FAIBLE.nrw

Didaktik der Programmierung

UNIVERSITÄT PADERBORN


Notional Machine
Modellieren
Projektunterricht
Bildung
Pair Programming
PRIMM
Programmiersprachen
Software Life Cycle
4C/ID
Kompetenzen
Tools
Softwareentwicklung
Problemfelder
Lernroboter
Implementieren
Unterrichtsmethoden
Lernerfolg- und Leistungsbewertung
Worked Example
Inhalte
Lernziele
Programmverstehen
Digitale Welt
Debugging
Cognitive Load
STREAM
Block-basierte Sprachen
Computational Thinking
Programmieren
Use-Modify-Create
Programmierkonzepte
Computational Notebooks
Softwareprojekte
Lehrpläne

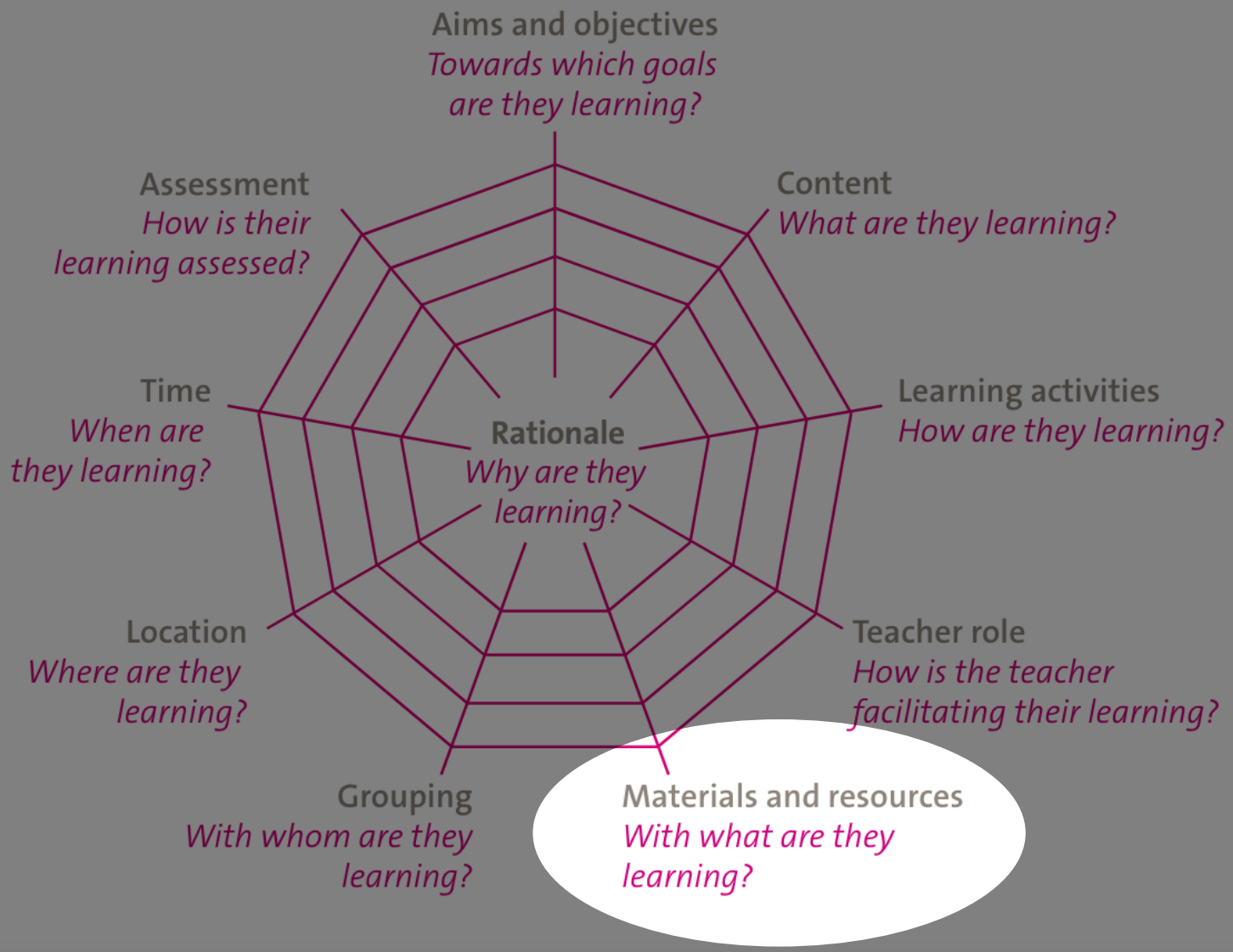
