

# Ermitteln Sie Ihr Python-Kompetenzniveau

Ein umfassender Selbsttest

## Einleitung

Willkommen zu Ihrem persönlichen Python-Kompetenztest! Egal, ob Sie gerade erst Ihre Reise in die Welt der Programmierung beginnen oder bereits erfahrene Schritte auf dem Gebiet der Softwareentwicklung unternommen haben, dieser Test wurde speziell entwickelt, um Ihnen zu helfen, Ihre aktuellen Fähigkeiten und Kenntnisse in Python präzise zu bewerten.

## Ziel des Tests

Unser Ziel ist es, Ihnen nicht nur eine klare Vorstellung von Ihrem aktuellen Stand in verschiedenen Bereichen der Python-Programmierung zu geben, sondern auch, Sie auf Ihrem Weg zur Verbesserung Ihrer Fähigkeiten zu unterstützen. Der Test ist in mehrere Abschnitte unterteilt, die verschiedene Aspekte abdecken – von grundlegenden Programmier- und Python-Konzepten über Datenanalyse und Data Science bis hin zu spezialisierten Themen wie Netzwerktechnologie und objektorientierter Programmierung.

## Motivation

Wir wissen, dass das Erlernen einer Programmiersprache manchmal entmutigend sein kann, aber gleichzeitig ist es ein unglaublich lohnender Weg. Diese Tests sind Ihr Kompass – sie zeigen Ihnen, wo Sie jetzt stehen, und helfen Ihnen, den besten Weg vorwärts zu planen. Egal, ob Sie Ihre Grundlagen festigen oder sich in anspruchsvollere Gebiete vorwagen möchten, dieser Test ist der erste Schritt auf einer spannenden Reise der kontinuierlichen Verbesserung und des Lernens.

## Wie Sie den Test verwenden können:

- **Selbsteinschätzung:** Beantworten Sie die Fragen jedes Tests, um eine Einschätzung Ihres Wissensstandes in den jeweiligen Bereichen zu erhalten.
- **Entwicklungspfad:** Nutzen Sie die Ergebnisse, um Bereiche zu identifizieren, in denen Sie sich weiterbilden können. Vor allem scheuen Sie sich nicht „das weiss ich noch nicht“ auszuwählen, da Ihnen das auch konkrete Hinweise auf Themen gibt, die Sie in der Literatur oder mit Hilfe des Internets anschauen sollten.
- **Kursauswahl:** Die Testergebnisse helfen Ihnen, die für Sie am besten geeigneten Kursteile zu identifizieren und so Ihren Lernpfad effektiver zu gestalten.

## Abschließende Worte

Denken Sie daran, dass jeder Test nicht nur eine Bewertung ist, sondern auch eine Lerngelegenheit. Fehler sind Teil des Lernprozesses, und jede Frage, die Sie heute vielleicht nicht beantworten können, könnte der erste Schritt zu einem tieferen Verständnis von Python morgen sein. Viel Erfolg und vor allem viel Spaß beim Testen Ihrer Python-Kenntnisse!

# Individueller Lernpfad basierend auf Testergebnissen

## Der Lernpfad

Für Teilnehmer mit geringen oder grundlegenden Programmierkenntnissen und geringen Kenntnissen in Python

- **Block 1 – Programmiergrundlagen verstärken**
  - Fokus auf das Verständnis von grundlegenden Programmierkonzepten wie Variablen, Schleifen und bedingte Anweisungen.
  - Einführung in die Syntax und Grundlagen von Python.
  - Voraussetzung für den folgenden Block

Für Teilnehmer mit guten Grundkenntnissen in der Programmierung mit Python

- **Block 2 – Python-Kenntnisse erweitern**
  - Vertiefung in Python-spezifische Konzepte wie Listen, Tupel und Dictionaries.
  - Einführung in Funktionen und einfache Datenmanipulationen.
  - Wenn abgeschlossen, liegen solide Kenntnisse für den Einsatz von Python in Skripten vor, die „klassisch“, also nicht objektorientiert entwickelt werden.
  - Voraussetzung für den folgenden Block

Für Teilnehmer mit Fortgeschrittenen Python-Kenntnissen

- **Block 3 – Erweiterung der Python-Fähigkeiten**
  - Erkundung fortgeschrittener Themen wie Klassen und Objekte (unter Berücksichtigung der OOP-Ergebnisse).
  - Vertiefung in fortgeschrittene Datenstrukturen und Algorithmen.
  - Abschluss der Einführung und Vertiefung in Python. Folgeblöcke sind optional
  - Voraussetzung für die Folgeblöcke

Für Teilnehmer mit fortgeschrittenen Kenntnissen in Python und Interesse an der Datenanalyse

- **Block 4 – Datenanalyse und Data Science**
  - Praktische Anwendung von Datenanalyse-Techniken mit pandas und NumPy.
  - Einführung in Data Science-Methoden und maschinelles Lernen.
  - Optionaler Block

## Für Teilnehmer mit Fortgeschrittenen Kenntnissen in Python und Interesse an der Netzwerktechnologie

- **Block 5 – Netzwerkprogrammierung und Sicherheit**
  - Vertiefung in Netzwerkprotokolle und client-seitige/server-seitige Programmierung.
  - Einführung in Netzwerksicherheit und fortgeschrittene Netzwerkkonzepte.
  - Optionaler Block

## Allgemeine Empfehlungen

- **Praxisorientiertes Lernen:** Regelmäßiges Üben durch realistische Programmierprojekte, um die theoretischen Kenntnisse praktisch anzuwenden.
- **Teilnahme an Workshops und Seminaren:** Besuch von spezialisierten Kursen oder Workshops, um spezifische Fähigkeiten zu vertiefen.
- **Kollaboratives Lernen:** Austausch mit Gleichgesinnten oder Beitritt zu einer Lerngruppe, um von kollektiven Erfahrungen zu profitieren.
- **Kontinuierliche Selbstbewertung:** Regelmäßige Überprüfung des eigenen Fortschritts und Anpassung des Lernpfades an individuelle Bedürfnisse und Ziele.

