Kompetenzübersicht

Kompetenzband:	HZ	Grundlagen	Fortgeschritten	Erweitert
Probleme erfassen und Lösungsansätze entwickeln.	1	A1G: Ich kenne die Eigenschaften von gut formulierten Anforderungen • LU03d - Arbeitspaket	A1F: Ich kann Anforderungen erfassen und erläutern.	A1E: Ich kann einfache Anforderungen formulieren.
		A2G: Ich kann eine strukturierte Vorgehensweise für das Entwickeln einer Lösung aufzeigen	A2F: Ich kann eine strukturierte Vorgehensweise für das Entwickeln einer Lösung anwenden.	AE2: Ich kann Vorgehensweise kritisch hinterfragen und Verbesserungsvorschläge nennen.
Daten, Datentypen und Variablen ableiten und einsetzen	2,4	B1G: Ich kann die Unterschiede von Datentypen erklären. (Bsp: Ganzzahlen, Gleitkommazahlen, Zeichen usw.) • LU01a - Was ist Blockly • LU01b - Bedienung von Blockly • LU01c - Variable und konstante Werte • LU02a - Ausgaben • LU02a - Variable und konstante Werte • LU02b - Benutzereingaben lesen • LU02c - Einlesen verschiedener Datentypen	B1F: Ich kann den richtigen Datentyp für eine Variable aufgrund der Aufgabenstellung wählen. • LU01b - Bedienung von Blockly • LU02.A02: Different Inputs • LU02a - Ausgaben • LU02b - Benutzereingaben lesen • LU02c - Einlesen verschiedener Datentypen • LU02d - Zeichenketten • LU02e - Rechnen mit Zahlen	B1E: Ich kann komplexe Datentypen (zBsp String, Integer, Double...) aufgrund der Aufgabenstellung wählen. In Python nicht relevant
			B2F: Ich kann den Unterschied zwischen primitiven und komplexen Datentypen erklären. In Python nicht relevant	
		B3G: Ich kann den Zweck von Variablen in einem Programm erläutern. • LU02a - Variable und konstante Werte	B3F: Ich kann Variablen deklarieren, initialisieren und Zuweisungen vornehmen. • LU02c - Variablen in Python	B3E: Ich kann Variablen als Konstanten deklarieren und initialisieren. • LU02a - Variable und konstante Werte

Kompetenzband:	HZ	Grundlagen	Fortgeschritten	Erweitert
		B4G: Ich kann die Möglichkeit der Verwendung von Variablen abhängig von Ihrem Datentyp erläutern.	B4F: Ich kann die Variablen abhängig von Ihrem Datentyp anwenden. (ZBps für arithmetische Operation) • LU02e - Rechnen mit Zahlen	B4E: Ich kann Variablen mit einem bestimmten Datentyp auf einen anderen umwandlen.(zBps: toString) • LU02d - Einlesen verschiedener Datentypen
Zusammengesetze Datentypen einsetzen	2	C1G: Ich kann den Zweck von zusammengesetzten Datentypen erläutern. (ZBsp eindimensionaler Array) • LU01c - Variable und konstante Werte • LU01d - Variablen deklarieren • LU02a - Ausgaben • LU02b - Benutzereingaben lesen • LU02c - Einlesen verschiedener Datentypen • LU02d - Zeichenketten • LU02e - Rechnen mit Zahlen • LU05a - Collections - Grundlagen • LU05b - Listen • LU11.A02 - Zahlzeichen • LU11a - Dictionary • LU11b - Verarbeitungen mit Dictionary • LU14.A01: Collections • LU14.A02: Längenumrechner • LU14.L02: Längenumrechner • LU14a - Listen mit Python • LU14b - Dictionary	C1F: Ich kann zusammengesetzte Datentypen deklarieren, initialisieren und Zuweisungen vornehmen. (ZBsp eindimensionaler Array) • LU01d - Variablen deklarieren • LU02c - Einlesen verschiedener Datentypen • LU02e - Rechnen mit Zahlen • LU05b - Listen • LU11.A02 - Zahlzeichen • LU11a - Dictionary • LU11b - Verarbeitungen mit Dictionary • LU12a - Eigene Datentypen • LU12a - Eigene Datentypen • LU12b - Klassen definieren • LU12b - Klassen definieren • LU14.A01: Collections • LU14.A02: Längenumrechner • LU14.L02: Längenumrechner • LU14a - Listen mit Python • LU14b - Dictionary	C1E: Ich kann Methoden der zusammengesetzten Datentypen anwenden. • LU05b - Listen • LU06a - Collections - Grundlagen • LU06b - Listen erstellen und befüllen • LU06c - Listen bearbeiten • LU11.A02 - Zahlzeichen • LU11a - Dictionary • LU11b - Verarbeitungen mit Dictionary • LU12a - Eigene Datentypen • LU12a - Eigene Datentypen • LU12b - Klassen definieren • LU12b - Klassen definieren • LU14.A01: Collections • LU14.A02: Längenumrechner • LU14.L02: Längenumrechner • LU14a - Listen mit Python • LU14b - Dictionary

