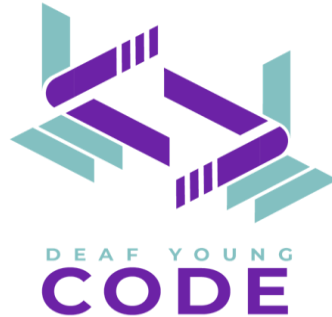




2023-2-IT03-KA220-YOU-000179130

Arbeitsblätter für Modul 3

Einführung – Programmierung

Verwendung als Einzelarbeit oder Teamarbeit



Co-funded by
the European Union

Finanziert von der Europäischen Union. Die geäußerten Ansichten und Meinungen sind jedoch ausschließlich die der Autor(en) und spiegeln nicht unbedingt wider jene der Europäischen Union oder der Europäischen Bildungs- und Kulturexekutivagentur (EACEA). Weder die Europäische Union noch die EACEA können dafür verantwortlich gemacht werden.

Arbeitsblatt – Einführung in die Programmierung

Modul: Einführung in die Programmierung

Dauer: 90 Minuten

Level: Anfänger*innen

Ziel/Zielgruppe: Gehörlose Jugendliche, Erwerb digitaler Basiskompetenzen

Warm-up

Quick-Check (Plenum, Gruppe)

Fragen:

1. Codieren = Anweisungen schreiben, Codieren ist ein Teil der Programmierung.

Programmieren ist mehr: Es bezeichnet den gesamten Prozess der Erstellung eines digitalen Produkts.

ja ☐

nein ☐

2. Codieren und Programmieren haben dieselbe Bedeutung. Es gibt keinen Unterschied.

ja ☐

nein ☐

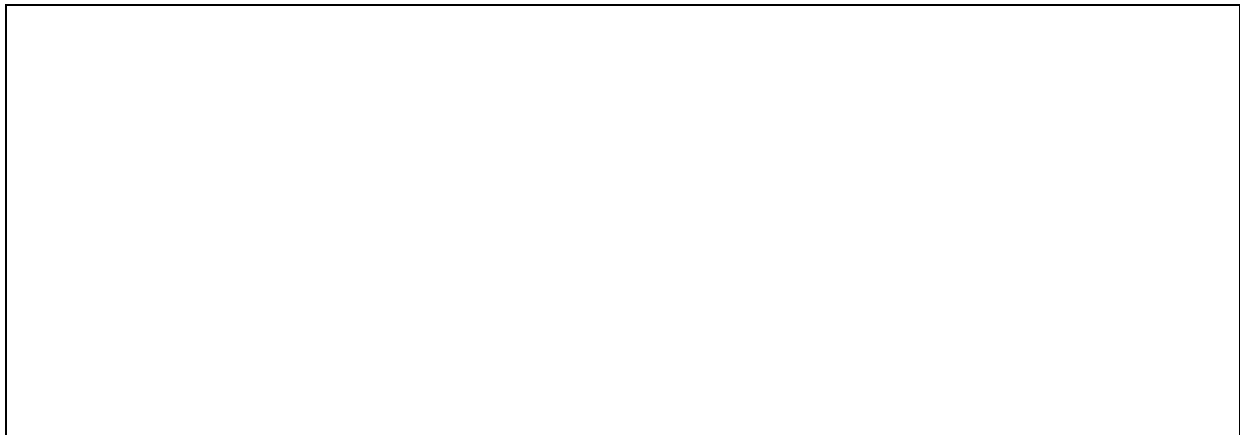
Teil 1 – Wichtige Beispiele aus dem Alltag

Brainstorming:

Dein Alltag: Denke an alle Anwendungen, die du nutzt und die programmiert werden müssen.

Beispiele: Taschenrechner, Ampeln, Wecker mit Licht und Schlummerfunktion.

- 1) Schreiben oder zeichne mindestens 5 weitere Anwendungen auf.



- 2) Gruppendiskussion

Präsentiere deine Beispiele und diskutiere mit den anderen.

Teil 2 – Schlüsselemente – Grundbegriffe

Diese 5 Elemente sind sehr wichtig:

- 2.1 Algorithmen
- 2.2 Datenstrukturen, Variablen und Datentypen
- 2.3 Kontrollstrukturen, Kontrollfluss
- 2.4 Grundlegende Operatoren und Ausdrücke
- 2.5 Debugging

2.1 Algorithmen

= Schritt-für-Schritt-Anleitungen zur Lösung von Problemen, ähnlich wie ein Kochrezept.

Übung: Ordne deinen Kühlschrank

Entwickle einen Ordnungs-Algorithmus für den Kühlschrank mit der Regel „Wenn abgelaufen, wegwerfen“. **Wichtig: Produkte mit rotem Stern sind abgelaufen.**



Los geht's:

Milch > Packung 1 > an der Seitentür des Kühlschranks, unteres Fach

[illegible]

2.2 Datenstrukturen, Variablen und Datei-Typen

Methoden zum Organisieren und Speichern von Daten, wie Listen oder Wörterbücher, Beispiel: Zahlen, Text

2.2 Datenstrukturen, Variablen und Datei-Typen

Methoden zum Organisieren und Speichern von Daten, wie Listen oder Wörterbücher, Beispiel: Zahlen, Text

Beispiel: Zahlen, Text

Übung: Apfelkuchen-Rezept

160 g Butter – 150 g Zucker – 1 Päckchen Vanillezucker – 3 Eier – 270 g Mehl – 3 TL

Backpulver – 1 Prise Salz – 4 Äpfel – 4 EL Milch – 50 g Mandeln

Jeder Wert muss zugewiesen werden.

