**LMS-Fragen zum Kurs Einführung in das Programmieren mit Python**

(Pro Lektion gibt es fünf Fragen mit je einer richtigen Antwort und drei falschen Antworten;

bitte geben Sie in der zweiten Spalte auch den Lernzyklus an, auf den sich die Frage bezieht.)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Lektion/****Frage-Nr.** | **Lern-zyklus** | **Frage** | **Richtige Antwort** | **Falsche Antwort** | **Falsche Antwort** | **Falsche Antwort** |
| 1/1 |  | Im Folgenden sind verschiedene mögliche Gründe für die Popularität von Python im Bereich Softwareentwicklung aufgeführt. Welcher davon trifft nicht zu?  | Python-Code wird durch den Compiler in eine ausführbare Datei umgewandelt. Das ist besonders vorteilhaft bei der Entwicklung von Webanwendungen. | Python-Code lässt sich relativ einfach lesen. | Die Python-Community zählt zahlreiche aktive Mitglieder, die Bibliotheken und Tools für die Softwareentwicklung mit Python erstellen. | Python-Programmcode ist sehr kompakt und präzise. Deshalb lassen sich hier mit weniger Zeilen mehr Features realisieren. |
| 1/2 |  | In welchen Spezialgebieten ist Python die Programmiersprache der Wahl für Fachleute und Forschungsteams?  | Data Science / Datenwissenschaften / Künstliche Intelligenz | Agrarwissenschaften / Biowissenschaften | Entwicklung von Betriebssystemen / Systemprogrammierung | Linguistik / Literaturwissenschaft  |
| 1/3 |  | Die wichtigste Funktion des Python-Interpreters besteht in der ... | ... Prüfung und Ausführung des eingelesenen Codes sowie der Übermittlung des Outputs. | ... Kompilierung von Programmcode und Erstellung einer ausführbaren Datei in Maschinensprache. | ... Unterstützung der simultanen Nutzung von Code-, Markup- und Textdokumenten. | ... Sicherstellung einer konsistenten Vergabe von Variablen- und Funktionsnamen. |
| 1/4 |  | Einer der größten Vorteile von Jupyter Notebook besteht darin, ... | ... dass Entwicklungsteams Programmcode, Mark-ups und unformatierten Text in einem Dokument kombinieren können. | ... dass Programmierfehler schon bei der Eingabe erkannt werden. | ... dass in der Umgebung kein Python-Interpreter für die Ausführung von Python-Code erforderlich ist. | ... dass der eingegebene Python-Code in C kompiliert wird und dadurch schneller läuft. |
| 1/5 |  | JupyterLab nutzt IPython als ... | ... Backend-Interpreter für Python-Code. | ... Tool zur Übersetzung von Mark-up-Formatierungen. | ... Verschlüsselungsmechanismus zum Schutz des Datenaustauschs zwischen Client und Server. | ... Compiler zur Erstellung ausführbarer Python-Dateien. |
| **Lektion/****Frage-Nr.** |  | **Frage** | **Richtige Antwort** | **Falsche Antwort** | **Falsche Antwort** | **Falsche Antwort** |
| 2/1 |  | Welche der hier angegebenen Zeichenfolgen stellt in Python eine Hexadezimalzahl dar? | 0x42 | Hex.42 | 42/16 | Hexadezimalzahlen lassen sich in Python nicht darstellen. |
| 2/2 |  | Wie lautet das Ergebnis der folgenden Rechenoperation?80 % 25 | 5 | 3 | 3.2 | ein Syntaxfehler |
| 2/3 |  | Wie können Sie Anführungszeichen in einen Python-String aufnehmen? | my\_string = "this has \"quotes\"." | my\_string = "this has """quotes"""." | my\_string = "this has \t"quotes\t"." | my\_string = "this has " + "quotes" + "." |
