.

Dabei geht es vor allem um:

- Datenformate
- Zeichencodierung
- Kompression
- Verschlüsselung
- Entschlüsselung
- Datenumwandlung
- Darstellung von Informationen

Merksatz:

Schicht 6 = Darstellung, Codierung, Kompression und Verschlüsselung.

Grundidee von Schicht 6

Schicht 6 sorgt dafür, dass Daten in einer passenden Form vorliegen.

Ein Sender kann Daten in einem bestimmten Format erzeugen.

Der Empfänger muss diese Daten richtig interpretieren können.

Beispiele:

Text muss mit der richtigen Zeichencodierung gelesen werden.
Bilder müssen im richtigen Dateiformat verstanden werden.
Verschlüsselte Daten müssen entschlüsselt werden.
Komprimierte Daten müssen entpackt werden.

Merksatz:

Schicht 6 kümmert sich darum,
wie Daten dargestellt und interpretiert werden.

Warum braucht man die Darstellungsschicht?

Unterschiedliche Systeme können Daten unterschiedlich speichern oder darstellen.

Beispiele:

- Textcodierung
- Zahlenformat
- Dateiformat
- Bildformat
- Audioformat
- Verschlüsselung
- Kompression

Damit Sender und Empfänger dieselben Daten gleich verstehen, braucht man Regeln für die Darstellung.

Merksatz:

Gleiche Daten müssen auf beiden Seiten gleich verstanden werden.

Einordnung im OSI-Modell

OSI-Schicht	Name	Aufgabe
7	Anwendungsschicht	Netzwerkdienste für Anwendungen
6	Darstellungsschicht	Darstellung, Codierung, Verschlüsselung

OSI-Schicht	Name	Aufgabe
5	Sitzungsschicht	Sitzungen aufbauen, verwalten und beenden
4	Transportschicht	TCP, UDP, Ports
3	Vermittlungsschicht / Netzwerkschicht	IP, Routing
2	Sicherungsschicht	Frames, MAC, VLAN
1	Bitübertragungsschicht	Bits, Signale

Merksatz:

Schicht 6 liegt zwischen Sitzung und Anwendung.

Schicht 5, 6 und 7 unterscheiden

Schicht	Grundidee	Beispiel
5	Sitzung verwalten	Benutzer bleibt angemeldet
6	Daten darstellen oder umwandeln	UTF-8, TLS, Kompression
7	Anwendung und Dienst	HTTP, DNS, SMTP

Merksatz:

Schicht 5 = Sitzung.

Schicht 6 = Darstellung.

Schicht 7 = Anwendung.

Darstellung bedeutet nicht Bildschirmdarstellung

Der Begriff Darstellungsschicht wird manchmal falsch verstanden.

Es geht nicht nur darum:

was auf dem Bildschirm angezeigt wird.

Es geht fachlich darum:

wie Daten codiert,
formatiert,

komprimiert,
verschlüsselt
oder interpretiert werden.

Beispiel:

Ein Text kann gleich aussehen,
aber intern unterschiedlich codiert sein.

Merksatz:

Darstellungsschicht meint Datenrepräsentation,
nicht nur grafische Anzeige.

Datenformate

Ein Datenformat legt fest, wie Informationen aufgebaut sind.

Beispiele:

Datenart	Beispiele
Text	TXT, HTML, XML, JSON
Bild	JPEG, PNG, GIF, SVG
Audio	WAV, MP3, AAC
Video	MP4, H.264, H.265
Dokument	PDF, DOCX
Archiv	ZIP, 7z

Wenn ein Empfänger das Format nicht versteht, kann er die Daten nicht korrekt verarbeiten.

Merksatz:

Datenformat = Regel,
wie Daten aufgebaut sind.

Beispiel: JSON

JSON steht für:

JavaScript Object Notation

JSON wird häufig verwendet, um strukturierte Daten zwischen Anwendungen auszutauschen.

Beispiel:

```
{  
  "name": "Felix",  
  "rolle": "FISI",  
  "aktiv": true  
}
```

JSON ist besonders häufig bei:

- APIs
- Webanwendungen
- Konfigurationsdaten
- Datenaustausch zwischen Diensten

Merksatz:

JSON ist ein häufiges Datenformat für strukturierte Daten.

Beispiel: XML

XML steht für:

Extensible Markup Language

XML ist ein textbasiertes Format zur strukturierten Darstellung von Daten.

