

# PCEP-30-02 1.1 – Grundlegende Begriffe und Definitionen verstehen

# Lernziele

Verstehen, was ein Algorithmus, Programm, Ausdruck und Statement ist.

Die Bedeutung von Syntax und Semantik in der Programmierung nachvollziehen.

Den Unterschied zwischen Interpreter und Compiler verstehen.

Grundlagen zu Variablen, Konstanten und Datentypen kennenlernen.

Wichtige Schlüsselkonzepte in Python identifizieren.

# Übersicht der Themen

## Übersicht der Themen

## Übersicht der Themen

1. Grundbegriffe und Definitionen
2. Syntax und Semantik
3. Ausdrücke und Anweisungen (Statements)
4. Interpreter vs. Compiler
5. Variablen, Konstanten und Datentypen
6. Schlüsselbegriffe in Python

# 1. Grundbegriffe und Definitionen (Teil 1)

**Algorithmus:** Eine eindeutige, endliche Abfolge von Schritten zur Lösung eines Problems.

Beispiel: Ein Kochrezept beschreibt die einzelnen Schritte, um ein Gericht zuzubereiten.

**Programm:** Eine Sammlung von Anweisungen, die ein Computer ausführt, um eine Aufgabe zu erfüllen.

Beispiel: Ein Skript, das zwei Zahlen addiert.



# 1. Grundbegriffe und Definitionen (Teil 2)

**Befehl/Statement:** Eine einzelne Anweisung, die eine Aktion im Programm ausführt.

Beispiel: `print("Hallo Welt")` gibt den Text “Hallo Welt” aus.

**Ausdruck (Expression):** Ein Segment, das einen Wert liefert.

Beispiel: `3 + 4` ergibt den Wert 7.



## 2. Syntax und Semantik

**Syntax:** Die formalen Regeln, die definieren, wie Code geschrieben werden muss, um gültig zu sein.

Beispiel: In Python müssen Codeblöcke durch Einrückungen strukturiert werden.

**Semantik:** Die Bedeutung und Interpretation der syntaktisch korrekten Anweisungen.

Beispiel: Bei `x = 5` wird der Variable `x` der Wert 5 zugewiesen.

# Code Beispiel: Syntax und Semantik einer Funktion

Betrachte folgenden Codeblock:

```
def factorial(n):  
    """  
    Berechnet rekursiv die Fakultät von n.  
  
    :param n: Eine nicht-negative ganze Zahl  
    :return: Die Fakultät von n (n!)  
    :raises ValueError: Falls n negativ ist  
    """  
    if n < 0:  
        raise ValueError("n muss eine nicht-negative Zahl sein")  
    if n == 0 or n == 1:  
        return 1  
    return n * factorial(n - 1)
```

Was kann man über Syntax und Semantik dieser Funktion sagen?

# 3. Ausdrücke und Anweisungen (Statements)

**Ausdruck (Expression):** Erzeugt einen Wert, der weiterverwendet werden kann.

Beispiel:  $2 * 3$  liefert den Wert 6.

**Statement (Anweisung):** Führt eine Aktion aus und verändert den Zustand des Programms.

*Unterscheidung:* Ein Ausdruck kann Teil eines Statements sein, während ein Statement auch einen Ausdruck beinhalten kann.



# Code Beispiel: If-Anweisung

If-Anweisung-Beispiel in Python.

*Beschreibung: Zeige, wie du eine If-Anweisung live implementierst, um zu überprüfen, ob eine Variable  $x$  größer als 0 ist.*

# Code Beispiel: Funktion zur Addition

*Ein Funktions-Beispiel in Python.*

*Beschreibung: Zeige, wie du eine Funktion erstellst, die zwei Zahlen addiert, und wie du das Ergebnis live ausgibst.*

## 4. Interpreter vs. Compiler

**Interpreter:** Übersetzt und führt den Code Zeile für Zeile aus.

Beispiel: Python wird in der Regel interpretiert.

**Compiler:** Übersetzt den gesamten Quellcode in Maschinensprache, bevor das Programm ausgeführt wird.

Beispiel: C und C++ werden typischerweise kompiliert.



## **Vorteile:**

- Interpreter: Einfachere Fehlerbehebung und hohe Flexibilität.
- Compiler: Optimierte Ausführungsgeschwindigkeit.