| 2/4 |  | Sie haben einen String mit der Bezeichnung *my\_string* und dem Wert „Python is so powerful!“ erstellt. Welche der folgenden Codezeilen liest aus dieser Variable lediglich den Teilstring „is so“ aus? | my\_string[7:12] | my\_string[7:] | my\_string[6:5] | my\_string[7:5] |
| 2/5 |  | Sie möchten eine Datei namens *thatfile.txt* öffnen, um dort Daten einzutragen. Falls die Datei noch nicht existiert, soll sie neu angelegt werden. Sollte jedoch schon eine derartige Datei bestehen, sollen die vorhandenen Inhalte bewahrt und alle neuen Einträge ganz am Ende eingefügt werden. Mit welcher Codezeile erreichen Sie dies? | open("thatfile.txt", "a") | open("thatfile.txt", "r") | open("thatfile.txt", "x")  | open("thatfile.txt", "c") |
| **Lektion/****Frage-Nr.** |  | **Frage** | **Richtige Antwort** | **Falsche Antwort** | **Falsche Antwort** | **Falsche Antwort** |
| 3/1 |  | Welchen Wert erhält die Variable c durch die folgende Anweisung?a=b=c=d=5 | 5 | d | nicht definiert | keinen Wert – es erscheint eine Fehlermeldung |
| 3/2 |  | Welchen Output erzeugt der folgende Code, wenn die Benutzer:innen in der Eingabeaufforderung die Zahl null eingeben?my\_num = int(input())if my\_num < 0:print(1)elif my\_num > 0:print(2)else:print(3) | 3 | 0 | 1 | 2 |
| 3/3 |  | Weshalb kann der folgende Code nicht ausgeführt werden?my\_num = 0int(input())if my\_num < 0print(1)elif my\_num > 0print(2)elseprint(3) | Es fehlen die Doppelpunkte am Ende der if-, elif- und else-Anweisungen. | Jede einzelne Zeile des obigen Codes ist ungültig. | Ein Wort ist falsch geschrieben. | Die per input()-Befehl eingegebene Zahl wird nicht auf dem Bildschirm ausgegeben. |
| 3/4 |  | Welcher der folgenden Befehle erzeugt eine Zahlenreihe, die bei 11 beginnt und bei 99 endet? | range(11,100) | range(11,99) | range(10,99) | range(10,100) |
| 3/5 |  | Sie verfügen über eine Liste, die verschiedene Werte umfasst und die Bezeichnung *my\_list* trägt. Mit welcher Codezeile können Sie alle Elemente der Liste nacheinander abrufen, ohne dass bei jeder Modifikation von *my\_list* Änderungen an Ihrem Programm vorgenommen werden müssen? | def my\_func(a, b=0, c=”Yes”): | my\_list.loop | while len(listy)>0: | def my\_func(a,b(0),c(“Yes): |
| **Lektion/****Frage-Nr.** |  | **Frage** | **Richtige Antwort** | **Falsche Antwort** | **Falsche Antwort** | **Falsche Antwort** |
| 4/1 |  | In welcher Hinsicht sind Funktionen nützlich? | Sie ermöglichen die Wiederverwendung vorhandener Codeblöcke. | Sie eliminieren alle Fehler im Code. | Programme mit Funktionen arbeiten genauer als Programme ohne Funktionen. | Funktionen erhöhen die Lesbarkeit des Codes. |
| 4/2 |  | Was ist unter dem Begriff Sichtbarkeit zu verstehen? | Der Ausdruck Sichtbarkeit bezeichnet den Geltungs- bzw. Gültigkeitsbereich von Variablen und Funktionen, beginnend an der Stelle im Programm, an der sie definiert werden. | Die Sichtbarkeit bestimmt die Zahl der Funktionen und Argumente, die im Rahmen eines Programms definiert werden können. | Die Sichtbarkeit einer Variablen umfasst alle Datentypen, die in dieser Variablen abgelegt werden können. | Der Ausdruck Sichtbarkeit bezeichnet Rückgabewerte von Funktionen. |
| 4/3 |  | Warum tritt bei der Ausführung der folgenden Codezeilen ein Fehler auf?a=int(input())b=int(input())if c < a + b:print(“Less than a + b”)a = b = c = 1print(a,b,c) | Die Variable c ist in Zeile 3 noch nicht sichtbar bzw. definiert. | Die Variable b ist in Zeile 3 noch nicht sichtbar bzw. definiert. | Die Variable a ist in Zeile 3 noch nicht sichtbar bzw. definiert. | Die letzte Codezeile ist fehlerhaft. |