Kapitelübersicht

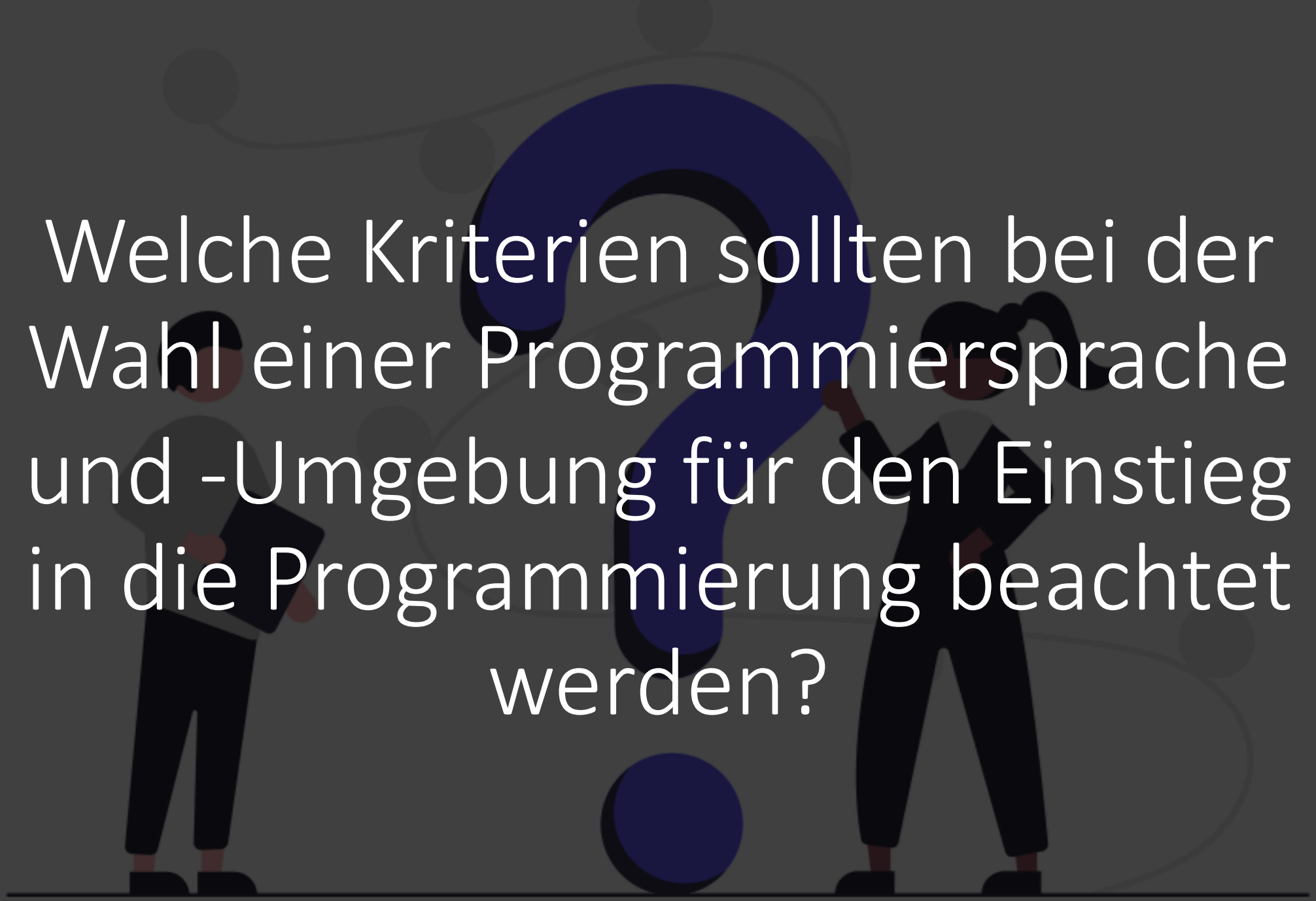
5. Programmiersprachen- und Umgebungen

Text-basierte versus Block-basierte Sprachen

Programmierumgebungen

Lernroboter


Aims and objectives
Towards which goals are they learning?
Content
What are they learning?
Learning activities
How are they learning?
Teacher role
How is the teacher facilitating their learning?
Materials and resources
With what are they learning?
Grouping
With whom are they learning?
Location
Where are they learning?
Time
When are they learning?
Assessment
How is their learning assessed?
Rationale
Why are they learning?



