

- **!!! Beim Test können AUCH andere/weitere Fragen kommen !!!**
-
- Erklären Sie Instanz, Referenz, Referenzzähler, Garbage Collection und `del`.
- Was sind Type Hints?
- Erklären Sie den Unterschied zwischen Compiler- und Interpretersprachen.
- Unterschiede zwischen Java und Assembler.
- Unterschied `if-elif-else` und `match case`.
- Was sind Kontrollstrukturen? Welche eignet sich, um viele Möglichkeiten eines Inhalts zu überprüfen?
- Unterschied zwischen iterativer und rekursiver Programmierung.
- Nennen Sie drei fachliche Unterschiede zwischen Java und Python.
- Erklären Sie mutable und immutable, sowie deren Vor- und Nachteile.
- Erklären Sie `__name__` in Python.
- Unterschied zwischen Arrays in Python und Java.
- Prozesse Threads allg. in Betriebssystemen
- Unterschied zwischen `array` und `list` in Python sowie die Vorteile von Arrays.
- Nennen Sie drei Vorteile von Python gegenüber C-ähnlichen Sprachen.
- Geben Sie Beispiele für mutable und immutable Datentypen in Python, und wann welche sinnvoll sind.
- Drei Gemeinsamkeiten zwischen Java und Python.
- Erklären Sie die `pass`-Anweisung in Python.
- Unterschied im Umgang mit Datentypen zwischen Java und Python, einschließlich Type Hints.
- List Comprehension für quadratische Werte positiver Zahlen. List Comprehension für positive gerade Zahlen. Beispiel für eine Liste aus jedem zweiten Element, solange das Element nicht `None` ist.
- Floor division in python3
- garbage collection allgemein
- unterschied zw. Tuple, range und set
- unterschied zw. copy und deepcopy
- was sind dundervariablen und dunder-methoden
- f-string nach vorgabe erstellen
- was ist bei unittest gemeint mit „automatische“ test
- Vorteile von `unittest` gegenüber eigenen Tests.
- warum ist es ratsam unittest vor der codeentwicklung zu schreiben
- beispiel für ternären operator geben
- was ist das pdb, im welchen umfeld ist es sinnvoll einzusetzen
- Erläutern Sie den ternären Operator und geben Sie ein konventionelles Beispiel sowie die Umsetzung als ternären Operator an.
- exceptions anhand eines komplett-code-beispiels erklären, alle exception teile müssen vorkommen
- Arten von Exceptions generell erklären, die vier verschiedenen Exception-Kombis als Code demonstrieren bzw. erklären
- unterschied `def` und `lambda`

- Was sind Lambdafunktionen? Erzeugen Sie Listen aus Objekten durch Nutzung von Lambda für verschiedene Operationen.
- Sinn und Vorteile von Virtual Environments in Python. Wie passt da **uv** dazu?
- Nutzen von **map** und **Lambda**, um Listen von Objekten zu verarbeiten.
- Zwei Methoden zur Erstellung der **requirements.txt** in Virtual Envs und deren Inhalt.
- Beispiel einer Funktion mit nur Schlüsselwortpaaren als Parameter.
- In Kontext OOP in python, was bedeutet für Sie wenn sie das schlüsselwort **self** bei einer methode sehen
- In Kontext OOP in python, erklären sie **super()** und welche besonderheit **super()** in vergleich zu **super()** in Java hat
- Unterschied zwischen einer Klasse und einem Objekt
- Was ist **__init__** in Python?
- Unterschied zwischen einer Instanzvariable und einer Klassenvariable
- sichtbarkeitsmodifikatoren in python in vergleich zu java
- was kann man in python tun um zu "verstecken"
- **__str__** **__repr__**
- klassenattribute
- Was passiert, wenn man ein Attribut auf ein Objekt schreibt, das nicht in der Klasse definiert ist?
- Was ist **super()** und wofür wird es verwendet?
- **@staticmethod** **@classmethod** unterschiede
- was ist polymorphismus
- mehrfachvererbung
- Was ist ein MRO (Method Resolution Order)?
- Zwei Unterschiede in der objektorientierten Programmierung zwischen Java und Python.
- Python-Namespaces, mit Erklärung zu global und local.
- Demonstration der Python-Funktion **map**.
- Erklärung und Beispiel zur Python-Funktion **zip**.
- Bedeutung von **self** in Methoden bei Vererbungsstrukturen.
- Risiko beim Packaging und der Namensgebung in Python.
- Drei Varianten zur Importierung von Python-Paketen und ihre Unterschiede.
- Was ist ein Python-Modul, und welche Risiken bestehen beim Einsatz/import?
- Unterschiedliche Handhabung der Vererbung zwischen Python und Java.
- Python-Modul schreiben und die Import-Optionen nutzen.
- Klasse zur Ausgabe der Anzahl erzeugter Instanzen.
- Erklären Sie Python-Dekoratoren, **args** und **kwargs**.
- Hauptvorteil von verketteten Listen gegenüber echten Arrays.
- Beispiel für JSON-Dateien und relationale Datenbanken.
- Wann ist es sinnvoll, Methoden in anderen Methoden zu definieren?
- Klasse mit Funktionalität zur Zählung der Instanzen in Python.
- Python-Dekorator zur Validierung von URLs.
- Hauptvorteil doppelt verketteter Listen gegenüber einfach verketteten Listen.
- Vorteile und Nachteile des Iterable-Protokolls.
- Programm zur Demonstration der vier Namensräume in Python.