

# COBOL Cheat Sheet

## Cobol Grundsyntax

### COBOL Datentypen

Typ	Format	Beispiel	Beschreibung
Alphanumerisch	PIC X(n)	01 NAME PIC X(30).	Beliebige Zeichen
Alphabetisch	PIC A(n)	01 LAND PIC A(20).	Nur Buchstaben
Numerisch	PIC 9(n)	01 ALTER PIC 9(3).	Nur Ziffern
Dezimal	PIC 9(n)V9(m)	01 PREIS PIC 9(5)V99.	Virtueller Dezimalpunkt
Signiert	PIC S9(n)	01 SALDO PIC S9(7)V99.	Mit Vorzeichen +/-

### Variablen

```
1      01 CUSTOMER-DATA .
2          05 CUSTOMER-NAME PIC X(30) VALUE SPACES .
3          05 CUSTOMER-AGE PIC 9(3) VALUE 0 .
```

01 ist die Hauptebene. 05, 10, 15 die Unterebenen.

```
1      MOVE "Emma" TO CUSTOMER-NAME. oder MOVE 25 TO CUSTOMER-AGE .
```

Zuweisung von Werten zu Variablen.

### Ablaufsteuerung

Statement	Bedeutung	Beispiel
DISPLAY	Ausgabe	DISPLAY "Eingabe (JA/NEIN):".
ACCEPT	Eingabe	ACCEPT ANTWORT.
MOVE	Zuweisung	MOVE "AKTIV" TO STATUS.
IF	Bedingung	IF ANTWORT = "JA" ... ELSE ... END-IF.
PERFORM	Paragraph aufrufen	PERFORM CALC-ABSCHNITT.
STOP RUN	Programm beenden	STOP RUN.

### Methoden

```
1      PROCEDURE DIVISION .
2          BEGIN .
3              DISPLAY "Programm gestartet.".
4              PERFORM CALC-SUM .
5              STOP RUN .
6
7          CALC-SUM .
8              MOVE 100 TO SUMME .
9              DISPLAY "Summe: " SUMME .
```

## Cobol Funktionen

Funktion	Beschreibung	Beispiel
CURRENT-DATE	Aktuelles Datum/Uhrzeit	DISPLAY FUNCTION CURRENT-DATE.
LENGTH(var)	Länge eines Feldes	MOVE FUNCTION LENGTH(NAME) TO LEN.
NUMVAL(str)	String zu Zahl	MOVE FUNCTION NUMVAL("123") TO NUM.
UPPER-CASE(str)	Großbuchstaben	MOVE FUNCTION UPPER-CASE(TEXT) TO UPPER.
LOWER-CASE(str)	Kleinbuchstaben	MOVE FUNCTION LOWER-CASE(TEXT) TO LOWER.

## Vollständiges Beispiel

```

1  IDENTIFICATION DIVISION.
2  PROGRAM-ID. WORKSHOP-DEMO.
3  ENVIRONMENT DIVISION.
4  CONFIGURATION SECTION.
5  SOURCE-COMPUTER. TECHNISCHE-HOCHSCHULE-MANNHEIM.
6  OBJECT-COMPUTER. TECHNISCHE-HOCHSCHULE-MANNHEIM.
7  DATA DIVISION.
8  WORKING-STORAGE SECTION.
9      01 FRAGE PIC X(30) VALUE "Ist der Kurs vorbei? (JA/NEIN)".
10     01 ANTWORT PIC X(10).
11     01 KURSSTATUS PIC X(15) VALUE "AKTIV".
12  PROCEDURE DIVISION.
13  BEGIN.
14      DISPLAY FRAGE.
15      ACCEPT ANTWORT.
16      IF ANTWORT = "JA"
17          MOVE "FEIERABEND" TO KURSSTATUS
18      DISPLAY "Status: " KURSSTATUS "."
19      ELSE
20          DISPLAY "Der Kurs geht weiter."
21      END-IF.
22      STOP RUN

```

# COBOL Cheat Sheet

## Hilfestellungen für Aufgabe 1

### Datei definieren

```
1  FILE-CONTROL .
2      SELECT VARIABLENNAME
3          ASSIGN TO "DATEINAME"
4          ORGANIZATION IS EINLESEART.
5
6  FILE SECTION.
7      FD VARIABLENNAME.
8          01 BUCHUNG-LINE PIC X(80).
```

### Datei öffnen und schließen

```
1  OPEN INPUT VARIABLENNAME.
2  CLOSE BUCHUNGEN.
```

### Eine Zeile einlesen

```
1  READ BUCHUNGEN
2  END-READ
```

### Berechnungen

```
1  COMPUTE VARIABLENNAME = X + Y * Z.
```

### Mathematische Funktionen

```
1  ADD X TO Y GIVING SUMXY.
2  SUBTRACT Y FROM X GIVING SUBTRACTION.
3  MULTIPLY X BY Y GIVING PRODUCT.
4  DIVIDE X BY Y GIVING QUOTIENT.
```

# COBOL Cheat Sheet

Hilfestellungen für Aufgabe 2 (a und b)

## Leerzeilen behandeln und Tabs entfernen

```
1  MOVE FUNCTION TRIM(BUCHUNG-LINE) TO LINE-TRIM.
2  INSPECT LINE-TRIM REPLACING ALL X"09" BY " ".    *> Tabs -> Spaces
3
4  IF LINE-TRIM = SPACES
5      OR LENGTH OF FUNCTION TRIM(LINE-TRIM) = 0
6      CONTINUE    *> Leere Zeile überspringen
7  END-IF.
```