Welche Kriterien sollten bei der Wahl einer Programmiersprache und -Umgebung für den Einstieg in die Programmierung beachtet werden?

Akzeptanzkriterien

(Perera et al., 2021)

- Wahrgenommener Nutzen
  - Programmierkonzepte und Prinzipien
  - Sichtbarkeit der Ausführung
  - Erweiterbarkeit der Sprache
  - Lebendigkeit und Anreiz zum Tüfteln
- Wahrgenommene Zugänglichkeit
  - Syntax und Grammatik
  - Benutzeroberfläche
  - Verfügbarkeit von (Lern-)Ressourcen

TAM Dimension	Language features	Effect on Attitude towards Using
Perceived Usefulness	Programming concepts and principles	Both the block-based and text-based languages contain basic programming concepts. Hence both create a positive attitude towards learning. However, at times block-based languages present the concepts more effectively to the novices.
	Execution visibility	Execution visibility creates a positive attitude towards the use of the programming language.
	Language extensibility	Language extensibility has a major contribution towards a positive attitude to use. This feature is mostly available in Block-based languages.
	Liveness and Tinkerbility	Block-based languages create a positive attitude towards learning as they offer better liveness and tinkerbility features.
Perceived Ease of Use	Grammar and Syntax	Block-based languages create a positive attitude towards learning. However, the long-term effect is negative due to difficulties in switching to more advanced programming languages. Text-based languages generate a less positive attitude initially due to syntax and grammar errors.
	User Interfaces	Highly visual interfaces create a positive attitude towards the use of the programming language/ platform.
	Availability of Resources	Greater availability of learning resources and support materials create a positive attitude in novices towards using the programming language/platform.

Perera et al. (2021). Technology Acceptance Model (TAM)

Perera, P., Tennakoon, G., Ahangama, S., Panditharathna, R., & Chathuranga, B. (2021). A Systematic Mapping of Introductory Programming Languages for Novice Learners.

Unterscheidungsmerkmale (Auswahl)

(vgl. Perera et al., 2021)

- Einsatzbereich
  - Professionell (Softwareentwicklung)
  - Edukativ (z.B. Schule)
- Darstellung
  - Text-basiert (z.B. Java, Python)
  - Block-basiert / Visuell (z.B. Scratch)
  - Modellierung (z.B. UML)
  - Pseudocode
- Typisierung
  - Stark typisiert (z.B. C#)
  - Schwach typisiert (z.B. JavaScript)
- Funktionsumfang
  - Code-Vervollständigung
  - Debugger
  - …
- Paradigma
  - Objektorientiert
  - Funktional
  - …
- Kontext
  - Datenbanken
  - Webentwicklung
  - Data Science
  - …
- Praktiken
  - Softwareentwicklung
  - Tinkering
  - Live coding
  - …

Perera, P., Tennakoon, G., Ahangama, S., Panditharathna, R., & Chathuranga, B. (2021). A Systematic Mapping of Introductory Programming Languages for Novice Learners.

Entwicklungsumgebungen

Professionelle Entwicklungsumgebungen

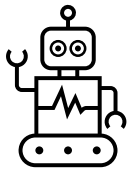
Block-basierte Umgebungen

Computational Notebooks

Lernroboter

Beispiele für Entwicklungsumgebung

- Professionelle Entwicklungsumgebungen
  - Eclipse
  - PyCharm
- Edukative Umgebungen
  - Text-basierte Umgebungen
    - BlueJ
    - Greenfoot
    - Jupyter Notebook
  - Block-basierte Umgebungen
    - Scratch
    - Blockly
  - Lernroboter
    - Lego Mindstorms EV3
    - mBlock