Diese Empfehlungen sollen den Teilnehmenden helfen, ihren Lernprozess effektiv zu gestalten und ihre Kenntnisse in den relevanten Bereichen der Python-Programmierung gezielt zu erweitern.

# Informationen zur Testauswertung

Hier möchten wir Ihnen Hinweise geben, wie Sie die Tests benutzen können. Dabei ist es uns sehr wichtig, darauf hinzuweisen, dass es sich nicht um eine Art Prüfung handelt. Das heißt, wenn Sie etwas wirklich nicht wissen, wählen Sie „Das weiss ich noch nicht“ aus.

Nachdem Sie einen Test gemacht haben, nehmen Sie sich die Zeit diesen erst zu bewerten. Die vorgeschlagenen Ergebnisse finden Sie im hinteren Teil dieses Dokuments.

## Ihre Auswertung

- Zählen Sie die „Das weiss ich noch nicht“ Einträge. Damit haben Sie starke Hinweise auf die Einzelthemen, die Sie in den Lernblöcken bearbeiten können. Je öfter Sie diese Option gewählt haben, desto mehr möchten wir ihnen die jeweiligen Kurse empfehlen. Wenn es mehr als 50% sind, sollten Sie unbedingt den jeweiligen Kurs belegen
- Zählen Sie Ihre Punkte. Um die Sache einfach zu machen, haben wir die Fragen nicht gewichtet. Wir halten jedes der angesprochenen Themen für gleich wichtig, um am Ende ein gutes Verständnis der Programmierung mit Python zu haben. Darauf basierend haben wir eine Einschätzung versucht, wo Sie stehen könnten und was ein guter weiterer Schritt auf ihrem Lernpfad sein könnte.

## Die Einschätzung basierend auf der Punktzahl

- **0-4 Punkte:** Einsteiger
  - Personen in diesem Bereich haben keine bis wenig Kenntnisse des Themas. Daher möchten wir das jeweilige Kursthema bzw. den Kursblock empfehlen. Achten Sie unbedingt auf die Punkte bei denen Sie „Das weiss ich noch nicht“ ausgewählt haben und versuchen Sie ihren Lernweg innerhalb des Block daran auszurichten.
- **5-9 Punkte:** Grundkenntnisse
  - Mit dieser Punktzahl verfügen die Teilnehmenden über Grundkenntnisse in dem Thema. Sie sind vertraut mit Basiskonzepten, fortgeschrittene Konzepte stellen aber sicherlich noch eine grössere Herausforderung dar, wir möchten auch Ihnen die Belegung der jeweiligen Kursthemen empfehlen.
- **10-14 Punkte:** Fortgeschritten
  - Teilnehmende in diesem Bereich haben ein solides Verständnis der Thematik und können eventuell zum nächsten Themenkomplex weiterspringen. Allerdings möchten wir Sie bitten noch einmal genau die „Das weiss ich noch nicht“-Antworten auszuwerten. Evtl. sollten Sie zur Auffrischung und Vertiefung den angesprochenen Themenbereich doch noch einmal bearbeiten.

- **15-20 Punkte:** Erfahren

- Personen mit einer Punktzahl in diesem Bereich zeigen ein umfassendes Verständnis der angesprochenen Themenfelder und können diesen Themenkomplex überspringen und sich weiteren Folgethemen zuwenden.

# Test zu allgemeinen Programmierkenntnissen

## Fragen

### 1. Was ist eine Variable in der Programmierung?

- ☐ a) Ein reservierter Speicherplatz zur Speicherung von Werten
- ☐ b) Ein Fehler im Code
- ☐ c) Ein Programmierwerkzeug
- ☐ d) Eine Art von Schleife
- ☐ e) Das weiss ich noch nicht

### 2. Was macht eine 'for'-Schleife?

- ☐ a) Führt einen Codeblock einmal aus
- ☐ b) Wiederholt einen Codeblock eine bestimmte Anzahl von Malen
- ☐ c) Wiederholt einen Codeblock, solange eine Bedingung wahr ist
- ☐ d) Teilt den Code in Module auf
- ☐ e) Das weiss ich noch nicht

### 3. Was ist ein Array?

- ☐ a) Eine Sammlung von Werten unterschiedlicher Typen
- ☐ b) Ein einzelner Wert
- ☐ c) Eine Sammlung von Werten des gleichen Typs, die über Indizes zugänglich sind
- ☐ d) Eine spezielle Art von Schleife
- ☐ e) Das weiss ich noch nicht

### 4. Was ist eine Funktion in der Programmierung?

- ☐ a) Ein Codeblock, der eine bestimmte Aufgabe ausführt und wiederholt aufgerufen werden kann
- ☐ b) Ein Datentyp
- ☐ c) Ein reserviertes Schlüsselwort

- d) Ein Teil des Speichers
- e) Das weiss ich noch nicht

**5. Was ist ein Pseudocode?**

- a) Eine informelle Beschreibung eines Algorithmus
- b) Ein spezieller Code, der nur in einer Programmiersprache verwendet wird
- c) Ein fehlerhafter Code
- d) Ein Code, der in einer nicht existierenden Programmiersprache geschrieben ist
- e) Das weiss ich noch nicht

**6. Welche Aussage über 'Schleifen' ist korrekt?**

- a) Schleifen erlauben die wiederholte Ausführung eines Codeblocks unter bestimmten Bedingungen
- b) Eine Schleife wird verwendet, um Daten in einer Datenbank zu speichern
- c) Schleifen können nur eine endliche Anzahl von Durchläufen haben
- d) Schleifen sind eine Form der Datenverschlüsselung
- e) Das weiss ich noch nicht

**7. Was ist ein Algorithmus in der Programmierung?**

- a) Eine Sammlung von Daten
- b) Eine Reihe von Anweisungen zur Lösung eines Problems
- c) Ein spezieller Datentyp
- d) Ein Programmierfehler
- e) Das weiss ich noch nicht

