

Didaktische Hinweise

Zeichenkettenverarbeitung mit Snap!¹

Zielgruppe

Der Leitfaden zur Zeichenkettenverarbeitung in Snap! richtet sich an Schülerinnen und Schüler in der Einführungsphase, die bereits mit den algorithmischen Kontrollstrukturen und mit einer der beiden Entwicklungsumgebungen Snap! oder Scratch² vertraut sind.

Voraussetzungen

Wenn die Schülerinnen und Schüler in Jahrgang 9 und / oder 10 bereits mit Scratch gearbeitet haben, sollte ihnen der Umstieg auf Snap! gelingen, ohne dass vorher eine umfangreiche Einführung notwendig ist. Die Schülerinnen und Schüler sollten mit den Kontrollstrukturen Verzweigung und Schleife, mit dem Konzept der Ereignissteuerung sowie mit dem Erzeugen und Verwenden von Variablen vertraut sein. Auch die Blöcke „frage...“ und „antwort“ sind im Umgang mit Zeichenketten hilfreich und sind vermutlich bereits aus der Arbeit mit Scratch bekannt sein. Spezielle Kontrollstrukturen wie die *for*-Schleife, die über den Sprachumfang von Scratch hinausgehen, können im Rahmen der Arbeit mit dem Leitfaden zur Zeichenkettenverarbeitung eingeführt werden.

Wenn noch keine Vorkenntnisse im algorithmischen Problemlösen vorhanden sind, kann zunächst das Materialpaket „Einstieg in das algorithmische Problemlösen mit Snap!“ für einen Einstieg verwendet werden.

Lernziele

Dieser Leitfaden führt in die Zeichenkettenverarbeitung mithilfe elementarer Zeichenkettenoperationen ein. Die Schülerinnen und Schüler lernen ...

1. ... Benutzereingaben auszuwerten.
2. ... auf einzelne Zeichen in einer Zeichenkette zuzugreifen.
3. ... eine Zeichenkette Zeichen für Zeichen zu durchlaufen und zu verarbeiten.
4. ... die ASCII-bzw. Unicode-Codierung der Zeichen bei der Verarbeitung zu verwenden.

Die Schülerinnen und Schüler verwenden dabei die folgenden Zeichenkettenoperationen:

- Bestimmen der Länge einer Zeichenkette
- Auslesen eines Zeichens an einer bestimmten Position
- Verbinden von zwei Zeichenketten zu einer
- Prüfen des Inhalts von zwei Zeichenketten auf Gleichheit
- Bestimmen des ASCII-Werts (Dezimalzahl, hier als Unicode) zu einem Zeichen (und umgekehrt)

¹ Snap! wird von der University of California, Berkeley zur Verfügung gestellt: <https://snap.berkeley.edu>

² Scratch ist ein Projekt der Scratch Foundation in Zusammenarbeit mit der Lifelong Kindergarten Group des MIT Media Lab. Es ist kostenlos unter <https://scratch.mit.edu> erhältlich.

Da in Niedersachsen ab dem Abitur 2027³ auch die folgenden Zeichenkettenoperationen zugelassen sind, werden diese in einem Ausblick vorgestellt:

- Auslesen einer Teilzeichenkette in einem bestimmten Bereich
- Prüfen einer Zeichenkette auf eine Teilzeichenkette

In der Einführungsphase kann aber auch auf diese Operationen verzichtet werden, da sie für die Anwendungsaufgaben in diesem Leitfaden nicht zwingend benötigt werden und nur durch das Laden der zusätzlichen Bibliothek "Strings, Multi-line input" zur Verfügung stehen. In der Qualifikationsphase können die Operationen dann in Kontexten, in denen sie hilfreich sind, eingeführt werden. Ähnliches gilt für das lexikographische Vergleichen von Zeichenketten. Der lexikographische Vergleich kann mithilfe der Vergleichs-Blöcke, die auch für den Vergleich von Zahlen verwendet werden, erfolgen. Zu beachten ist, dass Großbuchstaben dabei „kleiner“ als Kleinbuchstaben sind. Dies kann dadurch erklärt werden, dass der Vergleich auf Basis des Unicodes erfolgt. Die Verwendung der Blöcke ist daher nicht ganz intuitiv und sollte erst erfolgen, wenn das nötige Hintergrundwissen bzgl. des Unicodes vorliegt. Die Aufgabenstellungen im Leitfaden kommen ohne lexikographisches Vergleichen aus. Sinnvolle Anwendungen ergeben sich, wenn z. B. Zeichenketten in einer Liste sortiert werden sollen.