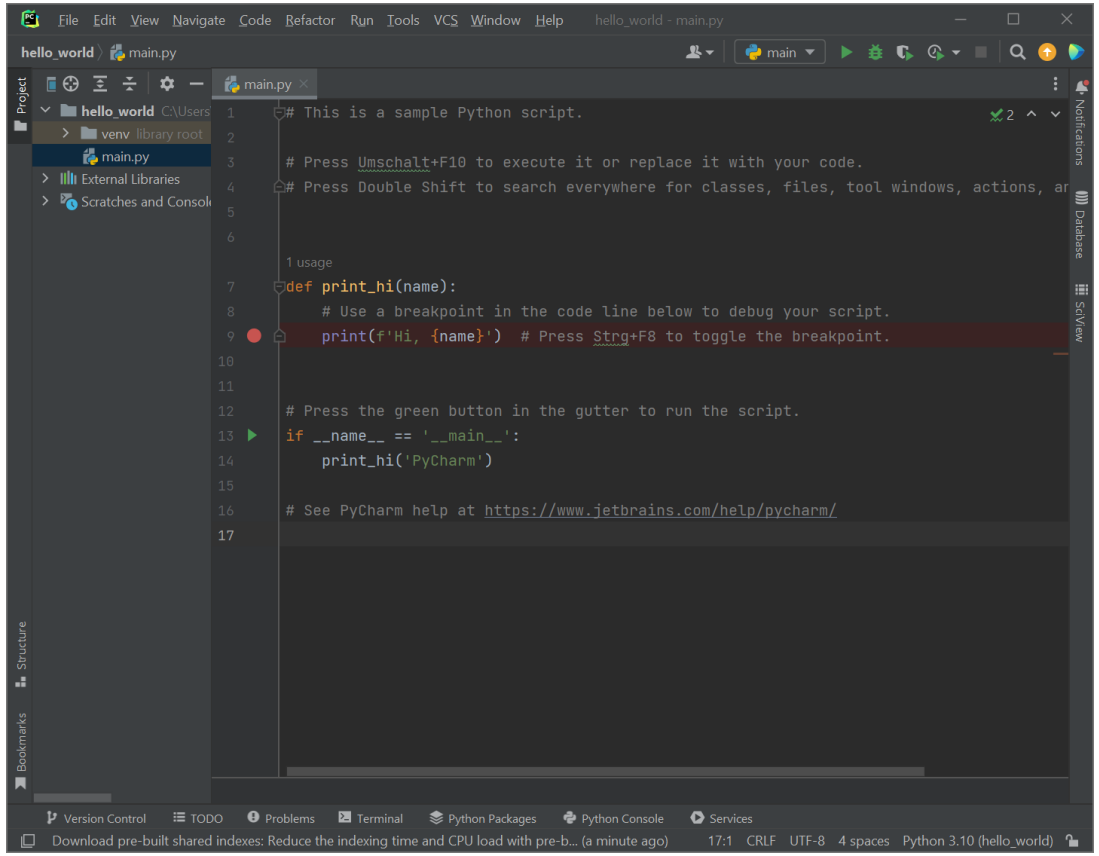
Text-basierte Programmierumgebungen

Professionelle & Didaktische Programmierumgebungen

Professionelle Entwicklungsumgebungen

(z.B. Eclipse, PyCharm)