**8. Was bedeutet 'Kompilierung' in der Programmierung?**

- a) Umwandlung von Quellcode in ausführbaren Code
- b) Das Testen eines Programms auf Fehler
- c) Das Hochladen eines Programms auf einen Server
- d) Das Schreiben von Code in einer Hochsprache
- e) Das weiss ich noch nicht

**9. Was ist der Unterschied zwischen einem Compiler und einem Interpreter?**

- a) Ein Compiler übersetzt den gesamten Quellcode auf einmal, während ein Interpreter den Code zeilenweise ausführt
- b) Ein Compiler wird nur in Webentwicklung verwendet, ein Interpreter nur in der Softwareentwicklung
- c) Ein Compiler ist eine Art von Datenbank, ein Interpreter eine Art von Algorithmus
- d) Es gibt keinen Unterschied
- e) Das weiss ich noch nicht

**10. Was sind Operatoren in der Programmierung?**

- a) Werkzeuge zur Fehlerbehebung im Code
- b) Befehle zur Steuerung des Programmablaufs
- c) Symbole, die spezifische Operationen auf Variablen oder Werte ausführen
- d) Komponenten, die zum Speichern von Daten verwendet werden
- e) Das weiss ich noch nicht

**11. Was ist der Zweck von 'if'-Anweisungen in der Programmierung?**

- a) Um eine Bedingung zu überprüfen und basierend darauf unterschiedliche Codeblöcke auszuführen
- b) Um Schleifen zu erstellen
- c) Um Daten in einer Datenbank zu speichern
- d) Um Fehler im Code zu identifizieren
- e) Das weiss ich noch nicht

**12. Was ist ein Modul in der Programmierung?**

- a) Eine Sammlung von verwandten Funktionen und Klassen
- b) Ein Fehler im Code
- c) Ein reservierter Speicherplatz für Variablen
- d) Ein Befehl, der ein Programm startet
- e) Das weiss ich noch nicht

**13. Was ist eine Schleife in der Programmierung?**

- a) Eine Funktion, die sich selbst aufruft



- b) Ein Codeblock, der unter bestimmten Bedingungen wiederholt ausgeführt wird
- c) Eine Methode, um Daten in einer Datenbank zu speichern
- d) Ein Fehler im Code, der eine unendliche Ausführung verursacht
- e) Das weiss ich noch nicht

**14. Was ist eine 'while'-Schleife in der Programmierung?**

- a) Eine Schleife, die eine unbestimmte Anzahl von Malen ausgeführt wird, solange eine Bedingung wahr bleibt
- b) Eine Schleife, die nur einmal ausgeführt wird
- c) Ein spezieller Typ von Funktion
- d) Ein Werkzeug zur Datenanalyse
- e) Das weiss ich noch nicht

**15. Was ist der Zweck von 'import'-Anweisungen in der Programmierung?**

- a) Um Code aus anderen Modulen oder Bibliotheken im aktuellen Programm verfügbar zu machen
- b) Um ein Programm schneller auszuführen
- c) Um Daten in einer Datenbank zu speichern
- d) Um Fehler im Code zu beheben
- e) Das weiss ich noch nicht

**16. Was beschreibt den Begriff 'Syntax' in der Programmierung?**

- a) Die Regeln und Struktur, die bestimmen, wie Code geschrieben werden muss, damit er vom Computer verstanden wird
- b) Eine Sammlung von Funktionen in einer Programmiersprache
- c) Die Geschwindigkeit, mit der ein Programm ausgeführt wird
- d) Die Benutzeroberfläche eines Programms
- e) Das weiss ich noch nicht

**17. Was ist ein logischer Operator in der Programmierung?**

- a) Ein Werkzeug zur Fehlerbehebung
- b) Ein Operator, der verwendet wird, um logische Verknüpfungen zwischen Ausdrücken herzustellen (z.B. AND, OR, NOT)

- c) Ein Teil eines Computers, der Logikoperationen durchführt
- d) Ein spezielles Programm zur Datenanalyse
- e) Das weiss ich noch nicht

**18. Was ist ein Kommentar im Code?**

- a) Ein Teil des Codes, der bei der Ausführung ignoriert wird und der in erster Linie dazu dient, den Code zu dokumentieren oder zu erklären
- b) Ein Fehler im Code
- c) Eine Anweisung, die eine Aktion im Programm auslöst
- d) Ein reservierter Bereich im Speicher
- e) Das weiss ich noch nicht

**19. Was bedeutet 'Deklaration' in der Programmierung?**

- a) Der Prozess der Zuweisung eines Wertes zu einer Variable
- b) Die Ankündigung einer Variablen oder Funktion, ohne ihr einen spezifischen Wert oder Körper zuzuweisen
- c) Das Schreiben von Code in einer Hochsprache
- d) Das Entfernen von Fehlern aus einem Programm
- e) Das weiss ich noch nicht

**20. Was ist der Zweck eines 'return'-Statements in einer Funktion?**

- a) Um einen Wert aus einer Funktion zurückzugeben
- b) Um eine Schleife zu beenden
- c) Um eine Bedingung zu überprüfen
- d) Um eine Fehlermeldung zu generieren
- e) Das weiss ich noch nicht

# Test zur Ermittlung grundlegender Python-Kenntnisse

## Fragen

### 1. Wie erstellt man in Python eine Variable?

- ☐ a) variable create myVar = 5
- ☐ b) myVar = 5
- ☐ c) var myVar = 5
- ☐ d) int myVar = 5
- ☐ e) Das weiss ich noch nicht

### 2. Wie wird in Python ein Kommentar gekennzeichnet?

- ☐ a) // Dies ist ein Kommentar
- ☐ b) <!-- Dies ist ein Kommentar -->
- ☐ c) # Dies ist ein Kommentar
- ☐ d) /\* Dies ist ein Kommentar \*/
- ☐ e) Das weiss ich noch nicht