Kompetenzband:	HZ	Grundlagen	Fortgeschritten	Erweitert
Anforderungen visuell darstellen	3	D1G: Ich kenne die Einsatzgebiete der grafischen Beschreibung eines Ablaufes. (zBsp mit Activity-Diagramm, Sequenz-Diagramm..) • LU03a - Was ist Blockly • LU03b - Bedienung von Blockly • LU03c - PyCharm - Projekt erstellen	D1F: Ich kann einen vorgegebenen Programmablauf grafisch darstellen. (zBsp Activity-Diagramm, Sequenz-Diagramm..) • LU03b - Bedienung von Blockly • LU04.A01: Briefanrede • LU04.A02: Notenschnitt • LU04.A03: Münzenspiel • LU04.A04: U = R * I • LU04.A05: Grösster gemeinsamer Teiler • LU04.A06: Alternativer Notenschnitt • LU04.A07: Münzenspiel gegen Computer • LU04a - Die drei Kontrollstrukturen • LU04b - Sequenz • LU04c - Selektion • LU04d - Iteration • LU04e - Verknüpfte Bedingungen	D1E: Ich kann beschriebenen Ablauf in einen Programmablauf überführen und grafisch darstellen.(zBsp Activity-Diagramm, Sequenz-Diagramm..)
				D2E: Ich kann Bedingungen mit mehr als zwei Teilbedingungen formulieren. • LU04e - Verknüpfte Bedingungen

Kompetenzband:	HZ	Grundlagen	Fortgeschritten	Erweitert
Entwicklungsumgebung einsetzen	4,6	E1G: Ich kann den Zweck einer Entwicklungsumgebung (IDE) erklären. • LU01b - PyCharm, Visual Studio Code, Spyder: Die Entwicklungsumgebung • LU05a - Kontrollstrukturen • LU05b - Selektion • LU05c - Iteration • LU05d - Verknüpfte Bedingungen • LU07a - Erweiterung der if-Anweisung • LU07b - While-Else-Anweisung in Python • LU07c - For-Schleife	E1F: Ich kann die Entwicklungsumgebung effizient einsetzen • LU04c - Arbeiten mit GitHub Classroom • LU04c - Working with GitHub Classroom • LU05c - Iteration • LU05d - Verknüpfte Bedingungen • LU07a - Erweiterung der if-Anweisung • LU07b - While-Else-Anweisung in Python • LU07c - For-Schleife • LU08c - Arbeiten mit GitHub Classroom	E1E: Ich kann die Shortcuts und Funktionen der Entwicklungsumgebung effizient einsetzen
		E2G: Ich kann erläutern für was das Compilieren dient.	E2F: Ich kann vom Compiler angezeigte Fehler- und Warnmeldung interpretieren. • LU08a - Fehlersuche Grundlagen • LU08b - Syntaxfehler suchen • LU08c - Logikfehler suchen • LU08e - Repetition Syntaxfehler und Logikfehler • LU10a - Fehlersuche Grundlagen • LU10b - Syntaxfehler suchen • LU10c - Logikfehler suchen	E2E: Ich kann die Ursachen der vom Compiler angezeigten Fehler- und Warnmeldungen beheben. • LU08a - Fehlersuche Grundlagen • LU08b - Syntaxfehler suchen • LU08c - Logikfehler suchen • LU08e - Repetition Syntaxfehler und Logikfehler • LU10a - Fehlersuche Grundlagen • LU10b - Syntaxfehler suchen • LU10c - Logikfehler suchen
		E3G: Ich kann erläutern für was ein Debugger dient. • LU08d - Debugger anwenden • LU10d - Debugger anwenden	E3F: Ich kann einen Debugger zur Programmausführung anwenden. • LU08d - Debugger anwenden • LU10d - Debugger anwenden	E3E: Ich kann einen Debugger für die Fehleranalyse einsetzen. • LU08d - Debugger anwenden • LU10d - Debugger anwenden