Beispiel:

```
<benutzer>  
  <name>Felix</name>  
  <rolle>FISI</rolle>  
</benutzer>
```

XML wird unter anderem genutzt bei:

- Schnittstellen
- Konfigurationen
- Dokumentenformaten
- älteren Webservices

Merksatz:

XML ist ein strukturiertes Textformat.

JSON und XML vergleichen

Merkmal	JSON	XML
Schreibweise	kompakter	ausführlicher
Struktur	Objekte und Arrays	Tags und Elemente
häufig bei	Web-APIs	ältere Schnittstellen, Dokumente
Lesbarkeit	meist einfach	oft länger
Datenaustausch	sehr verbreitet	weiterhin verbreitet

Merksatz:

JSON und XML sind Formate für strukturierten Datenaustausch.

Zeichencodierung

Zeichencodierung legt fest, wie Zeichen als Zahlen oder Bits gespeichert werden.

Beispiele für Zeichen:

A
ä
ß
€
□
□□

Damit ein Computer diese Zeichen speichern kann, braucht er eine Codierung.

Merksatz:

Zeichencodierung übersetzt Zeichen in digitale Werte.

ASCII

ASCII ist eine ältere Zeichencodierung.

ASCII umfasst grundlegende Zeichen wie:

- englische Buchstaben
- Ziffern
- Satzzeichen
- Steuerzeichen

ASCII ist auf einfache englische Zeichen beschränkt.

Deutsche Umlaute wie:

ä
ö
ü
ß

sind in klassischem ASCII nicht enthalten.

Merksatz:

ASCII ist einfach,
aber für viele Sprachen zu begrenzt.

Unicode

Unicode ist ein Standard, der Zeichen vieler Schriftsysteme abbilden kann.

Dazu gehören zum Beispiel:

- lateinische Zeichen
- Umlaute
- griechische Zeichen
- kyrillische Zeichen
- asiatische Schriftzeichen
- Symbole
- Emojis

Unicode beschreibt, welche Zeichen es gibt und welche Nummer sie haben.

Merksatz:

Unicode ist ein großer Zeichensatz für viele Sprachen und Symbole.

UTF-8

UTF-8 ist eine sehr verbreitete Codierung für Unicode-Zeichen.

UTF-8 ist wichtig für:

- Webseiten
- Linux-Systeme
- Konfigurationsdateien
- APIs
- Datenbanken
- E-Mails

Vorteil:

UTF-8 kann sehr viele Zeichen darstellen
und ist mit einfachem ASCII weitgehend kompatibel.

Merksatz:

UTF-8 ist die wichtigste moderne Zeichencodierung.

Typisches Problem: falsche Zeichencodierung

Wenn Text mit falscher Codierung interpretiert wird, entstehen fehlerhafte Zeichen.

Beispiel:

ä
wird falsch angezeigt als:
Ã¤

Mögliche Ursachen:

- falsche Zeichencodierung
- falscher HTTP-Header
- falsche Datenbankcodierung
- falsche Dateicodierung
- falsche Import- oder Exporteinstellung

Merksatz:

Falsche Codierung führt zu Zeichenfehlern.

Beispiel: Umlaute falsch dargestellt

Fehlerbild:

Müller
wird angezeigt als:
MÃ¼ller

Das deutet häufig auf ein Codierungsproblem hin.

Mögliche Ursache:

UTF-8-Daten werden als eine andere Codierung interpretiert.

Prüfung:

Dateicodierung prüfen.
Datenbankcodierung prüfen.

HTTP-Header prüfen.

Anwendungseinstellung prüfen.

Merksatz:

Kaputte Umlaute sind oft ein Schicht-6-Thema.

Kompression

Kompression bedeutet:

Daten werden verkleinert.

Ziel:

weniger Speicherplatz
weniger Übertragungsvolumen
schnellere Übertragung

Beispiele:

- ZIP
- gzip
- Brotli
- JPEG
- MP3
- H.264

Merksatz:

Kompression reduziert Datenmenge.

Verlustfreie Kompression

Verlustfreie Kompression bedeutet:

Die Originaldaten können vollständig wiederhergestellt werden.

Beispiele:

- ZIP
- gzip
- PNG
- FLAC

Typisch für:

- Textdateien
- Programme
- Dokumente
- Datenbanken
- Quellcode

Merksatz:

Verlustfrei = nach dem Entpacken exakt wiederherstellbar.

Verlustbehaftete Kompression

Verlustbehaftete Kompression bedeutet:

Daten werden verkleinert,
aber ein Teil der ursprünglichen Informationen geht verloren.