### 3. Welcher Datentyp wird in Python NICHT unterstützt?

- ☐ a) list
- ☐ b) array
- ☐ c) tuple
- ☐ d) dictionary
- ☐ e) Das weiss ich noch nicht

### 4. Wie wird in Python eine Liste erstellt?

- ☐ a) meineListe = list(1, 2, 3)
- ☐ b) meineListe = [1, 2, 3]
- ☐ c) meineListe = (1, 2, 3)
- ☐ d) meineListe = {1, 2, 3}

- e) Das weiss ich noch nicht

**5. Wie greift man auf das erste Element einer Liste namens 'meineListe' zu?**

- a) meineListe[0]
- b) meineListe[1]
- c) meineListe.first()
- d) meineListe.get(0)
- e) Das weiss ich noch nicht

**6. Welche Schleife gibt es in Python NICHT?**

- a) for
- b) while
- c) do-while
- d) foreach
- e) Das weiss ich noch nicht

**7. Wie wird in Python eine bedingte Anweisung geschrieben?**

- a) if x > 10 then:
- b) if (x > 10):
- c) if x > 10:
- d) if x > 10 ->
- e) Das weiss ich noch nicht

**8. Wie definiert man eine Funktion in Python?**

- a) function meineFunktion():
- b) def meineFunktion():
- c) meineFunktion() =>:
- d) func meineFunktion():
- e) Das weiss ich noch nicht

**9. Wie gibt eine Funktion in Python einen Wert zurück?**

- a) return: Wert

- ☐ b) return Wert
- ☐ c) -> Wert
- ☐ d) output Wert
- ☐ e) Das weiss ich noch nicht

**10. Was wird nach Ausführung dieses Codes ausgegeben? `x = 5; y = x + 2; print(y)`**

- ☐ a) 5
- ☐ b) 7
- ☐ c) `x + 2`
- ☐ d) Fehler
- ☐ e) Das weiss ich noch nicht

**11. Welcher Operator wird für die Potenzierung in Python verwendet?**

- ☐ a) ^
- ☐ b) \*\*
- ☐ c) ^^
- ☐ d) `pow()`
- ☐ e) Das weiss ich noch nicht

**12. Wie wird ein mehrzeiliger String in Python definiert?**

- ☐ a) "Dies ist ein mehrzeiliger String"
- ☐ b) "'Dies ist ein mehrzeiliger String'" (jeweils 3 Anführungszeichen)
- ☐ c) // Dies ist ein mehrzeiliger String //
- ☐ d) ("Dies ist " + "ein mehrzeiliger " + "String")
- ☐ e) Das weiss ich noch nicht

**13. Wie wird in Python der Datentyp eines Objekts überprüft?**

- ☐ a) `check(type)`
- ☐ b) `type()`
- ☐ c) `typeof()`
- ☐ d) `get_type()`
- ☐ e) Das weiss ich noch nicht

**14. Wie wird in Python ein Wörterbuch (dictionary) erstellt?**

- a) `meinWörterbuch = dict("schlüssel1":"wert1", "schlüssel2":"wert2")`
- b) `meinWörterbuch = {"schlüssel1":"wert1", "schlüssel2":"wert2"}`
- c) `meinWörterbuch = ["schlüssel1":"wert1", "schlüssel2":"wert2"]`
- d) `meinWörterbuch = ("schlüssel1":"wert1", "schlüssel2":"wert2")`
- e) Das weiss ich noch nicht

**15. Wie wird ein Fehler in Python abgefangen?**

- a) `catch Fehler:`
- b) `try: ... except Fehler:`
- c) `try: ... catch Fehler:`
- d) `error: ... except:`
- e) Das weiss ich noch nicht

**16. Was macht der `len()` Befehl in Python?**

- a) Ändert die Länge eines Objekts
- b) Gibt die Länge eines Objekts zurück
- c) Startet eine Schleife über die Länge eines Objekts
- d) Gibt einen Fehler, wenn das Objekt keine Länge hat
- e) Das weiss ich noch nicht

**17. Wie fügt man in Python einem String einen Wert hinzu?**

- a) `"String" + Wert`
- b) `"String".add(Wert)`
- c) `"String".append(Wert)`
- d) `"String".concat(Wert)`
- e) Das weiss ich noch nicht

**18. Wie wird in Python eine unendliche Schleife erstellt?**

- a) `while True:`
- b) `for ;;:`

- c) while (1==1):
- d) for ever:
- e) Das weiss ich noch nicht

**19. Wie wird in Python ein Element aus einer Liste entfernt?**

- a) meineListe.remove(Element)
- b) meineListe.delete(Element)
- c) meineListe.pop(Element)
- d) meineListe.cut(Element)
- e) Das weiss ich noch nicht

**20. Wie importiert man in Python ein Modul namens 'math'?**

- a) import math
- b) include math
- c) using math
- d) require math
- e) Das weiss ich noch nicht

# Test zur Ermittlung fortgeschrittener Python-Kenntnisse

## Fragen

### 1. Was macht die `enumerate()` Funktion in Python?

- ☐ a) Zählt die Anzahl der Elemente in einer Liste
- ☐ b) Fügt automatisch eine Nummerierung zu den Elementen einer Sequenz hinzu
- ☐ c) Wandelt eine Liste in ein Enum um
- ☐ d) Sortiert eine Liste
- ☐ e) Das weiss ich noch nicht

### 2. Was ist 'List Comprehension' in Python?

- ☐ a) Eine Methode, um Listen zu komprimieren
- ☐ b) Eine Methode, um Listen schnell zu erstellen und zu manipulieren
- ☐ c) Eine spezielle Bibliothek für Listenoperationen
- ☐ d) Eine Schleifenstruktur speziell für Listen
- ☐ e) Das weiss ich noch nicht