# 5. Variablen, Konstanten und Datentypen (Teil 1)

**Variable:** Ein benannter Speicherplatz, der Werte speichern kann.

Beispiel:  $a = 10$

**Konstante:** Ein Wert, der nach der Initialisierung nicht mehr verändert werden sollte.

Beispiel:  $PI = 3.14159$

# 5. Variablen, Konstanten und Datentypen (Teil 2)

## Datentypen:

- **Integer:** Ganze Zahlen (z. B. 5, -3)
- **Float:** Kommazahlen (z. B. 3.14, -0.001)
- **String:** Zeichenketten (z. B. “Hallo”)
- **Boolean:** Wahrheitswerte (True oder False)

# 6. Schlüsselbegriffe in Python

## (Teil 1)

**Operatoren:** Symbole zur Durchführung von Operationen (z. B. +, -, \*, /).

**Kontrollstrukturen:** Anweisungen, die den Ablauf eines Programms steuern (z. B. `if`, `for`, `while`).

# 6. Schlüsselbegriffe in Python

## (Teil 2)

**Funktionen:** Wiederverwendbare Codeblöcke, die eine spezifische Aufgabe erfüllen.

Beispiel: Eine Funktion, die zwei Zahlen addiert.

**Module und Pakete:** Dienen der Organisation und Strukturierung des Codes.

# Multiple-Choice Fragen

# Frage 1

Was ist ein Algorithmus?

- A) Eine fehlerhafte Anweisung
- B) Eine endliche Abfolge von Schritten zur Lösung eines Problems
- C) Ein Programm, das nur in Python geschrieben werden kann
- D) Ein Compiler





# Frage 1 – Antwort

Was ist ein Algorithmus?

A) Eine fehlerhafte Anweisung

**B) Eine endliche Abfolge von Schritten zur Lösung eines Problems (Korrekt)**

C) Ein Programm, das nur in Python geschrieben werden kann

D) Ein Compiler

Erklärung: Ein Algorithmus beschreibt klar und endlich, wie ein Problem schrittweise gelöst werden kann.

## Frage 2

Was beschreibt die Syntax in der Programmierung?

- A) Die Bedeutung des Codes
- B) Die logische Reihenfolge der Befehle
- C) Die formalen Regeln zur Strukturierung des Codes
- D) Die Geschwindigkeit der Ausführung

## Frage 2 – Antwort

Was beschreibt die Syntax in der Programmierung?

A) Die Bedeutung des Codes

B) Die logische Reihenfolge der Befehle

**C) Die formalen Regeln zur Strukturierung des Codes (Korrekt)**

D) Die Geschwindigkeit der Ausführung

Erklärung: Die Syntax legt fest, wie der Code korrekt geschrieben werden muss.

## Frage 3

Welche Aussage trifft auf einen Ausdruck (Expression) zu?

- A) Er führt eine Aktion aus
- B) Er erzeugt einen Wert
- C) Er ändert den Zustand des Programms
- D) Er speichert Daten dauerhaft

# Frage 3 – Antwort

Welche Aussage trifft auf einen Ausdruck (Expression) zu?

- A) Er führt eine Aktion aus
- B) Er erzeugt einen Wert (Korrekt)**
- C) Er ändert den Zustand des Programms
- D) Er speichert Daten dauerhaft

Erklärung: Ein Ausdruck wird ausgewertet und liefert einen Wert.



## Frage 4

Was ist der Hauptunterschied zwischen einem Interpreter und einem Compiler?

A) Ein Interpreter übersetzt den gesamten Code vor der Ausführung, während ein Compiler zeilenweise arbeitet.

B) Ein Compiler übersetzt den gesamten Code in Maschinensprache, während ein Interpreter zeilenweise übersetzt.

C) Beide arbeiten identisch.

D) Ein Interpreter wird nur für JavaScript verwendet.

# Frage 4 – Antwort

Was ist der Hauptunterschied zwischen einem Interpreter und einem Compiler?

A) Ein Interpreter übersetzt den gesamten Code vor der Ausführung, während ein Compiler zeilenweise arbeitet.

**B) Ein Compiler übersetzt den gesamten Code in Maschinensprache, während ein Interpreter zeilenweise übersetzt. (Korrekt)**

C) Beide arbeiten identisch.

D) Ein Interpreter wird nur für JavaScript verwendet.

Erklärung: Compiler übersetzen den gesamten Quellcode, bevor das Programm läuft, während Interpreter den Code schrittweise ausführen.

## Frage 5

Was ist eine Variable in der Programmierung?

