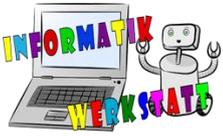


Grundlagen zur Programmierung des micro:bit in Python

Zielgruppe:	Schüler*innen ab der 4. Klasse Sekundarstufe 1 bzw. Sekundarstufe 2
Zeitraumen:	Ca. 10 Unterrichtseinheiten
Fach:	Digitale Grundbildung, Informatik, Mathematik
Lehrplanbezug:	Digitale Grundbildung, Informatik: Computational Thinking, Algorithmen, Programmierung, Problemlösen, Mathematik: Variablen, Zufallsbegriff, Problemlösen, algorithmisches Denken
Informatikkonzepte	Computational Thinking, Algorithmen, Software
Typ/Art des Unterrichtsmaterials:	Theorie mit Beispielen und zugehörigen Aufgaben; ähnlich wie Schulbuch
Benötigte Dateien:	SW_I_microbit-GLPython-xxx: Informationen und Anleitungen SW_AA_microbit-GLPython-xxx: Arbeitsaufträge SW_LO_microbit-GLPython-xxx: Lösungen
Utensilien:	Computer mit Internetzugang, micro:bit und USB-Verbindungskabel
Sozialform:	Einzel- oder Partnerarbeit
Lehrziele:	Die Schüler*innen lernen das Konzept der Algorithmen spielerisch kennen und verstehen. Dabei setzen sie sich mit den wichtigsten Programmierkonzepten (Verzweigung, Variablen, Schleifen, ...) auseinander und wenden diese selbstständig in Aufgaben an.
Quellen:	AHS-Lehrpläne in BGBl. II Nr. 133/2000: https://www.bmb.gv.at/schulen/unterricht/lp/lp_ahs_unterstufe.html (19.4.2018) Digitale Grundbildung BGBl. II Nr. 71/2018: https://www.ris.bka.gv.at/eli/bgbl/II/2018/71/20180419 (19.4.2018) Alle Bilder CC-BY-NC-SA Informatik-Werkstatt 2019 oder lizenzfreie Grafiken von Pixabay, wenn nicht explizit angegeben
Autor/innen:	Nina Lobnig, Markus Wieser, Katharina Brugger
Lizenz:	CC-BY-NC-SA Informatik-Werkstatt AAU 2020



Vorbereitung:

Drucken Sie für alle Schüler*innen die Informationsblätter **SW_I_microbit-GLPython-xxx** sowie die dazugehörigen Arbeitsaufträge **SW_AA_microbit-GLPython-xxx** aus. Wenn die Lernenden später nochmal etwas nachlesen wollen, können sie dort nachsehen und zudem können sie sich mit den ausgedruckten Aufgaben gut unterschiedlich lange beschäftigen und auch üben (z.B. zuhause).

In den Dateien steht AA für Arbeitsblatt und I für die Informationsblätter, die Sie in den Vorführungsphasen verwenden. Bezüglich der Reihenfolge der Arbeitsblätter orientieren Sie sich an der römischen Nummerierung (I, II, III, IV, V, VI) im Dateinamen.

Alle verwendeten Computer benötigen eine Internetverbindung. Zudem empfiehlt es sich mehrere micro:bit zu kaufen, um für je zwei Schüler*innen einen parat zu haben. Man kann zwar auch rein mit der Simulation auf der Webseite bzw. im Computerprogramm arbeiten, die Motivation der Schüler*innen ist aber definitiv eine andere, wenn man die Minicomputer auch tatsächlich anfassen und physisch damit arbeiten kann (z.B. Tasten drücken, Schütteln, Neigen).

Die Schüler*innen sollten ohne Kenntnis von Python und blockbasierten Programmiersprachen diese Einheiten bewältigen können. Man kann aber als Lehrperson darauf aufbauen. Vor allem beim Einstieg kann es helfen, wenn man das Konzept der Methode erklären möchte, da hier in dem ersten Informationsblatt auch auf blockbasiertes Programmieren verwiesen wird.

Hinweis:

Allgemein wird empfohlen, sich als Lehrperson die Aufgaben nicht nur anzuschauen, sondern auch zu programmieren, um Problemfelder und Hindernisse zu erkennen und im Unterricht zu thematisieren.

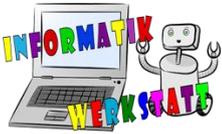
Weiters muss man bedenken, dass sich die Python-Programmierung mit dem micro:bit *grundlegend* von „normalen“ Pythoncode *unterscheidet*. Somit ist ein *direkter Umstieg* von micro:bit-Python auf strukturierte Programmierung mit Python nicht möglich.