Die Aufgaben

Viele der Übungsaufgaben stammen aus dem Bereich der Kryptologie, so dass sich dieser Leitfaden gut mit dem Erlernen der Kompetenzen aus dem Modul Kryptologie kombinieren lässt. Eine Implementierung des Caesar-Verfahrens wird im niedersächsischen Kerncurriculum für die Einführungsphase explizit gefordert. Dies ist daher eine Problemstellung, die sich die Lernenden anhand des Leitfadens Schritt für Schritt erarbeiten können.

Die Aufgaben üben und festigen jeweils die vorhergehenden Erläuterungen im Leitfaden. Der Schwierigkeitsgrad variiert. Die Aufgaben 7b, 10c, 16d eignen sich vor allem für besonders leistungsstarke Schülerinnen und Schüler.

Zu Beginn wird ein Überblick über die zur Verfügung stehenden Blöcke gegeben. In den Aufgaben 1 bis 3 liegt der Schwerpunkt bei der Auswertung von Nutzereingaben zunächst auf dem Vergleich von Zeichenketten und der Abfrage der Länge einer Zeichenkette. In Aufgabe 4 wird dann das systematische Durchlaufen einer Zeichenkette eingeführt. Hier fällt es Lernenden gelegentlich schwer, zwischen dem Wert der Zählvariablen i und dem Zeichen an der Position i zu unterscheiden. In Zeile 5 des Beispiels zu Aufgabe 4 wird das i -te Zeichen der durchlaufenen Zeichenkette daher in einer separaten Variablen gespeichert, um diesen Schritt explizit zu machen. Zur Veranschaulichung des Ablaufs der *for*-Schleife erstellen die Lernenden in Aufgabenteil b) dann eine Trace-Tabelle. Die *for*-Schleife kann durch eine *repeat*-Schleife mit einer eigenen Zählvariablen, die explizit erhöht wird, ersetzt werden, um den Ablauf der *for*-Schleife zu verdeutlichen. Das Konzept des zeichenweisen Durchlaufens und Verarbeitens einer Zeichenkette kann mithilfe der Aufgaben 6 bis 10 gefestigt werden. Die Implementierung zu Aufgabe 8 vereinfacht sich etwas, wenn der Block "text ... contains ..." zum Prüfen einer Zeichenkette auf eine Teilzeichenkette verwendet wird. Je nach Lerngruppe bietet es sich daher ggf. an, die Methode hier bereits einzuführen.

³ Niedersächsisches Kultusministerium (Hrsg.) (2025) *Ergänzende Hinweise zum Kerncurriculum Informatik für die gymnasiale Oberstufe am Gymnasium, an der Gesamtschule sowie für das Kolleg.*
<https://cuvo.nibis.de/index.php?p=download&upload=736> [Datum des Zugriffs: 13.08.2025]

Schließlich wird die Umwandlung von Zeichen in den zugehörigen Zahlenwert des Unicodes und umgekehrt thematisiert. Diese Umwandlung wird insbesondere zur Implementierung des Caesar-Verfahrens benötigt. Dieses kann daher als Motivation dienen. Da das Caesar-Verfahren aus mehreren Teilschritten besteht und die selbständige Implementierung einigen Schülerinnen und Schülern erfahrungsgemäß schwer fällt, werden in den Aufgaben 11 bis 13 zunächst Teilschritte erarbeitet, bevor das Verfahren dann in Aufgabe 16 implementiert wird.

Als Hilfestellung kann gemeinsam mit den Lernenden eine Grafik wie in Abbildung 1 und falls weitere Hilfestellung notwendig ist, ein Struktogramm entwickelt werden.