### 3. Wie implementiert man in Python eine anonyme (Lambda) Funktion?

- ☐ a) `lambda x, y: x + y`
- ☐ b) `function(x, y) => x + y`
- ☐ c) `def (x, y): return x + y`
- ☐ d) `anonymous(x, y): x + y`
- ☐ e) Das weiss ich noch nicht

### 4. Wie wird ein Generator in Python erstellt?

- ☐ a) Mit dem Schlüsselwort `generator`
- ☐ b) Durch eine Funktion mit `yield`-Anweisungen
- ☐ c) Mit einer speziellen Klasse `Generator`
- ☐ d) Durch die Verwendung von `()`



- e) Das weiss ich noch nicht

**5. Wie behandelt man in Python eine Datei?**

- a) Mit `file.open()` und `file.close()`
- b) Mit `open()` und `close()`
- c) Mit `File()` und `File.close()`
- d) Mit `openFile()` und `endFile()`
- e) Das weiss ich noch nicht

**6. Was ist der Zweck des `with`-Statements in Python?**

- a) Zum Erstellen von Schleifen
- b) Zur Fehlerbehandlung
- c) Zur effizienten und sicheren Handhabung externer Ressourcen wie Dateien
- d) Zum Importieren von Modulen
- e) Das weiss ich noch nicht

**7. Was macht der `map()` Befehl in Python?**

- a) Kartiert Daten auf eine geographische Karte
- b) Wendet eine Funktion auf jedes Element einer Liste oder eines anderen iterierbaren Datentyps an
- c) Erstellt eine Abbildung von Schlüsseln zu Werten
- d) Zeichnet Grafiken basierend auf Daten
- e) Das weiss ich noch nicht

**8. Wie wird ein mehrzeiliger String in Python definiert?**

- a) `"Dies ist ein mehrzeiliger String"`
- b) `""Dies ist ein mehrzeiliger String""`
- c) `// Dies ist ein mehrzeiliger String //`
- d) `("Dies ist " + "ein mehrzeiliger " + "String")`
- e) Das weiss ich noch nicht

**9. Wie erstellt man in Python ein Set (Menge)?**

- a) `meinSet = set([1, 2, 3])`
- b) `meinSet = {1, 2, 3}`
- c) `meinSet = (1, 2, 3)`
- d) `meinSet = [1, 2, 3]`
- e) Das weiss ich noch nicht

**10. Wie wird in Python ein regulärer Ausdruck definiert und verwendet?**

- a) `re.compile('[a-z]+')`
- b) `regex('[a-z]+')`
- c) `r'[a-z]+'`
- d) `regular('[a-z]+')`
- e) Das weiss ich noch nicht

**11. Was macht das `zip()` in Python?**

- a) Komprimiert Dateien
- b) Verbindet zwei Listen zu einer Liste von Tupeln
- c) Entpackt zip-Dateien
- d) Erstellt ein Backup der Daten
- e) Das weiss ich noch nicht

**12. Wie fängt man spezifische Fehler in Python ab?**

- a) `try: ... catch FehlerTyp:`
- b) `if error: ...`
- c) `try: ... except FehlerTyp:`
- d) `error: ... except:`
- e) Das weiss ich noch nicht

**13. Was ist eine korrekte Art, einen String in Python zu formatieren?**

- a) `"{0} ist {1}".format("Python", "toll")`
- b) `"Python ist " + "toll"`
- c) `"Python ist %s" % "toll"`
- d) Alle oben genannten

- e) Das weiss ich noch nicht

**14. Wie wird in Python eine unendliche Schleife erstellt?**

- a) while True:
- b) for ;;;
- c) while (1==1):
- d) for ever:
- e) Das weiss ich noch nicht

**15. Wie wird in Python ein Element aus einer Liste entfernt?**

- a) meineListe.remove(Element)
- b) meineListe.delete(Element)
- c) meineListe.pop(Element)
- d) meineListe.cut(Element)
- e) Das weiss ich noch nicht

**16. Wie kann man in Python überprüfen, ob ein Element in einer Liste vorhanden ist?**

- a) inList(Element, meineListe)
- b) meineListe.contains(Element)
- c) Element in meineListe
- d) meineListe.has(Element)
- e) Das weiss ich noch nicht

**17. Wie wird in Python eine tiefe Kopie eines Objekts erstellt?**

- a) Mit dem copy-Modul und der Funktion deepcopy
- b) Durch Zuweisung (neuesObjekt = altesObjekt)
- c) Mit der Methode clone()
- d) Durch manuelles Kopieren aller Elemente
- e) Das weiss ich noch nicht

**18. Was ist der Output des folgenden Codes? (Zeilennummern sind nur zur Orientierung)**

```
001 x = [1, 2, 3]
```

```
002 y = x
003 y.append(4)
004 print(x)
```

- ☐ a) [1, 2, 3]
- ☐ b) [1, 2, 3, 4]
- ☐ c) [4]
- ☐ d) Fehler
- ☐ e) Das weiss ich noch nicht

**19. Wie wird in Python ein Slice einer Liste erstellt?**

- ☐ a) meineListe[1:3]
- ☐ b) meineListe.slice(1, 3)
- ☐ c) slice(meineListe, 1, 3)
- ☐ d) meineListe[1, 3]
- ☐ e) Das weiss ich noch nicht

**20. Welches Modul wird für die Arbeit mit regulären Ausdrücken in Python verwendet?**

- ☐ a) regex
- ☐ b) re
- ☐ c) regexp
- ☐ d) regular
- ☐ e) Das weiss ich noch nicht

# Test zur Ermittlung der OOP-Kenntnisse in Python

## Fragen

### 1. Wie wird in Python eine Klasse definiert?

- ☐ a) `class MeineKlasse():`
- ☐ b) `MeineKlasse = Class():`
- ☐ c) `new class MeineKlasse():`
- ☐ d) `define class MeineKlasse():`
- ☐ e) Das weiss ich noch nicht