- A) Ein unveränderlicher Wert
- B) Ein benannter Speicherplatz, der Werte speichern kann
- C) Ein spezieller Datentyp
- D) Eine Funktion, die Berechnungen durchführt

# Frage 5 – Antwort

Was ist eine Variable in der Programmierung?

- A) Ein unveränderlicher Wert
- B) Ein benannter Speicherplatz, der Werte speichern kann (Korrekt)**
- C) Ein spezieller Datentyp
- D) Eine Funktion, die Berechnungen durchführt

Erklärung: Eine Variable ist ein benannter Ort im Speicher, an dem Daten abgelegt werden.

## Frage 6

Welche der folgenden Aussagen über Datentypen in Python ist korrekt?

- A) Python hat nur Integer- und String-Datentypen.
- B) Boolean-Datentypen können nur die Werte 0 und 1 annehmen.
- C) Float-Datentypen repräsentieren Kommazahlen.



D) Konstanten können nach der Definition verändert werden.

# Frage 6 – Antwort

Welche der folgenden Aussagen über Datentypen in Python ist korrekt?

A) Python hat nur Integer- und String-Datentypen.

B) Boolean-Datentypen können nur die Werte 0 und 1 annehmen.

**C) Float-Datentypen repräsentieren Kommazahlen. (Korrekt)**

D) Konstanten können nach der Definition verändert werden.

Erklärung: Float-Datentypen dienen der Darstellung von Kommazahlen.

# Frage 7

Welche der folgenden Optionen beschreibt eine Funktion in Python am besten?

A) Ein wiederverwendbarer Codeblock, der eine bestimmte Aufgabe erfüllt.

B) Ein Syntaxfehler, der vermieden werden muss.

C) Ein Datentyp, der nur numerische Werte speichert.

D) Ein Befehl, der den Code beendet.

# Frage 7 – Antwort

Welche der folgenden Optionen beschreibt eine Funktion in Python am besten?

**A) Ein wiederverwendbarer Codeblock, der eine bestimmte Aufgabe erfüllt. (Korrekt)**

B) Ein Syntaxfehler, der vermieden werden muss.

C) Ein Datentyp, der nur numerische Werte speichert.

D) Ein Befehl, der den Code beendet.

Erklärung: Funktionen kapseln wiederkehrende Aufgaben in einem Codeblock, der mehrfach verwendet werden kann.

## Frage 8

Was wird durch die Semantik eines Programms definiert?

- A) Die Schreibweise und Formatierung des Codes
- B) Die logische Bedeutung und das Verhalten des Codes
- C) Die Reihenfolge der Befehle
- D) Die Hardware, auf der das Programm läuft





# Frage 8 – Antwort

Was wird durch die Semantik eines Programms definiert?

A) Die Schreibweise und Formatierung des Codes

**B) Die logische Bedeutung und das Verhalten des Codes (Korrekt)**

C) Die Reihenfolge der Befehle

D) Die Hardware, auf der das Programm läuft

Erklärung: Semantik bezieht sich auf die Bedeutung und Funktionsweise des Programmcodes.

## Frage 9

Welcher Begriff beschreibt in Python die Organisation und Strukturierung von Code in mehreren Dateien?

A) Funktion

B) Modul

C) Variable

D) Ausdruck



# Frage 9 – Antwort

Welcher Begriff beschreibt in Python die Organisation und Strukturierung von Code in mehreren Dateien?

A) Funktion

**B) Modul (Korrekt)**

C) Variable

D) Ausdruck

Erklärung: Module ermöglichen es, den Code in übersichtliche, wiederverwendbare Teile zu gliedern.

## Frage 10

Was ist ein Statement in der Programmierung?

- A) Ein Ausdruck, der immer einen Wert zurückgibt
- B) Eine Anweisung, die eine Aktion im Programm ausführt
- C) Ein Befehl, der den Code formatiert
- D) Eine Beschreibung der Programmlogik



# Frage 10 – Antwort

Was ist ein Statement in der Programmierung?

- A) Ein Ausdruck, der immer einen Wert zurückgibt
- B) Eine Anweisung, die eine Aktion im Programm ausführt (Korrekt)**
- C) Ein Befehl, der den Code formatiert
- D) Eine Beschreibung der Programmlogik

Erklärung: Statements sind Anweisungen, die bestimmte Aktionen im Programm auslösen und den Zustand verändern.