Beispiele:

- JPEG
- MP3
- AAC
- H.264
- H.265

Typisch für:

- Fotos
- Musik
- Videos

Merksatz:

Verlustbehaftet = kleiner,
aber nicht mehr exakt original.

Kompression im Web

Webserver können Inhalte komprimiert senden.

Beispiele:

gzip
Brotli

Das reduziert die Datenmenge bei:

- HTML
- CSS
- JavaScript
- JSON
- XML

Der Browser entpackt die Daten wieder.

Merksatz:

Web-Kompression spart Bandbreite.

Verschlüsselung

Verschlüsselung bedeutet:

Daten werden so umgewandelt,
dass sie ohne passenden Schlüssel nicht lesbar sind.

Ziele:

- Vertraulichkeit
- Schutz vor Mitlesen
- Schutz sensibler Informationen

Beispiele:

- HTTPS
- VPN
- E-Mail-Verschlüsselung
- Festplattenverschlüsselung

Merksatz:

Verschlüsselung schützt die Vertraulichkeit von Daten.

Entschlüsselung

Entschlüsselung bedeutet:

Verschlüsselte Daten werden mit dem passenden Schlüssel wieder lesbar gemacht.

Ohne passenden Schlüssel bleiben die Daten unverständlich.

Merksatz:

Entschlüsselung macht verschlüsselte Daten wieder lesbar.

TLS

TLS steht für:

Transport Layer Security

TLS wird unter anderem bei HTTPS verwendet.

HTTPS bedeutet:

HTTP über TLS

TLS schützt unter anderem:

- Vertraulichkeit
- Integrität
- Authentizität des Servers

Typisches Beispiel:

`https://example.com`

Merksatz:

HTTPS = HTTP mit TLS-Schutz.

TLS und OSI-Schicht 6

TLS wird im OSI-Modell oft der Darstellungsschicht zugeordnet, weil es Daten verschlüsselt und entschlüsselt.

In der Praxis liegt TLS zwischen Anwendung und Transport.

Beispiel:

```
HTTP
→ TLS
→ TCP
→ IP
```

Wichtig für die Prüfung:

TLS hat starken Bezug zu Schicht 6,
weil Verschlüsselung und Darstellung betroffen sind.

Merksatz:

TLS schützt Daten zwischen Anwendung und Transport.

TLS-Zertifikat

Ein TLS-Zertifikat hilft dem Client zu prüfen, ob er mit dem richtigen Server spricht.

Ein Zertifikat enthält unter anderem:

- Domainname
- öffentlicher Schlüssel
- Aussteller
- Gültigkeitszeitraum
- Signatur der Zertifizierungsstelle

Merksatz:

Zertifikat hilft bei der Vertrauensprüfung des Servers.

Typische Zertifikatsfehler

Typische Fehler sind:

- Zertifikat abgelaufen
- Zertifikat gilt für falschen Namen
- Zertifikat nicht vertrauenswürdig
- Zwischenzertifikat fehlt
- Uhrzeit am Client falsch
- selbstsigniertes Zertifikat ohne Vertrauen
- falsches Zertifikat auf dem Server

Merksatz:

Zertifikatsfehler sind oft Darstellungs- oder Anwendungsschicht-Themen.

Beispiel: Name passt nicht zum Zertifikat

Fehlerbild:

Browser warnt vor Zertifikat.

Mögliche Ursache:

Zertifikat wurde ausgestellt für:
server.local

Aufgerufen wurde aber:

wiki.firma.de

Dann passt der Name nicht.

Merksatz:

Der aufgerufene Name muss zum Zertifikat passen.

Beispiel: Zertifikat abgelaufen

Ein Zertifikat hat einen Gültigkeitszeitraum.

Wenn dieser überschritten ist, warnen Clients.

Mögliche Ursache:

Zertifikat wurde nicht rechtzeitig erneuert.

Folge:

Browser zeigt Warnung.
API-Verbindungen können scheitern.

Anwendungen verweigern Verbindung.

Merksatz:

Zertifikate müssen gültig und aktuell sein.

Datenintegrität

Integrität bedeutet:

Daten wurden nicht unbemerkt verändert.

TLS schützt nicht nur vor Mitlesen, sondern auch vor unbemerkter Manipulation während der Übertragung.

Merksatz:

Integrität = Daten bleiben unverändert.

Authentizität

Authentizität bedeutet:

Die Gegenstelle ist die,
für die sie sich ausgibt.

Bei HTTPS prüft der Browser:

Passt das Zertifikat zur Domain?
Ist das Zertifikat vertrauenswürdig?
Ist es gültig?