Kompetenzband:	HZ	Grundlagen	Fortgeschritten	Erweitert
Applikation implementieren	3,4	F1G: Ich kann Aufbau, Syntax und Struktur eines einfachen Programmes erklären. • LU04a - Die drei Kontrollstrukturen • LU04b - Sequenz • LU04c - Selektion • LU04d - Iteration • LU07a - Erweiterung der if-Anweisung • LU07b - While-Else-Anweisung in Python	F1F: Ich kann einen detailliert vorgegeben Ablauf mit einer Programmiersprache umsetzen. • LU06.A01: Positivity • LU06.A02: Larger Than or Equal To • LU06.A03: Grades and Points • LU06.A04: Odd or Even • LU06.A05: Checking the age • LU06.A06: Leap year • LU06.A08: Carry on? • LU06.A09: Only positives • LU06.A10: Counting • LU06.A11: Factorial • LU06.A12: Split in pieces • LU06.A13: Bank loan • LU09.A01 - Einfache Funktionen definieren • LU09.A02 - Inhalte in Funktionen auslagern • LU09.A03 - Erweiterte Aufgaben	F1E: Ich kann einen grob beschriebenen Ablauf detaillieren und mit einer Programmiersprache umsetzen. • LU06.A06: Leap year • LU06.A08: Carry on? • LU06.A09: Only positives • LU06.A10: Counting • LU06.A11: Factorial • LU06.A12: Split in pieces • LU06.A13: Bank loan • LU09.A01 - Einfache Funktionen definieren • LU09.A02 - Inhalte in Funktionen auslagern • LU09.A03 - Erweiterte Aufgaben
		F2G: Ich kann erklären wieso die Aufteilung eines Programmes in verschiedene Methoden sinnvoll ist. • LU09.A01 - Einfache Funktionen definieren • LU09.A02 - Inhalte in Funktionen auslagern • LU09.A03 - Erweiterte Aufgaben • LU09a - Funktionen • LU09a - Funktionen • LU09b - Ergänzedes Wissen • LU09b - Funktionen erweitert	F2F: Ich kenne den Aufbau und den Aufruf einer Methode (Deklaration und Implementation) und kann diese korrekt einsetzen. (zBsp Instanzvariablen, Parametern, lokalen Variablen, Return Werte) • LU09.A01 - Einfache Funktionen definieren • LU09.A02 - Inhalte in Funktionen auslagern • LU09.A03 - Erweiterte Aufgaben • LU09a - Funktionen • LU09a - Funktionen • LU09b - Ergänzedes Wissen • LU09b - Funktionen erweitert	F2E: Ich kann Instanzvariablen, Parametern, lokalen Variablen und Return Werte gezielt einsetzen. • LU09.A01 - Einfache Funktionen definieren • LU09.A02 - Inhalte in Funktionen auslagern • LU09.A03 - Erweiterte Aufgaben • LU09a - Funktionen • LU09a - Funktionen • LU09b - Ergänzedes Wissen • LU09b - Funktionen erweitert