Einsatz/Handhabung:

In den Unterrichtseinheiten wechseln sich Vorführungsphasen und Arbeitsphasen ab. Zum Ablauf der Einheit und zu den Übungsaufgaben kann man sich direkt an den Dateien **SW_I_microbit-GLPython_XxX** (Informationen und Erklärungen) und **SW_AA_microbit-GLPython_XxX** (Übungsaufgaben) bzw **SW_LO_microbit-Grundlagen** orientieren und leiten lassen. Es ist dort der Ablauf, Aufgaben und alles Weitere genau erklärt.

Am Anfang der Unterrichtsstunden zum Thema micro:bit sollen sich die Schüler*innen damit zunächst einmal vertraut machen. Man kann dazu diese schon austeilen oder durch die Reihen geben und dann den micro:bit und seine Eingabe- und Ausgabemöglichkeiten kurz besprechen (Tastendruck, Bewegungen wie Schütteln, LED-Anzeige usw.). Danach geht es ans Verbinden des micro:bit mit der USB-Buchse des Computers und folgend um die Art der Programmierung und Erklärung der Benutzeroberfläche. Eine Anleitung dafür findet sich in der Informationsdatei **SW_I_microbit-GLPython- I_Einfuehrung** gleich am Beginn.

Nach der kurzen Einführung in die Software beginnt man schon mit der Vorführung und Erstellung des ersten Programms, das aus Befehlen der Kategorie „Grundlagen“ besteht. Dazu erklärt man zunächst die wichtigsten Befehle der Kategorie und erzeugt dann ein Programm, welches beim Start „Hallo“ anzeigt und danach dauerhaft einen Smiley. Ist man so weit, gibt man dem Programm einen sinnvollen Namen und ladet das Programm herunter und verschiebt es auf den micro:bit. Dieses Prozedere muss langsam und deutlich gezeigt werden, da die Lernenden dieses bei allen folgenden Beispielen



benötigen werden. Wenn die Schüler*innen in dieser Phase aufmerksam zuhören, benötigen sie später deutlich seltener die Lehrperson beim Ausprobieren. Deshalb empfiehlt es sich, zumindest am Anfang gemeinsam das erste Programm zu erstellen. Später kann man je nach Klasse und Leistung der Schüler*innen weitere, kurze Inputphasen einbauen.

Eine Erklärung der Grundlagen-Befehle und das zugehörige Anfangsbeispiel findet sich natürlich auch auf den Informationsblättern **SW_I_microbit-GLPython-XxX**. Danach sollen die Schüler*innen sich kurz selbst an ein paar zugehörigen Aufgaben versuchen, diese sind auf den Arbeitsblättern **SW_AA_microbit-GLPython-XxX** zu finden. Bei den Aufgaben können entweder zwei bis drei ausgewählte Aufgaben probiert werden, oder die Schüler*innen suchen sich selbst welche aus. Es sind auch genügend Aufgaben vorhanden, sodass schnellere Lernende hier auch Beschäftigung finden (indem sie alle Aufgaben lösen). Natürlich können – und sollen – eigene Ideen der Schüler*innen gerne in den Arbeitsphasen umgesetzt werden. Alle Arbeitsblätter wurden so gestaltet, dass jede Aufgabe eine kleine Checkbox hat, das nach getaner Arbeit abgehakt werden kann. Dadurch erhält man bei den Schüler*innen den Arbeitswillen, falls ihnen freies Arbeiten schwerer fällt.

Varianten und Ergänzungsmöglichkeiten:

Man kann die Schüler*innen am Ende an einem größeren eigenen Projekt arbeiten lassen, in dem sie die gelernten Kenntnisse nochmals anwenden. Weitere Befehlskategorien, wie Funk können selbst erkundet werden. Erweiterungen können gekauft werden, wie z.B. BitBots (fahrende Spielzeugautos, in die man den micro:bit hineinsteckt) oder man programmiert Spiele und andere Miniprojekte (Anleitungen gibt es dazu viele im Internet oder auch auf RFDZ-Informatik – z.B. *micro:bit Stationenbetrieb Vorübung in Python*). Man kann auch mit den Pins arbeiten und ein Steckbrett dazu nehmen (z.B. fächerübergreifend mit Physik und dem Thema Strom - Arbeitspaket mit Arduino in C/C++ oder micro:bits blockbasiert auf RFDZ-Informatik). Aber auch ein Umstieg auf JavaScript ist ohne weiteres möglich. Der Kreativität sind beim micro:bit keine Grenzen gesetzt.

Nützliche Links

- <https://makecode.microbit.org/>
- <http://www.microbit.at/>
- https://microbit.eeducation.at/wiki/Arbeiten_mit_dem_BBC_micro:bit
- <https://microbit.org/de/quide/>
- <https://makecode.microbit.org/docs> (Dokumentation und Beschreibung aller Blöcke in allen drei Sprachen)