Merksatz:

Authentizität = Echtheit der Gegenstelle.

Codierung und Verschlüsselung unterscheiden

Codierung und Verschlüsselung werden oft verwechselt.

Begriff	Ziel	Schutz?
Codierung	Daten in anderes Format bringen	nein
Kompression	Datenmenge reduzieren	nein
Verschlüsselung	Daten vor unbefugtem Lesen schützen	ja

Beispiel:

Base64 ist Codierung,
aber keine Verschlüsselung.

Merksatz:

Codierung ist nicht automatisch Sicherheit.

Base64

Base64 ist eine Codierung.

Sie wandelt Binärdaten in Textzeichen um.

Typische Nutzung:

- E-Mail-Anhänge
- Tokens
- Zertifikatsdaten
- API-Daten
- Binärdaten in Textformaten

Wichtig:

Base64 ist leicht rückgängig zu machen.
Es ist keine Verschlüsselung.

Merksatz:

Base64 ist Codierung,
keine Verschlüsselung.

Hashing kurz eingeordnet

Hashing ist keine Verschlüsselung.

Ein Hash bildet Daten auf einen festen Prüfwert ab.

Beispiele:

- SHA-256
- SHA-512

Eigenschaften:

- Einwegfunktion
- gleiche Eingabe ergibt gleichen Hash
- kleine Änderung ergibt anderen Hash
- Originaldaten sollen nicht aus dem Hash zurückgerechnet werden

Merksatz:

Hashing prüft Integrität,
verschlüsselt aber nicht.

Hash und Passwortspeicherung

Passwörter sollten nicht im Klartext gespeichert werden.

Stattdessen werden Passwort-Hashes gespeichert.

Wichtig:

Für Passwörter nutzt man spezielle Verfahren,
zum Beispiel mit Salt und langsamen Hashfunktionen.

Beispiele:

bcrypt
Argon2
PBKDF2

Merksatz:

Passwörter nicht verschlüsseln,
sondern sicher hashen.

Datenumwandlung

Schicht 6 kann Daten zwischen Formaten umwandeln.

Beispiele:

- Textcodierung ändern
- Bildformat konvertieren
- Dateiformat umwandeln
- Daten serialisieren
- Daten deserialisieren
- Zahlenformat anpassen

Merksatz:

Datenumwandlung sorgt dafür,
dass Empfänger Daten verstehen.

Serialisierung

Serialisierung bedeutet:

Datenstrukturen werden in ein übertragbares oder speicherbares Format umgewandelt.

Beispiel:

Objekt in Anwendung
→ JSON

→ Übertragung über Netzwerk

Empfänger:

JSON

→ Objekt in Anwendung

Merksatz:

Serialisierung macht Daten übertragbar.

Deserialisierung

Deserialisierung ist die Rückumwandlung.

Beispiel:

JSON-Text wird in ein Objekt der Anwendung umgewandelt.

Wichtig:

Unsichere Deserialisierung kann Sicherheitsrisiken erzeugen.

Merksatz:

Deserialisierung macht übertragene Daten wieder nutzbar.

Endianness kurz erklärt

Endianness beschreibt die Reihenfolge von Bytes bei mehrbyteigen Zahlen.

Begriffe:

Big Endian

Little Endian

In Netzwerken ist oft die sogenannte Netzwerk-Byte-Reihenfolge wichtig.

Für AP1/AP2 reicht meistens:

Unterschiedliche Systeme können Zahlen intern unterschiedlich darstellen.

Merksatz:

Darstellung betrifft auch die interne Reihenfolge von Daten.

Schicht 6 und Webanwendungen

Bei Webanwendungen sieht man Schicht-6-Themen häufig bei:

- TLS
- Zertifikaten
- Zeichencodierung
- JSON
- XML
- gzip-Kompression
- Datenformaten
- Dateiformaten
- MIME-Typen

Merksatz:

Webprobleme können Darstellungsprobleme sein.

MIME-Type

Ein MIME-Type beschreibt den Medientyp von Daten.

Beispiele:

MIME-Type	Bedeutung
text/html	HTML-Dokument
application/json	JSON-Daten
image/png	PNG-Bild
image/jpeg	JPEG-Bild

MIME-Type	Bedeutung
text/css	CSS-Datei
application/pdf	PDF-Dokument

Der MIME-Type hilft dem Client, Daten richtig zu interpretieren.

Merksatz:

MIME-Type sagt,
was für Daten übertragen werden.