| 4/4 |  | Bei welcher der folgenden Codezeilen handelt es sich um die Definition einer Funktion namens *my\_func* mit drei Parametern (a, b und c) und den Standardargumenten *b=0* und *c="Yes"*? | def my\_func(a, b=0, c=”Yes”): | my\_func(a, b, c): | def my\_func(a, b, c): | def my\_func(a, b(0), c(“Yes”): |
| 4/5 |  | Bei welcher der folgenden Codezeilen handelt es sich nicht um einen gültigen Aufruf der Funktion *my\_func* mit drei Parametern (a, b und c) und den Standardargumenten *b=0* und *c="Yes"*?  | my\_func( ) | my\_func(1) | my\_func(1, c=”No”) | my\_func(1, 2) |
| **Lektion/****Frage-Nr.** |  | **Frage** | **Richtige Antwort** | **Falsche Antwort** | **Falsche Antwort** | **Falsche Antwort** |
| 5/1 |  | Was ist der Unterschied zwischen einem Syntaxfehler und einer Ausnahme? | Ausnahmen sind Laufzeitfehler, die einen Programmabbruch verursachen. Im Gegensatz dazu tritt ein Syntaxfehler auf, wenn der Programmcode die Syntaxregeln verletzt. | Syntaxfehler sind Laufzeitfehler, die einen Programmabbruch verursachen. Im Gegensatz dazu tritt eine Ausnahme auf, wenn der Programmcodes die Syntaxregeln verletzt. | Syntaxfehler verursachen einen Programmabsturz. Ausnahmen ziehen eine Fehlermeldung des Interpreters nach sich. | Es gibt keinen Unterschied zwischen den beiden Begriffen. |
| 5/2 |  | Welche der folgenden Maßnahmen ist am besten zur Diagnose und Behebung eines Syntaxfehlers geeignet? | Zunächst muss das in der Fehlermeldung des Interpreters bezeichnete Programmsegment lokalisiert werden. Dann wird dessen Funktionsweise sowie die Funktionsweise des es umgebenden Codes analysiert. Auf dieser Grundlage lässt sich das aufgetretene Problem diagnostizieren und beheben. | Der in der Fehlermeldung des Interpreters bezeichnete Codeabschnitt muss lokalisiert und geändert werden. | Die in der Fehlermeldung des Interpreters bezeichnete Codezeile muss lokalisiert und geändert werden. | Der in der Fehlermeldung des Interpreters bezeichnete Codeabschnitt muss lokalisiert und gelöscht werden. |
| 5/3 |  | Welche Folgen hat es, wenn Sie versuchen, mit der open-Funktion eine nicht existierende Datei zu öffnen? | Es wird die Ausnahme *FileNotFoundError* ausgegeben.  | Es wird der Syntaxfehler *FileNotFoundError* ausgegeben. | Es wird der Syntaxfehler *IncorrectFilePathError* ausgegeben. | Es wird die Ausnahme *IncorrectFilePathError* ausgegeben. |
| 5/4 |  | Sie wissen, dass in einem von Ihnen programmierten Codeabschnitt die Ausnahme *ZeroDivisionError* auftreten kann, und möchten in diesem Fall einen Absturz des Programms durch die Implementierung geeigneter Maßnahmen zur Ausnahmebehandlung verhindern. Außerdem möchten Sie sicherstellen, dass bestimmte Bereinigungsanweisungen in jedem Fall ausgeführt werden – also unabhängig davon, ob eine Ausnahme ausgelöst wird oder nicht. Wie können Sie dies bewerkstelligen? | durch die Erstellung eines try-Blocks, gefolgt von einer auf den ZeroDivisionError zugeschnittenen except-Klausel und einer finally-Klausel  | durch die Implementierung eines try-Blocks und einer generischen except-Klausel ohne weitere Komponenten zur Ausnahmebehandlung | durch die Hinzufügung eines einfachen try-Blocks ohne weitere Komponenten zur Ausnahmebehandlung | durch die Erstellung eines try-Blocks und einer auf den ZeroDivisionError zugeschnittenen except-Klausel ohne weitere Komponenten zur Ausnahmebehandlung |