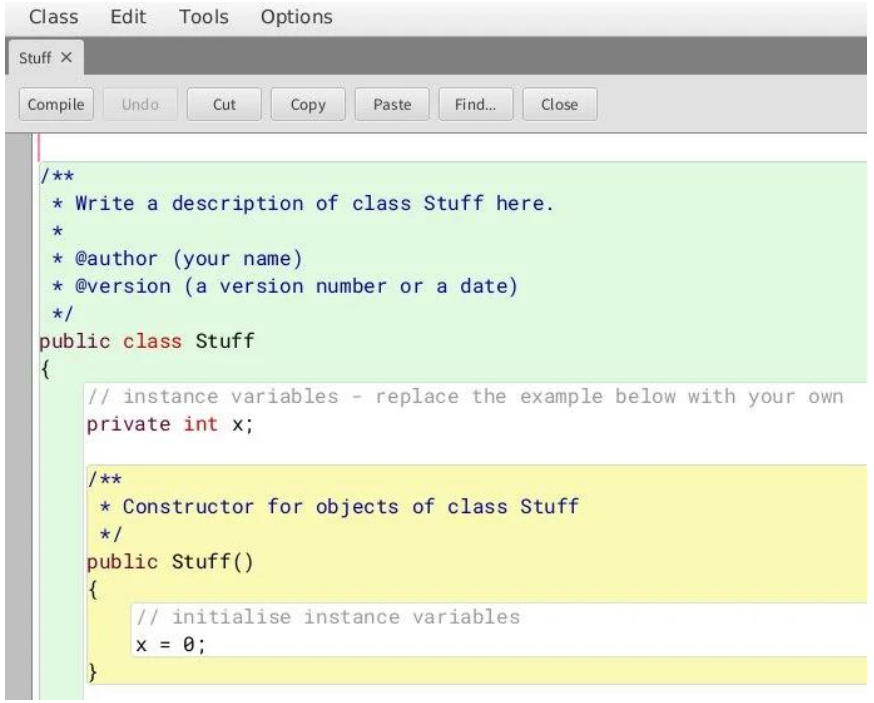
- Großer Funktionsumfang
  - Kann zu Überforderung und kognitiver Überlastung führen *(siehe Kapitel 3)*
- Skalierbarkeit
- „Authentische“ Programmierung
- Integration von Werkzeugen
  - Versionskontrolle
  - Debugger
  - Terminal
  - Paket-Manager



Didaktische Entwicklungsumgebungen

am Beispiel BlueJ *(vgl. Kölling et al., 2003)*

- Fokus: Objektorientierte Programmierung (OOP)
  - Grafische Klassenansicht
  - UML Klassendiagram
  - Interaktive Objekterzeugung & Methodenaufrufe
  - Object Inspector (Belegung der Attribute)
- Wenig UI-Elemente
- Scope-Highlighting *(siehe Screenshot)*
  - vgl. Block Modell *(Kapitel 3)*
- Weitere Features
  - Javadoc
  - Debugger



Kölling, M., Quig, B., Patterson, A., & Rosenberg, J. (2003). The BlueJ System and its Pedagogy. *Computer Science Education*, *13*(4), 249–268. https://doi.org/10.1076/csed.13.4.249.17496

Block-basierte Programmierumgebungen

Block-basierte / Visuelle Umgebungen

(z.B. Scratch)

- Vermeidung von Syntax-Fehler
  - Entlastung kognitiver Ressourcen
- Programmierelemente sind jederzeit sichtbar
  - Schüler:innen können daraus wählen und damit experimentieren
  - Auswendiglernen wird reduziert „Wie mache ich noch mal eine Schleife?“
- Kein „authentisches“ Programmieren
- Nicht beliebig skalierbar

Block-basierte vs. Text-basierte Sprachen

(vgl. Perera et. al., 2021)

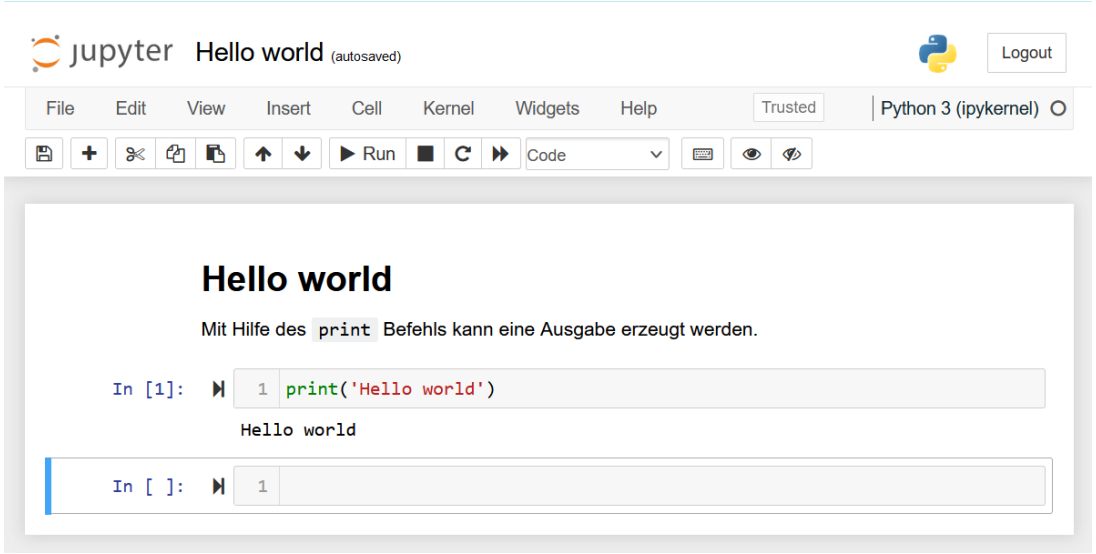
	Block-basiert	**Text-basiert**
Vorteile	• Vermeidung von Syntax-Fehlern • Drag & Drop • Sichtbarkeit der Elemente • Ausführung der Anweisungen ist sichtbar	• Authentische Programmierung • Konkrete Fehlermeldungen • Beliebig skalierbar • Kompakter
Nachteile	• Flexibilität und Interkompatibilität eingeschränkt • Ungeeignet für komplexe Projekte • Plattformabhängig • Erschwerter Übergang zu text-basierten Sprachen	• Syntaxfehler möglich • Schlüsselwörter und Funktionsnamen müssen behalten werden