Fehlerbild: Datei wird falsch angezeigt

Mögliche Ursachen:

- falscher MIME-Type
- falsche Dateiendung
- falsches Dateiformat
- beschädigte Datei
- fehlende Anwendung zum Öffnen
- falsche Codierung

Beispiel:

JSON wird als Text angezeigt,
statt als application/json verarbeitet zu werden.

Merksatz:

Falsche Darstellung kann am Format oder MIME-Type liegen.

Fehlerbild: Webseite zeigt komische Zeichen

Mögliche Ursachen:

- falsche Zeichencodierung
- fehlende UTF-8-Angabe

- Datenbank speichert andere Codierung
- Import falsch durchgeführt
- Datei in falscher Codierung gespeichert

Merksatz:

Komische Zeichen sind oft Codierungsprobleme.

Fehlerbild: HTTPS-Warnung

Mögliche Ursachen:

- Zertifikat abgelaufen
- Domainname passt nicht
- Zertifizierungsstelle nicht vertraut
- Zertifikatskette unvollständig
- Uhrzeit am Client falsch
- selbstsigniertes Zertifikat

Merksatz:

HTTPS-Warnungen oft durch Zertifikatsprobleme.

Fehlerbild: API versteht Daten nicht

Eine API kann Daten ablehnen, wenn Format oder Codierung nicht stimmen.

Mögliche Ursachen:

- ungültiges JSON
- falscher Content-Type
- falsche Zeichencodierung
- fehlende Pflichtfelder
- falsche Datentypen
- falsches Datumsformat

Merksatz:

API-Fehler können Darstellungs- oder Formatprobleme sein.

Fehlerbild: Komprimierte Daten können nicht gelesen werden

Mögliche Ursachen:

- falsches Kompressionsverfahren
- beschädigte Datei
- falsche Dateiendung
- Empfänger unterstützt Verfahren nicht
- Header gibt falsche Kompression an

Merksatz:

Kompression muss vom Empfänger unterstützt werden.

Darstellungsschicht und Sicherheit

Viele Sicherheitsfunktionen haben Bezug zur Darstellungsschicht.

Beispiele:

- TLS-Verschlüsselung
- Zertifikate
- sichere Codierung
- sichere Deserialisierung
- Integritätsprüfung
- Schutz vor Datenmanipulation

Wichtig:

Falsche Darstellung oder unsichere Verarbeitung kann Sicherheitslücken erzeugen.

Merksatz:

Schicht 6 ist auch für Sicherheit wichtig.

Typische Angriffsbezüge

Darstellung und Datenverarbeitung können Angriffsflächen sein.

Beispiele:

- unsichere Deserialisierung
- fehlerhafte Parser
- manipulierte Dateiformate
- falsche Zertifikatsprüfung
- schwache Verschlüsselung
- XSS durch falsche Ausgabe-Codierung

Merksatz:

Daten müssen sicher interpretiert und verarbeitet werden.

Was Schicht 6 nicht macht

Schicht 6 macht nicht:

- Bits über Kabel übertragen
- MAC-Adressen verwenden
- IP-Pakete routen
- TCP-Verbindungen aufbauen
- Ports bereitstellen
- Sitzungen verwalten
- konkrete Netzwerkdienste vollständig bereitstellen

Diese Aufgaben liegen auf anderen Schichten.

Merksatz:

Schicht 6 stellt Daten dar,
transportiert sie aber nicht selbst.

Einordnung in das OSI-Modell

Thema	Schicht
-------	---------

Kabel, Signal	1
MAC-Adresse, Frame	2
IP-Adresse, Routing	3
TCP, UDP, Port	4
Sitzung, Session-ID	5
UTF-8, JSON, TLS, Kompression	6
HTTP, DNS, SMTP, Anwendung	7

Merksatz:

Codierung, Kompression und Verschlüsselung = Schicht 6.

Schicht 6 und Schicht 7 unterscheiden

Thema	zugehörige Schicht
HTTP-Anfrage	7
DNS-Abfrage	7
JSON-Datenformat	6 / 7-Bezug
UTF-8-Codierung	6
TLS-Verschlüsselung	6
Zertifikat	6 / 7-Bezug
gzip-Kompression	6
Anwendung verarbeitet Formular	7

Merksatz:

Schicht 6 beschreibt die Form der Daten.

Schicht 7 beschreibt den Dienst.

Typische IHK-Fragen

In AP1 und AP2 kann zum Beispiel gefragt werden:

- Welche Aufgabe hat die Darstellungsschicht?
- Was bedeutet Zeichencodierung?