| 5/5 |  | Mit welcher der folgenden Codezeilen können Sie die Dringlichkeitsschwelle der Protokollierung auf die Stufe *CRITICAL* setzen? | logging.basicConfig(level = logging. CRITICAL) | logging.basicConfig = logging. CRITICAL | logging.basicConfig.level = logging. CRITICAL | logging. level = logging.CRITICAL |
| **Lektion/****Frage-Nr.** |  | **Frage** | **Richtige Antwort** | **Falsche Antwort** | **Falsche Antwort** | **Falsche Antwort** |
| 6/1 |  | Mit welcher der folgenden Codezeilen wird die Funktion *add\_numbers* aus der Bibliothek *adding\_library* importiert? | from adding\_library import add\_numbers | import adding\_library | import add\_numers from adding\_library | import adding\_library.adding\_numbers |
| 6/2 |  | In Ihrem Programm gibt es eine Funktion namens *my\_function*, in der eine Variable mit der Bezeichnung *my\_var* erstellt wird. Welchem Namensraum ist *my\_var* zuzuordnen? | lokaler Namensraum | globaler Namensraum | Built-in-Namensraum | Funktionsnamensraum |
| 6/3 |  | Die int()-Funktion von Python wandelt ein angegebenes Argument in eine Ganzzahl um. Welchem Namensraum ist int() zuzuordnen? | Built-in-Namensraum | lokaler Namensraum | Funktionsnamensraum | globaler Namensraum |
| 6/4 |  | Welche der folgenden Aussagen beschreibt den Unterschied zwischen einem Python-Kommentar und einem Python-Docstring am besten? | Kommentare haben im Programm keine funktionelle Bedeutung, während Docstrings als Hilfetexte in die Dokumentation aufgenommen werden. | Kommentare und Docstrings sind syntaktisch voneinander unterschieden, dienen jedoch dem gleichen Zweck. | Docstrings haben im Programm keine funktionelle Bedeutung, während Kommentare als Hilfetexte in die Dokumentation aufgenommen werden. | Kommentare werden beim Kompilieren in die ausführbare Datei aufgenommen, Docstrings dagegen nicht. |
| 6/5 |  | Bei welcher der nachfolgend aufgeführten Antworten handelt es sich nicht um eine Python-Bibliothek für datenwissenschaftliche Anwendungen? | DataPyScience | NumPy | SciPy | Pandas |
| **Lektion/****Frage-Nr.** |  | **Frage** | **Richtige Antwort** | **Falsche Antwort** | **Falsche Antwort** | **Falsche Antwort** |
| 7/1 |  |  |  |  |  |  |
| 7/2 |  |  |  |  |  |  |
| 7/3 |  |  |  |  |  |  |
| 7/4 |  |  |  |  |  |  |
| 7/5 |  |  |  |  |  |  |
| **Lektion/****Frage-Nr.** |  | **Frage** | **Richtige Antwort** | **Falsche Antwort** | **Falsche Antwort** | **Falsche Antwort** |
| 8/1 |  |  |  |  |  |  |
| 8/2 |  |  |  |  |  |  |
| 8/3 |  |  |  |  |  |  |
| 8/4 |  |  |  |  |  |  |
| 8/5 |  |  |  |  |  |  |
| **Lektion/****Frage-Nr.** |  | **Frage** | **Richtige Antwort** | **Falsche Antwort** | **Falsche Antwort** | **Falsche Antwort** |
| 9/1 |  |  |  |  |  |  |
| 9/2 |  |  |  |  |  |  |
| 9/3 |  |  |  |  |  |  |
| 9/4 |  |  |  |  |  |  |
| 9/5 |  |  |  |  |  |  |
| **Lektion/****Frage-Nr.** |  | **Frage** | **Richtige Antwort** | **Falsche Antwort** | **Falsche Antwort** | **Falsche Antwort** |
| 10/1 |  |  |  |  |  |  |
| 10/2 |  |  |  |  |  |  |
| 10/3 |  |  |  |  |  |  |
| 10/4 |  |  |  |  |  |  |
| 10/5 |  |  |  |  |  |  |
|  | Lern-zyklus |  |  |  |  |  |