[illegible]

2.3 Kontrollstrukturen, Kontrollfluss, if-else-Fluss (WENN > DANN)

if-else: Lassen Sie das Programm Entscheidungen treffen.

„if“ eine Bedingung ist wahr > eine Sache tun.

„else“ > etwas anderes tun.

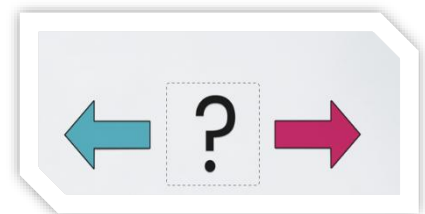
Beispiel: Brauche ich heute eine Jacke?

Programmablauf erstellen: „Wenn es kalt ist, ziehe ich eine Jacke an, sonst nicht.“

Bedingung: „Ist es kalt?“

Wenn die Antwort „Ja“ lautet → Jacke anziehen.

Wenn die Antwort „Nein“ lautet → keine Jacke anziehen.



Beispiel: Schreibe eine if-else-Schleife für die Funktion einer Ampel

.....

.....

.....

.....

.....

.....



2.4 Operatoren und Ausdrücke

Wiederverwendbare Codeabschnitte, die bestimmte Aufgaben ausführen.

Operators and expressions can be explained well with 'calculating and comparing', and 'reusable sections' such as small helper functions.

You will be familiar with some operators from your maths lessons.

Examples:

+ (plus): Adds two numbers, e.g. $5 + 3$ equals 8.

- (minus): Subtracts, e.g. $10 - 4$ equals 6.

* (Multiply): Multiplies, e.g. $4 * 3$ equals 12.

/ (Divide): Divides, e.g. $12 / 3$ equals 4.

Operatoren und Ausdrücke lassen sich gut mit „**Berechnen und Vergleichen**“ erklären.

Einige Operatoren sind dir bereits aus dem Mathematikunterricht bekannt.

Beispiele:

+ (Plus): addiert zwei Zahlen, z. B. $5 + 3$ ergibt 8.

- (Minus): subtrahiert, z. B. $10 - 4$ ergibt 6.

* (Multiplikation): multipliziert, z. B. $4 * 3$ ergibt 12.

/ (Division): dividiert, z. B. $12 / 3$ ergibt 4.

Übung:

Du möchtest berechnen, wieviele Stunden du insgesamt im Internet verbringst.

Berechnen Sie dies für eine Woche.

1) Welche Operatoren hast du verwendet?

2) Was ist dein Ergebnis?

.....

.....

2.5 Debuggig

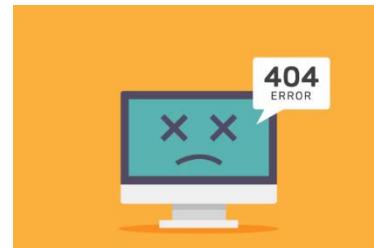
bug = Fehler

Debugging bedeutet,

Fehler in einem Programm zu finden und zu beheben.

Übung: Zeichen-Spiel

Zweier-Arbeit: teste – überprüfe - teste



Ein Teilnehmer ist der „**Programmierer**“, der andere ist der „**Roboter**“.

Der Programmierer gibt sehr genaue Schritt-für-Schritt-Anweisungen, wie ein Bild zu zeichnen ist.

Wenn etwas schief geht oder komisch aussieht, dann gebärdet der Roboter: „Es gibt einen Fehler in unserem Programm.“

Dann muss der Programmierer die Anweisung ändern, damit der Roboter die Aufgabe korrekt ausführen kann.

Beispiele für die Zeichnungen:



Teil 3 —Quiz

Markiere die richtige Antwort:

Was ist ein Programm?

- A) Eine Art von Hardware, die in Computern verwendet wird.
- B) Ein Spiel, das auf einem Computer gespielt wird.
- C) Eine Reihe von Anweisungen, die ein Computer befolgt, um eine Aufgabe auszuführen.
- D) Ein Gerät, das Daten speichert.

2. Welches der folgenden Beispiele ist eine Variable in der Programmierung?

- A) Ein Befehl zum Drucken von Text auf dem Bildschirm.
- B) Ein Speicherort, an dem Daten wie Zahlen oder Text gespeichert werden.
- C) Eine Meldung, die erklärt, was der Code tut.
- D) Eine Art von Programmiersprache.

3. Was macht eine Schleife in einem Programm?

- A) Sie prüft, ob eine Bedingung wahr oder falsch ist.
- B) Sie wiederholt eine Reihe von Anweisungen mehrmals.
- C) Sie speichert Daten für die spätere Verwendung.
- D) Sie sendet Daten an einen anderen Computer.

4. Warum ist Debugging beim Programmieren wichtig?

- A) Um dem Programm neue Funktionen hinzuzufügen.
- B) Um Fehler oder Bugs im Code zu identifizieren und zu beheben.
- C) Um das Programm schneller laufen zu lassen.
- D) Um Code in verschiedene Sprachen zu übersetzen.

Teil 4 — Reflektion, Feedback

- Was hast du heute gelernt?
- Was war für dich schwierig?
- Was möchtest du noch mehr üben?