

MODULPLAN UND INHALT

Modul 5

Scratch & Flutter



MODULPLAN

Modulplan	
Modultitel	Scratch & Flutter
Dauer	3 Stunden (gesamt)
Lernziele	Ziele: • die Grundlagen des blockbasierten Programmierens mit Scratch verstehen • Erkennen, wie visuelle Programmierkonzepte in die Entwicklung mobiler Apps mit Flutter übertragen werden. • Ein einfaches, interaktives Projekt in Scratch bauen und diskutieren, wie ähnliche Logik in Flutter implementiert werden kann.
Unterrichtsplan	Grundlagen des digitalen Designs und der Entwicklung Ausbildungsmodul für gehörlose Jugendliche Dauer: ~90 Minuten (2 Themen × 40 Minuten + Pause 10 Minuten)



Modulplan	
	1. Einführung in Scratch (40 Minuten) Ziel: Blockbasierte Programmierung verstehen und einfache Animationen erstellen. Hauptpunkte: ○ Was ist Scratch? ○ Schnittstelle: Bühne, Sprites, Blöcke. ○ Ereignisgesteuerte Programmierung: "Wenn die Flagge klickte". ○ Eine einfache Animation oder ein Minispiel erstellen. Aktivität: Erstelle ein einfaches Projekt: Bewege eine Figur über den Bildschirm mit Pfeiltasten. Methoden: Demonstration + geführte praktische Übungen 2. Übergang von Scratch zu Flutter (40 Minuten)

Modulplan	
	• Ziel: Verstehen, wie die in Scratch gelernte Programmierlogik auf die Entwicklung von Flutter-Apps angewendet wird. • Hauptpunkte: ○ Wichtige Gemeinsamkeiten: Ereignisse, Schleifen, Konditional. ○ Einführung in Flutter: Widgets, UI-Struktur, Grundlagen der Dart-Sprache. ○ Das Verständnis der visuellen und codebezogenen Verbindung. • Aktivität: Vergleiche ein Scratch-"Button-Click"-Event mit einem Flutter onPressed Button-Event. • Lehrmethodik: Visueller Vergleich, vergleichende Beispiele, Diskussion.
Benötigte Materialien	• Laptop • Internet • Projektor • Gedruckte Materialien



Co-funded by
the European Union



MODULINHALT

Inhalt	Beschreibung	Inhalt
Unterrichtsinhalt	Digitale Kompetenz	Digitale Kompetenz bedeutet, das Wissen und die Fähigkeiten zu besitzen, um digitale Werkzeuge wie Designsoftware und Programmiersprachen zu nutzen, um Apps, Websites und Online-Inhalte zu erstellen. Dazu gehört das Verständnis, wie digitale Systeme funktionieren, und die Fähigkeit, einfache digitale Produkte zu bauen und zu gestalten. • Verständnis digitaler Systeme – wie Computer, Anwendungen und das Internet funktionieren. • Kreativer Einsatz von Werkzeugen – das Entwerfen und Entwickeln einfacher digitaler Produkte. • Problemlösung durch Technologie – Anwendung digitaler Lösungen auf reale Bedürfnisse. Wichtige digitale Begriffe: 1. Blockbasierte Codierung ○ Eine visuelle Programmiermethode, indem "Blöcke" wie Puzzleteile miteinander verbunden werden.

Inhalt	Beschreibung	Inhalt
		○ Beispiel: Scratch – bei dem Blöcke verwendet werden, um Charaktere, Geräusche und Animationen zu steuern. 2. Ereignis & Schleifen ○ Ereignis: eine Aktion, die ein Programm auslöst (z. B. "wenn grüne Flagge geklickt hat" in Scratch oder "onPressed"-Button in Flutter). ○ Schleife: eine bestimmte Aktion mehrfach wiederholen (z. B. "für immer" oder "wiederholen bis" in Scratch). 3. Widgets ○ Die grundlegenden Bausteine in Flutter, die eine App erstellen (z. B. Buttons, Bilder, Textfelder). ○ Jeder Teil einer Flutter-App ist ein Widget, das angepasst oder kombiniert werden kann. 4. Plattformübergreifende Entwicklung ○ Entwicklung von Anwendungen, die auf verschiedenen Geräten und Betriebssystemen (Android, iOS, Web) funktionieren.