Kompetenzband:	HZ	Grundlagen	Fortgeschritten	Erweitert
		F3G: Ich kann Selektionen und Iterationen (Kopf und Fussgesteuert) mit Bedingungen codieren. • LU04.A01: Briefanrede • LU04.A02: Notenschnitt • LU04.A03: Münzenspiel • LU04.A04: U = R * I • LU04.A05: Grösster gemeinsamer Teiler • LU04.A06: Alternativer Notenschnitt • LU04.A07: Münzenspiel gegen Computer • LU04c - Selektion • LU04d - Iteration • LU04e - Verknüpfte Bedingungen • LU06.A01: Positivity • LU06.A02: Larger Than or Equal To • LU06.A03: Grades and Points • LU06.A04: Odd or Even • LU06.A05: Checking the age • LU06.A06: Leap year • LU06.A08: Carry on? • LU06.A09: Only positives • LU06.A10: Counting • LU06.A11: Factorial • LU06.A12: Split in pieces • LU06.A13: Bank loan • LU06a - Selektion • LU06b - Iteration • LU06c - For-Schleife • LU07a - Erweiterung der if-Anweisung • LU07b - While-Else-Anweisung in Python • LU09.A01 - Einfache Funktionen definieren • LU09.A02 - Inhalte in Funktionen auslagern • LU09.A03 - Erweiterte Aufgaben	F3F: Ich kann zwei Teilbedingungen mit AND oder OR verknüpfen. • LU04e - Verknüpfte Bedingungen • LU06.A01: Positivity • LU06.A02: Larger Than or Equal To • LU06.A03: Grades and Points • LU06.A04: Odd or Even • LU06.A05: Checking the age • LU06.A06: Leap year • LU06.A08: Carry on? • LU06.A09: Only positives • LU06.A10: Counting • LU06.A11: Factorial • LU06.A12: Split in pieces • LU06.A13: Bank loan • LU06a - Selektion • LU09.A01 - Einfache Funktionen definieren • LU09.A02 - Inhalte in Funktionen auslagern • LU09.A03 - Erweiterte Aufgaben	F3E: Ich kann verschachtelte Kontrollstrukturen einsetzen. • LU06.A01: Positivity • LU06.A02: Larger Than or Equal To • LU06.A03: Grades and Points • LU06.A04: Odd or Even • LU06.A05: Checking the age • LU06.A06: Leap year • LU06.A08: Carry on? • LU06.A09: Only positives • LU06.A10: Counting • LU06.A11: Factorial • LU06.A12: Split in pieces • LU06.A13: Bank loan • LU06a - Selektion • LU06b - Iteration • LU06c - For-Schleife • LU09.A01 - Einfache Funktionen definieren • LU09.A02 - Inhalte in Funktionen auslagern • LU09.A03 - Erweiterte Aufgaben

Kompetenzband:	HZ	Grundlagen	Fortgeschritten	Erweitert
Konventionen einhalten	5	G1G: Ich kann mein Programm mit ein- und mehrzeiligen Kommentaren ergänzen. • LU02c - Variablen in Python • LU12b - Klassen definieren	G1F: Ich kenne Möglichkeiten Kommentare zu Formatieren oder zu Annotieren (zBsp 🔧Fix Me!, TODO etc.) • LU14.A01: Collections	G1E: Ich setze Konventionen ein. (zBsp Clean Code, Coding Guidelines..) • BZZ Codingstandards für Python • LU02c - Variablen in Python • LU12b - Klassen definieren

From:
https://wiki.bzz.ch/ - **BZZ - Modulwiki**

Permanent link:
https://wiki.bzz.ch/modul/archiv/m319python/kompetenzuebersicht

Last update: **2024/03/28 14:07**