### 2. Wie wird in Python ein neues Objekt einer Klasse erstellt?

- ☐ a) `obj = MeineKlasse.new()`
- ☐ b) `obj = new MeineKlasse()`
- ☐ c) `obj = MeineKlasse()`
- ☐ d) `obj = create MeineKlasse()`
- ☐ e) Das weiss ich noch nicht

### 3. Wie definiert man einen Konstruktor in einer Python-Klasse?

- ☐ a) `def __init__(self):`
- ☐ b) `def MeineKlasse():`
- ☐ c) `def __construct__(self):`
- ☐ d) `def create(self):`
- ☐ e) Das weiss ich noch nicht

### 4. Wie greift man in Python auf eine Eigenschaft eines Objekts zu?

- ☐ a) `obj->eigenschaft`
- ☐ b) `obj.eigenschaft`
- ☐ c) `obj::eigenschaft`
- ☐ d) `obj[eigenschaft]`

- e) Das weiss ich noch nicht

**5. Wie wird in Python Vererbung implementiert?**

- a) class Subklasse(Superklasse):
- b) class Subklasse extends Superklasse:
- c) class Subklasse inherits Superklasse:
- d) class Subklasse: Superklasse
- e) Das weiss ich noch nicht

**6. Wie kann man in Python eine Methode einer Superklasse aufrufen?**

- a) super().methode()
- b) Superklasse.methode(self)
- c) superklasse::methode()
- d) Superklasse.methode()
- e) Das weiss ich noch nicht

**7. Was ist Polymorphismus in Python?**

- a) Eine Klasse, die mehrere Formen annimmt
- b) Methoden mit demselben Namen, die in verschiedenen Klassen unterschiedlich implementiert sind
- c) Die Fähigkeit, mehrere Anweisungen gleichzeitig auszuführen
- d) Das Verändern des Wertes einer Variablen
- e) Das weiss ich noch nicht

**8. Wie definiert man in Python eine abstrakte Klasse?**

- a) class MeineKlasse(abstract):
- b) abstract class MeineKlasse:
- c) class MeineKlasse: pass
- d) from abc import ABC, abstractmethod; class MeineKlasse(ABC):
- e) Das weiss ich noch nicht

**9. Wie wird eine statische Methode in Python definiert?**

- a) `def static methode():`
- b) `@staticmethod def methode():`
- c) `static def methode():`
- d) `def methode(static):`
- e) Das weiss ich noch nicht

**10. Was ist eine Klassenvariable in Python?**

- a) Eine Variable, die innerhalb einer Klasse definiert und nur dort verwendet wird
- b) Eine globale Variable, die in einer Klasse verwendet wird
- c) Eine Variable, die für alle Instanzen einer Klasse gleich ist
- d) Eine Variable, die nur in statischen Methoden verwendet wird
- e) Das weiss ich noch nicht

**11. Wie wird in Python die Verkapselung umgesetzt?**

- a) Durch das Verwenden von privaten und öffentlichen Schlüsselwörtern
- b) Durch das Voranstellen eines Unterstrichs vor den Variablennamen
- c) Durch das Verwenden spezieller Verkapselungsmethoden
- d) In Python gibt es keine Verkapselung
- e) Das weiss ich noch nicht

**12. Wie wird in Python eine Eigenschaft (Property) definiert?**

- a) `property def eigenschaft():`
- b) `def eigenschaft(self):`
- c) `@property def eigenschaft(self):`
- d) `eigenschaft = property()`
- e) Das weiss ich noch nicht

**13. Was ist die Funktion von `__str__` in einer Python-Klasse?**

- a) Gibt eine offizielle String-Darstellung des Objekts zurück
- b) Konvertiert das Objekt in einen String
- c) Ist eine spezielle Methode zum Drucken von Objekten
- d) Gibt den Typ des Objekts als String zurück

- e) Das weiss ich noch nicht

**14. Wie kann in Python eine Klasse eine Schnittstelle implementieren?**

- a) class MeineKlasse(implements Schnittstelle):
- b) Python unterstützt keine Schnittstellen
- c) class MeineKlasse(Schnittstelle):
- d) from interface import implement; class MeineKlasse(implement Schnittstelle):
- e) Das weiss ich noch nicht

**15. Wie kann man in Python mehrere Vererbungen implementieren?**

- a) class Subklasse(Superklasse1, Superklasse2):
- b) class Subklasse extends Superklasse1, Superklasse2:
- c) class Subklasse: Superklasse1, Superklasse2
- d) Mehrfachvererbungen sind in Python nicht möglich
- e) Das weiss ich noch nicht

**16. Wie kann man in Python den Destruktor einer Klasse definieren?**

- a) def \_\_del\_\_(self):
- b) def \_\_destroy\_\_(self):
- c) def \_\_destructor\_\_(self):
- d) def destroy(self):
- e) Das weiss ich noch nicht

**17. Was ist der richtige Weg, um in Python eine private Methode zu definieren?**

- a) def \_\_meineMethode(self):
- b) private def meineMethode(self):
- c) def meineMethode(self): private
- d) def \_meineMethode(self):
- e) Das weiss ich noch nicht

**18. Wie wird in Python das Überladen von Methoden umgesetzt?**



- a) Durch das Definieren von Methoden mit demselben Namen, aber unterschiedlichen Parametern
- b) Python unterstützt kein methodisches Überladen
- c) Durch das Verwenden spezieller Überladungsdekoren
- d) Durch das Verwenden des Schlüsselworts overload
- e) Das weiss ich noch nicht

**19. Was macht die `__call__` Methode in einer Python-Klasse?**

- a) Ermöglicht, dass Instanzen der Klasse wie Funktionen aufgerufen werden können
- b) Ruft eine Methode auf
- c) Ist eine spezielle Methode für Konstruktoren
- d) Wird beim ersten Aufruf der Instanz aufgerufen
- e) Das weiss ich noch nicht

**20. Wie wird in Python der Kopierkonstruktor einer Klasse definiert?**

- a) `def __copy__(self):`
- b) `def __init__(self, other):`
- c) `def __clone__(self):`
- d) Python unterstützt keine Kopierkonstruktoren
- e) Das weiss ich noch nicht