Inhalt	Beschreibung	Inhalt
		Flutter ist ein Beispiel für ein plattformübergreifendes Entwicklungstool.
Videozusammenfassung	Youtube-Links	<i>Scratch-Basics</i>: www.scratch.mit.edu <i>Einführung in Flutter</i>: www.flutter.dev <i>Dart-Sprachübersicht</i>: www.dart.dev
Aktivität		Gruppenaktivität: Baue ein Minispiel von Grund auf Ziel: Teamarbeit und Kreativität fördern bei der Anwendung blockbasierter Codierungskonzepte. Anweisungen: - Die Teilnehmer*innen werden in Kleingruppen (3–4 Mitglieder) unterteilt. - Jede Gruppe wählt ein einfaches Spielkonzept (z. B. Charakter-Sammeln von Objekten, Labyrinth-Flucht oder Reaktionszeit-Herausforderung).



Inhalt	Beschreibung	Inhalt
		3. Mit Scratch entwerfen Gruppen Charaktere (Sprites), fügen Hintergründe hinzu und programmieren Interaktionen mithilfe von Ereignissen und Schleifen. 4. Ermutige dazu, mindestens ein Punktesystem oder eine Gewinn-/Niederlage-Bedingung hinzuzufügen. Ergebnis: Jede Gruppe erstellt ein spielbares Minispiel, das sie mit der Klasse teilen kann. 2. Rollenspiel: Übersetze Scratch-Logik in Flutter Ziel: Teilnehmer*innen können visuelles blockbasiertes Programmieren mit textbasiertem Coding in Flutter verbinden. Anweisungen: 1. Nach Abschluss des Minispiels sollte ein Gruppenmitglied als "Scratch Developer" und ein anderes als "Flutter Developer" fungieren. 2. Der Scratch-Entwickler erklärt die Logik seines Minispiels (z. B. "Wenn der Sprite den Apfel berührt, steigt die Punktzahl um 1").

Inhalt	Beschreibung	Inhalt
		3. Der Flutter Developer zeigt, wie diese Logik in Flutter ausgedrückt werden könnte (z. B. durch Verwendung eines onPressed-Ereignisses , bedingte Anweisungen oder Aktualisierung einer Score-Variablen in Dart). Ergebnis: Die Teilnehmer*innen üben die Erklärung der Programmierlogik auf zwei Plattformen hinweg und verbessern so die digitale Kompetenz und Kommunikationsfähigkeiten.
Bewertung	Quiz	1. Wofür wird Scratch hauptsächlich verwendet? A) Hardwaresteuerung B) Visuelles Programmieren für Anfänger ✓ C) Nur textbasierte Programmierung D) keine 2. Welche Sprache wird von Flutter verwendet? A) Java B) Dart ✓ C) Python D) C++

Inhalt	Beschreibung	Inhalt
		3. In Scratch, welcher Block startet die meisten Programme? A) Wenn die grüne Flagge geklickt wurde ✓ B) Wiederholen bis C) Für immer D) Warten 4. Für welche Art von Programmierung ist Scratch am bekanntesten? A) Blockbasierte Programmierung ✓ B) Kommandozeilenprogrammierung C) Maschinelles Lernen D) Datenbankverwaltung 5. Was ist in Flutter ein "Widget"? A) Hardwaregerät B) grundlegender Baustein der Benutzeroberfläche ✓ C) Programmiersprache D) Datenbankverbindung

Inhalt	Beschreibung	Inhalt
		6. Was bedeutet "plattformübergreifende Entwicklung"? A) Apps nur für Android bauen B) Apps erstellen, die auf mehreren Betriebssystemen laufen ✓ C) Code ohne Computer schreiben D) ausschließlich blockbasierte Programmierung verwenden