## Strings per UNSTRING zerlegen

```
1  MOVE SPACES TO WS-VORNAME WS-NACHNAME
2                      WS-STUNDEN WS-GEHALT WS-GEBURT.
3
4  UNSTRING LINE-TRIM
5      DELIMITED BY ALL SPACE
6      INTO WS-VORNAME
7              WS-NACHNAME
8              WS-STUNDEN
9              WS-GEHALT
10             WS-GEBURT
11  END-UNSTRING.
```

## Strings zusammenbauen (STRING)

```
1  MOVE FUNCTION TRIM(WS-VORNAME) TO WS-NAME-ZUS.
2  MOVE 1 TO WS-J.
3
4  STRING
5      " " DELIMITED BY SIZE
6      FUNCTION TRIM(WS-NACHNAME) DELIMITED BY SIZE
7      INTO WS-NAME-ZUS
8      WITH POINTER WS-J
9  END-STRING.
```

## Längenprüfung eines Strings

```
1  MOVE LENGTH OF FUNCTION TRIM(WS-GEBURT) TO WS-LEN.
2
3  IF WS-LEN NOT = 10
4      *> Fehlerfall behandeln
5  END-IF.
```

## Elektronische Zeichenprüfung (INSPECT TALLYING)

```
1  MOVE 0 TO WS-CNT.
2
3  INSPECT ALLOWED-NAME-CHARS
4      TALLYING WS-CNT
5      FOR ALL WS-NAME-ZUS(WS-I:1).
6
7  IF WS-CNT = 0
8      *> Sonderzeichen gefunden
9  END-IF.
```

## Numerische Prüfung

```
1  IF WS-STUNDEN(1:1) = "-"
2      *> Negative Werte nicht zulässig
3  END-IF.
4
5  IF WS-WERT NUMERIC
6      *> gültige Zahl
7  ELSE
8      *> Fehler: nicht numerisch
9  END-IF.
```

## Mehrfach-Trenner prüfen

```
1  MOVE 0 TO WS-CNT.
2  INSPECT WS-STUNDEN TALLYING WS-CNT FOR ALL ".".
3
4  IF WS-CNT > 1
5      *> zu viele Dezimalpunkte
6  END-IF.
```

## Datum zerlegen und prüfen

```
1  MOVE WS-GEBURT(1:2) TO WS-TAG.
2  MOVE WS-GEBURT(4:2) TO WS-MONAT.
3  MOVE WS-GEBURT(7:4) TO WS-JAHR.
4
5  IF WS-MONAT < 1 OR WS-MONAT > 12
6      *> Ungültiger Monat
7  END-IF.
8
9  *> Max. Tageszahl je Monat (inkl. Schaltjahrregel)
10 EVALUATE WS-MONAT
11     WHEN 2
12         IF (FUNCTION MOD(WS-JAHR 400) = 0)
13             OR (FUNCTION MOD(WS-JAHR 4) = 0
14                 AND FUNCTION MOD(WS-JAHR 100) NOT = 0)
15             MOVE 29 TO WS-TAG-MAX
16         ELSE
17             MOVE 28 TO WS-TAG-MAX
18         END-IF
19     WHEN 4 MOVE 30 TO WS-TAG-MAX
20     WHEN 6 MOVE 30 TO WS-TAG-MAX
21     WHEN 9 MOVE 30 TO WS-TAG-MAX
22     WHEN 11 MOVE 30 TO WS-TAG-MAX
23     WHEN OTHER MOVE 31 TO WS-TAG-MAX
24 END-EVALUATE.
25
26 IF WS-TAG < 1 OR WS-TAG > WS-TAG-MAX
27     *> Unmögliches Datum
28 END-IF.
```

## Fehlerlog schreiben (STRING WRITE)

```
1  MOVE SPACES TO FEHLER-RECORD.
2
3  STRING
4      FUNCTION TRIM(LINE-TRIM)
5      " ; Fehler: " FUNCTION TRIM(ARG-FELD)
6      " - "          FUNCTION TRIM(ARG-MSG)
7      INTO FEHLER-RECORD
8  END-STRING.
9
10 WRITE FEHLER-RECORD.
```

## Fehlerzähler erhöhen

```
1  ADD 1 TO CNT-ERROR.
```

# COBOL Cheat Sheet

## Hilfestellungen für Aufgabe 3

### Arrays in Cobol

```
1 01 FirstArray  
2 05 ITEM OCCURS 10 TIMES PIC x VALUE "0"
```

So wird ein Array der Länge 10 mit String Variablen definiert. Jedes Element hat als Defaultwert "0".

```
1 ITEM(3)
```

So wird auf das dritte Element eines Array zugegriffen.

### Consolen Eingaben in Cobol

```
1 ACCEPT EINGABE.
```

Mit *ACCEPT* wird die Eingabe des Users in die Variable EINGABE geschrieben.

### Systemcalls in Cobol

```
1 CALL "SYSTEM" USING "cat 'hello world'"
```

Mit *CALL "SYSTEM"* können direkt im Programm Komandozeilen Befehle aufgerufen werden

### Floating Points in Cobol

```
1 01 FLOATING_POINT PIC 9V9(4)
```

Definiert eine Kommzahl mit vier Nachkommastellen

### Zufallszahlen in Cobol

```
1 COMPUTE RANDOM_NUMBER = FUNCTION RANDOM
```

Weist der Variable *RANDOM\_NUMBER* eine zufällige Zahl zwischen 0.0 und 1.0 zu.