# Test zur Ermittlung der Kenntnisse in Python-Datenanalyse und Data Science

## Fragen

1. **Welche Bibliothek wird in Python am häufigsten für Datenmanipulation und -analyse verwendet?**

- ☐ a) TensorFlow
- ☐ b) Pandas
- ☐ c) Matplotlib
- ☐ d) Scikit-learn
- ☐ e) Das weiss ich noch nicht

2. **Wie erstellt man in Pandas einen DataFrame aus einem Dictionary?**

- ☐ a) `pd.DataFrame.from_dict(mein_dict)`
- ☐ b) `pd.DataFrame(mein_dict)`
- ☐ c) `pd.createDataFrame(mein_dict)`
- ☐ d) `DataFrame(pd, mein_dict)`
- ☐ e) Das weiss ich noch nicht

3. **Welche Funktion wird verwendet, um fehlende Daten in einem Pandas DataFrame zu behandeln?**

- ☐ a) `fillna()`
- ☐ b) `dropna()`
- ☐ c) `replaceNull()`
- ☐ d) `removeNan()`
- ☐ e) Das weiss ich noch nicht

4. **Wie wird in NumPy ein Array mit Nullen erstellt?**

- ☐ a) `np.zeros(5)`
- ☐ b) `np.empty(5)`
- ☐ c) `np.array(5, fill=0)`

- d) np.nulls(5)
- e) Das weiss ich noch nicht

**5. Welche Bibliothek wird in Python hauptsächlich für Datenvisualisierung verwendet?**

- a) Seaborn
- b) Pandas
- c) NumPy
- d) Scikit-learn
- e) Das weiss ich noch nicht

**6. Für was wird groupby() in Pandas verwendet?**

- a) Um Daten in Gruppen zu teilen
- b) Um Reihenfolge in Daten zu bringen
- c) Um Daten zu sortieren
- d) Um Daten zu verschlüsseln
- e) Das weiss ich noch nicht

**7. Welche Funktion wird in Matplotlib verwendet, um ein Diagramm zu erstellen?**

- a) plot()
- b) draw()
- c) show()
- d) graph()
- e) Das weiss ich noch nicht

**8. Wie wird ein DataFrame in Pandas nach einer bestimmten Spalte sortiert?**

- a) df.sort('Spaltenname')
- b) df.order\_by('Spaltenname')
- c) df.sort\_values(by='Spaltenname')
- d) df.arrange('Spaltenname')
- e) Das weiss ich noch nicht

**9. Welche Bibliothek wird in Python für maschinelles Lernen verwendet?**

- a) TensorFlow
- b) PyTorch
- c) Scikit-learn
- d) Keras
- e) Das weiss ich noch nicht

**10. Wie wird in Scikit-learn ein maschinelles Lernmodell trainiert?**

- a) `modell.fit(daten, labels)`
- b) `modell.train(daten, labels)`
- c) `modell.learn(daten, labels)`
- d) `modell.execute(daten, labels)`
- e) Das weiss ich noch nicht

**11. Welche Funktion in NumPy wird verwendet, um eine Zufallszahl zu generieren?**

- a) `np.random()`
- b) `np.randomNumber()`
- c) `np.rand()`
- d) `np.random.randint()`
- e) Das weiss ich noch nicht

**12. Wie wird ein Subset von Spalten in einem Pandas DataFrame ausgewählt?**

- a) `df['Spalte1', 'Spalte2']`
- b) `df[['Spalte1', 'Spalte2']]`
- c) `df.select('Spalte1', 'Spalte2')`
- d) `df.columns['Spalte1', 'Spalte2']`
- e) Das weiss ich noch nicht

**13. Welche Methode wird verwendet, um statistische Zusammenfassungen in Pandas zu erhalten?**

- a) `describe()`
- b) `summary()`
- c) `info()`

- d) statistics()
- e) Das weiss ich noch nicht

**14. Wie wird ein lineares Regressionsmodell in Scikit-learn erstellt?**

- a) LinearRegression().fit(daten, labels)
- b) sklearn.linear\_model(daten, labels)
- c) LinearRegressionModel().train(daten, labels)
- d) regress(daten, labels)
- e) Das weiss ich noch nicht

**15. Wie kann in Pandas ein neuer DataFrame aus einer CSV-Datei erstellt werden?**

- a) pd.read\_csv('dateiname.csv')
- b) pd.open\_csv('dateiname.csv')
- c) pd.csv('dateiname.csv')
- d) pd.load\_csv('dateiname.csv')
- e) Das weiss ich noch nicht

**16. Wie führt man eine Korrelationsanalyse in Pandas durch?**

- a) df.correlate()
- b) df.corr()
- c) df.correlation()
- d) df.computeCorrelation()
- e) Das weiss ich noch nicht

**17. Welche Methode wird verwendet, um fehlende Werte in einem Pandas DataFrame zu finden?**

- a) isna()
- b) findNull()
- c) isNull()
- d) missingValues()
- e) Das weiss ich noch nicht

**18. Wie wird in Seaborn ein Streudiagramm erstellt?**

- ☐ a) `sns.scatterplot(data=df)`
- ☐ b) `sns.scatter(df)`
- ☐ c) `sns.plot.scatter(df)`
- ☐ d) `sns.drawScatter(df)`
- ☐ e) Das weiss ich noch nicht

**19. Welche Funktion in NumPy wird verwendet, um eine statistische Standardabweichung zu berechnen?**

- ☐ a) `np.std()`
- ☐ b) `np.variance()`
- ☐ c) `np.deviation()`
- ☐ d) `np.average()`
- ☐ e) Das weiss ich noch nicht

**20. Wie führt man eine Pivot-Tabelle in Pandas durch?**

- ☐ a) `df.pivot_table()`
- ☐ b) `df.pivot()`
- ☐ c) `df.makePivot()`
- ☐ d) `df.createPivot()`
- ☐ e) Das weiss ich noch nicht