Perera, P., Tennakoon, G., Ahangama, S., Panditharathna, R., & Chathuranga, B. (2021). A Systematic Mapping of Introductory Programming Languages for Novice Learners.

Computational Notebooks

(z.B. Jupyter Notebooks)

- Integration von
  - ausführbaren Programmcode,
  - formatierten Text und
  - Programmausgabe in einem Dokument
- Verringerung von Medienbrüchen
- Literate Programming *(Knuth, 1984)*
- Zellen können selektiv ausgeführt werden
  - Regt zum Explorieren und Tüfteln an
  - Belegung der Variablen bleibt erhalten



„Jupyter Notebook Screenshot“ von Sören Sparmann

Knuth, D. E. (1984). Literate Programming.

„Physical Computing bedeutet im weitesten Sinne, interaktive, physische Systeme durch die Verwendung von Hardware und Software zu erstellen. Diese Systeme reagieren auf Ereignisse in der realen, analogen Welt und/oder wirken auf sie ein.“ *(Wikipedia, Physical Computing)*

Physical Computing

Lernroboter

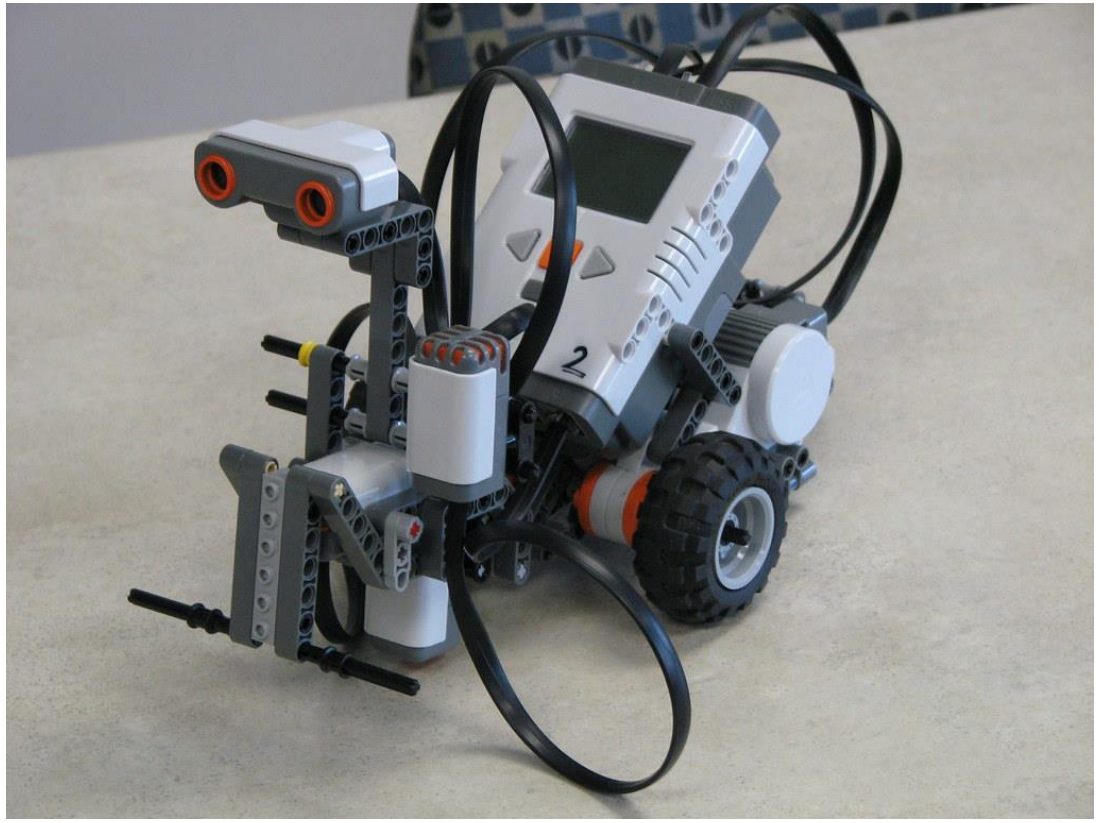
senseBox

https://de.wikipedia.org/wiki/Physical_Computing

Lernroboter

(Alam, 2022)

- Haptisches Lernen
- Problemlösekompetenz
- Interdisziplinäres Lernen
  - Vermittlung von technischem und ingenieurwissenschaftlichem Wissen
  - Möglichkeit zum Entwerfen, Bauen und Testen eigener Erfindungen
- Motivations- und Spaßfaktor
- Interaktion mit der physischen Welt



Alam, A. (2022). Educational Robotics and Computer Programming in Early Childhood Education: A Conceptual Framework for Assessing Elementary School Students' Computational Thinking for Designing Powerful Educational Scenarios.

senseBox

- Umweltmessstation
- Einstieg in die Microcontroller Programmierung
  - Block-basierte Sprache oder Arduino IDE
- Sensoren
  - Temperatur, Luftfeuchtigkeit, Luftdruckt, Beleuchtungsstärke, UV-Intensität, Distanzen und Lärm
- Daten können gespendet werden
  - vgl. Citizen Science
  - openSenseMap