- Was ist UTF-8?
- Was ist der Unterschied zwischen Codierung und Verschlüsselung?
- Was ist Kompression?
- Was ist der Unterschied zwischen verlustfrei und verlustbehaftet?
- Was macht TLS?
- Warum gibt es Zertifikatswarnungen?
- Was ist ein MIME-Type?
- Warum kann eine Webseite falsche Zeichen anzeigen?
- Warum ist Base64 keine Verschlüsselung?
- Was ist Serialisierung?
- Warum können Formatfehler bei APIs auftreten?

Typische Prüfungsfallen

Schicht 6 heißt Darstellungsschicht.

Darstellung bedeutet Datenrepräsentation,
nicht nur Bildschirmdarstellung.

UTF-8 ist eine Zeichencodierung.

ASCII ist begrenzt.

Unicode ist ein Zeichensatzstandard.

JSON und XML sind Datenformate.

Kompression ist keine Verschlüsselung.

Codierung ist keine Verschlüsselung.

Base64 ist keine Verschlüsselung.

Hashing ist keine Verschlüsselung.

TLS verschlüsselt und schützt Daten.

HTTPS nutzt TLS.

Zertifikat muss gültig und passend sein.

MIME-Type hilft beim richtigen Interpretieren von Daten.

Falsche Codierung führt zu Zeichenfehlern.

Schicht 6 ist in der Praxis oft mit Schicht 7 verbunden.

Wichtige Begriffe kurz erklärt

Begriff	Kurze Erklärung
Darstellungsschicht	OSI-Schicht 6
Datenformat	Struktur und Aufbau von Daten
Zeichencodierung	Zuordnung von Zeichen zu digitalen Werten
ASCII	ältere einfache Zeichencodierung
Unicode	Standard für viele Zeichen weltweit
UTF-8	verbreitete Unicode-Codierung
Kompression	Verkleinerung von Daten
verlustfrei	Original vollständig wiederherstellbar
verlustbehaftet	Datenverlust zugunsten kleinerer Größe
Verschlüsselung	Daten für Unbefugte unlesbar machen
Entschlüsselung	verschlüsselte Daten wieder lesbar machen
TLS	Verschlüsselung und Schutz zwischen Anwendung und Transport
Zertifikat	Vertrauensnachweis für Serveridentität
MIME-Type	Angabe des Medientyps von Daten
Base64	Codierung von Binärdaten als Text
Hashing	Einweg-Prüfwertbildung
Serialisierung	Umwandlung in übertragbares Format
Deserialisierung	Rückumwandlung aus übertragbarem Format

IHK-sichere Kurzformulierung

Die Darstellungsschicht ist Schicht 6 des OSI-Modells. Sie sorgt dafür, dass Daten in einer Form vorliegen, die vom Empfänger korrekt interpretiert werden kann. Dazu gehören Datenformate, Zeichencodierung, Kompression, Verschlüsselung und Datenumwandlung. Beispiele sind UTF-8 für

Zeichencodierung, JSON oder XML als Datenformate, gzip als Kompression und TLS für verschlüsselte Kommunikation. Codierung und Kompression sind nicht dasselbe wie Verschlüsselung. In modernen Netzwerken sind Aufgaben der Darstellungsschicht oft eng mit der Anwendungsschicht verbunden.

Merksätze

Schicht 6 = Darstellungsschicht.

Schicht 6 = Daten darstellen und umwandeln.

Darstellung bedeutet Datenrepräsentation.

Datenformat legt Aufbau von Daten fest.

Zeichencodierung übersetzt Zeichen in digitale Werte.

ASCII ist begrenzt.

Unicode umfasst viele Zeichen.

UTF-8 ist sehr verbreitet.

Falsche Codierung erzeugt Zeichenfehler.

Kompression reduziert Datenmenge.

Verlustfrei = exakt wiederherstellbar.

Verlustbehaftet = kleiner,
aber nicht exakt original.

Verschlüsselung schützt Vertraulichkeit.

Entschlüsselung macht Daten wieder lesbar.

TLS schützt Daten bei HTTPS.

Zertifikat prüft Serververtrauen.

Codierung ist keine Verschlüsselung.

Base64 ist keine Verschlüsselung.

Hashing ist keine Verschlüsselung.

MIME-Type beschreibt den Datentyp.

JSON und XML sind Datenformate.

Serialisierung macht Daten übertragbar.

Schicht 6 ist oft eng mit Schicht 7 verbunden.