# Test zur Ermittlung der Kenntnisse in technischem Python mit Schwerpunkt Netzwerktechnologie

## Fragen

**1. Welches Modul wird in Python für SSH-Verbindungen verwendet?**

- ☐ a) paramiko
- ☐ b) ssh
- ☐ c) telnetlib
- ☐ d) netmiko
- ☐ e) Das weiss ich noch nicht

**2. Wie erstellt man eine SSH-Verbindung in Python?**

- ☐ a) `ssh.connect(hostname)`
- ☐ b) `paramiko.SSHClient().connect(hostname)`
- ☐ c) `telnetlib.SSH(hostname)`
- ☐ d) `netmiko.connect(hostname)`
- ☐ e) Das weiss ich noch nicht

**3. Was ist ein häufig verwendetes Modul für Netzwerkanfragen in Python?**

- ☐ a) requests
- ☐ b) http
- ☐ c) urllib
- ☐ d) socket
- ☐ e) Das weiss ich noch nicht

**4. Wie führt man einen Befehl auf einem entfernten Server über SSH in Python aus?**

- ☐ a) `ssh.execute(command)`
- ☐ b) `paramiko.exec_command(command)`
- ☐ c) `ssh.run(command)`

- d) paramiko.SSHClient().exec\_command(command)
- e) Das weiss ich noch nicht

**5. Wie kann man in Python Dateien über SSH übertragen?**

- a) Verwendung von paramiko.SFTPClient
- b) Verwendung von ssh.file\_transfer
- c) Verwendung von scp
- d) Verwendung von ftplib
- e) Das weiss ich noch nicht

**6. Welche Bibliothek wird in Python für fortgeschrittene Netzwerkanalyse und Paketmanipulation verwendet?**

- a) socket
- b) requests
- c) urllib
- d) scapy
- e) Das weiss ich noch nicht

**7. Wie wird in Python ein DNS-Lookup durchgeführt?**

- a) socket.gethostbyname('hostname')
- b) dns.lookup('hostname')
- c) urllib.dns('hostname')
- d) requests.gethost('hostname')
- e) Das weiss ich noch nicht

**8. Wie wird in Python eine HTTP-GET-Anfrage gesendet?**

- a) http.get('url')
- b) requests.get('url')
- c) urllib.request('url')
- d) socket.request('GET', 'url')
- e) Das weiss ich noch nicht



**9. Wie kann man in Python den Inhalt einer Webseite herunterladen?**

- a) urllib.request('url')
- b) requests.get('url')
- c) http.get('url')
- d) socket.download('url')
- e) Das weiss ich noch nicht

**10. Wie wird in Python eine Netzwerkschnittstelle abgefragt?**

- a) Mit dem Modul netifaces
- b) Durch direkte Abfrage des Betriebssystems
- c) Mit dem socket-Modul
- d) Mit dem requests-Modul
- e) Das weiss ich noch nicht

**11. Wie wird in Python das Scannen von Netzwerkports durchgeführt?**

- a) Mit socket.port\_scan()
- b) Mit nmap-Modul
- c) Mit scapy-Modul
- d) Mit netmiko-Modul
- e) Das weiss ich noch nicht

**12. Welche Funktion wird in Python für SSL-verschlüsselte HTTP-Anfragen verwendet?**

- a) http.request('url', ssl=True)
- b) requests.get('url', verify=False)
- c) urllib.request('url', secure=True)
- d) socket.ssl('url')
- e) Das weiss ich noch nicht

**13. Wie wird in Python ein Netzwerk-Ping durchgeführt?**

- a) os.system('ping ' + hostname)
- b) socket.ping(hostname)
- c) requests.ping(hostname)

- d) `http.ping(hostname)`
- e) Das weiss ich noch nicht

**14. Wie überprüft man in Python, ob eine Netzwerkverbindung aktiv ist?**

- a) `socket.is_connected()`
- b) `netmiko.is_alive()`
- c) `paramiko.is_active()`
- d) `requests.is_online()`
- e) Das weiss ich noch nicht

**15. Welche Methode wird verwendet, um Netzwerkdaten in Python zu analysieren?**

- a) `scapy.analyze()`
- b) `nmap.analyze()`
- c) `socket.analyze()`
- d) `requests.analyze()`
- e) Das weiss ich noch nicht

**16. Wie wird in Python eine sichere Dateiübertragung mit SFTP durchgeführt?**

- a) Verwendung von `paramiko.SFTPClient`
- b) Verwendung von `scp`
- c) Verwendung von `ftplib`
- d) Verwendung von `socket`
- e) Das weiss ich noch nicht

**17. Wie kann man in Python einen HTTP-Header analysieren?**

- a) `requests.get('url').headers`
- b) `http.header('url')`
- c) `urllib.request('url').header()`
- d) `socket.get_header('url')`
- e) Das weiss ich noch nicht

**18. Wie wird in Python ein REST API-Aufruf durchgeführt?**

- a) `requests.post('api_url', data={...})`
- b) `http.request('api_url', method='POST', data={...})`
- c) `urllib.request('api_url', data={...})`
- d) `socket.request('api_url', method='POST', data={...})`
- e) Das weiss ich noch nicht

**19. Wie authentifiziert man sich in Python bei einer Netzwerkanfrage?**

- a) Durch Hinzufügen von Benutzername und Passwort in der URL
- b) Verwendung von API-Schlüsseln im Anfrageheader
- c) Versenden von Benutzername und Passwort über POST-Daten
- d) Verwendung einer OAuth-Bibliothek
- e) Das weiss ich noch nicht

**20. Wie wird in Python ein Websocket-Client erstellt?**

- a) Verwendung des websocket-Moduls
- b) Verwendung des socket-Moduls
- c) Verwendung des requests-Moduls
- d) Verwendung des http-Moduls
- e) Das weiss ich noch nicht