- Ermöglicht verschiedene Projekte, in denen die Schüler:innen Daten aus ihrer Umwelt sammeln, aufbereiten und auswerten

Referenzen

- Alam, A. (2022). Educational Robotics and Computer Programming in Early Childhood Education: A Conceptual Framework for Assessing Elementary School Students' Computational Thinking for Designing Powerful Educational Scenarios. In *2022 International Conference on Smart Technologies and Systems for Next Generation Computing (ICSTSN) (S.* 1 – 7). https://doi.org/10.1109/ICSTSN53084.2022.9761354.
- Knuth, D. E. (1984). Literate Programming. *The computer journal*, *27*(2), 97–111.
- Kölling, M., Quig, B., Patterson, A., & Rosenberg, J. (2003). The BlueJ System and its Pedagogy. *Computer Science Education*, *13*(4), 249–268. https://doi.org/10.1076/csed.13.4.249.17496
- Kelleher, C. & Pausch, R.
  (2005): Lowerin the barriers to programming: A taxonomy of programming environments and languages for novice programmers. In ACM Computing Surveys 37 (2) (S. 83 –
  137). https://doi.org/10.1145/1089733.1089734.
- Perera, P., Tennakoon, G., Ahangama, S., Panditharathna, R., & Chathuranga, B. (2021). A Systematic Mapping of Introductory Programming Languages for Novice Learners. In *IEEE Access 9 (S.* 88121 – 88136). https://doi.org/10.1109/ACCESS.2021.3089560.

Illustrationen

- Illustrationen von Limpitsouni, K. unter freier Lizenz via https://undraw.co

FAIBLE.nrw

Beteiligte Hochschulen:

- RWTH-Aachen
- Westfälische Wilhelms-Universität Münster
- Universität Duisburg-Essen
- Universität Bonn
- Universität Paderborn
- Technische Universität Dresden
- Carl von Ossietzky Universität Oldenburg

ORCA.nrw
Das Landesportal für Studium und Lehre.

Ein Kooperationsvorhaben empfohlen durch die:
DIGITALE HOCHSCHULE NRW
INNOVATION DURCH KOOPERATION

gefördert durch:
Ministerium für Kultur und Wissenschaft des Landes Nordrhein-Westfalen