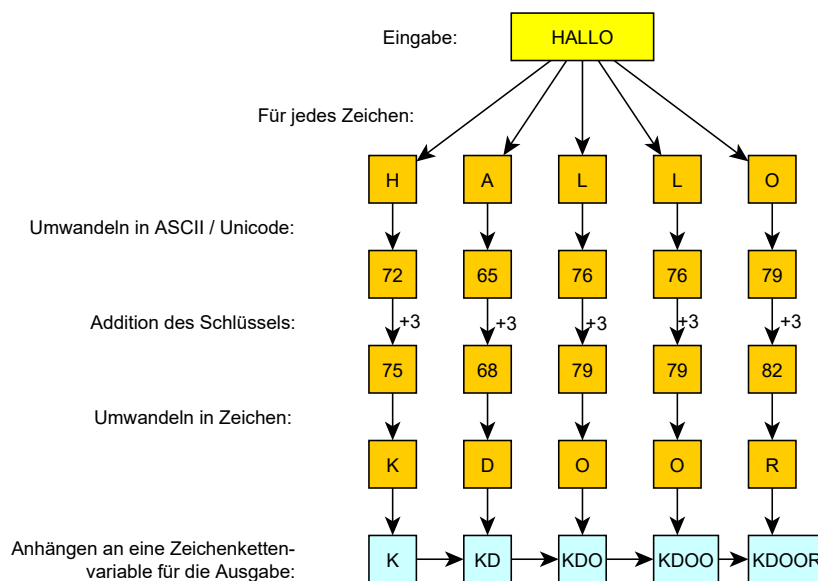


Abbildung 1: Veranschaulichung der Teilschritte beim Caesar-Verfahren

Der Leitfaden mündet in einem Projekt, in dem die Schülerinnen und Schüler alle erlernten Kompetenzen in Rahmen einer etwas umfangreicheren Problemlösung erproben und anwenden können. Dabei können die prozessbezogenen Kompetenzen aus dem Bereich Kreatives Schaffen und Problemlösen besonders gefördert werden⁴. Da die Einführung des Vigenère-Verfahrens erst in der Qualifikationsphase verpflichtend vorgesehen ist, können hier je nach Leistungsstärke der Lernenden auch einfachere Verfahren implementiert werden.

Lösungsvorschläge für die Aufgaben 1 bis 20 können dem Ordner Lösungen entnommen werden. Dabei ist zu beachten, dass es in der Regel viele verschiedene mögliche Implementierungen gibt und hier nur exemplarisch eine vorgestellt wird.

Allgemeine Überlegungen zur Wahl der Werkzeuge

Seit der Einführung des Pflichtfachs Informatik in Jahrgang 9 und 10 haben die Schülerinnen und Schüler, wenn sie in die Einführungsphase eintreten, in der Regel schon Erfahrungen mit dem algorithmischen Problemlösen in Scratch gesammelt. Darauf kann in Jahrgang 11 aufgebaut werden, auch wenn ein Wechsel des Werkzeugs zu empfehlen ist, da der Funktionsumfang von Scratch den Anforderungen in Jahrgang 11 nicht mehr uneingeschränkt gerecht wird.

⁴ vgl. Niedersächsisches Kultusministerium (Hrsg.) (2017) Kerncurriculum für das Gymnasium - gymnasiale Oberstufe, die Gesamtschule – gymnasiale Oberstufe, das Kolleg. Informatik. Hannover: unidruck

Je nach Angebot des Faches in der Oberstufe kann die Wahl unterschiedlicher Werkzeuge für Jahrgang 11 sinnvoll sein. Aufbauend auf Scratch kann mit der blockbasierten Sprache Snap! gearbeitet werden. Der Aufbau der Entwicklungsumgebungen ist ähnlich, so dass der Umstieg relativ leicht ist. Die zur Verfügung gestellten Blöcke, insbesondere im Bereich der Zeichenkettenverarbeitung, sind aber umfangreicher. Dieses Werkzeug ist vor allem für Schülerinnen und Schüler, die Informatik nur in Jahrgang 11, aber nicht in der Qualifikationsphase belegen, interessant. Soll Informatik hingegen in der Qualifikationsphase fortgeführt werden, ist es für manche Themen und Konzepte hilfreich, auch eine textbasierte Programmiersprache wie Processing/Java zur Verfügung zu haben. Ist in Jahrgang 12/13 eine textbasierte Sprache vorgesehen, sollte in Jahrgang 11 ein entsprechender Einstieg erfolgen, damit die Lernenden sich darauf einstellen können. Häufig wünschen sich Schülerinnen und Schüler zu Beginn von Jahrgang 11 auch einen Einstieg in eine textbasierte Sprache.

Die Zeichenkettenverarbeitung kann daher wahlweise in einer textbasierten Sprache, (die zur Verfügung gestellten Materialien verwenden hier Processing) oder in Snap! erfolgen. Die Materialpakete zur Zeichenkettenverarbeitung in Processing und Snap! sind weitgehend gleich aufgebaut, auch hinsichtlich der Aufgabenstellungen, so dass auch beide Sprachen parallel eingesetzt werden können. Nach einem gemeinsamen Einstieg in das textbasierte Programmieren in Jahrgang 11 könnten die Schülerinnen und Schüler also entscheiden, ob sie lieber mit der blockbasierten Sprache Snap! oder der textbasierten Sprache Processing arbeiten. Wichtig ist, darauf zu achten, dass die Schülerinnen und Schüler, die Informatik möglicherweise in der Qualifikationsphase fortsetzen möchten, ausreichend Übung mit dem dort verwendeten Werkzeug erlangen.

Damit im Unterricht programmiersprachenunabhängig über Algorithmen zur Zeichenkettenverarbeitung gesprochen werden kann, bietet sich in diesem Zusammenhang die Einführung von Struktogrammen an. Dabei ist jedoch zu beachten, dass auch im Struktogramm für eine *for*-Schleife der konkrete Start- und Endwert der Zählvariablen angegeben werden muss. Während Snap! bei 1 beginnt, die Zeichen zu nummerieren, beginnt Processing bei 0. In den ergänzenden Hinweisen zum Kerncurriculum Informatik ist für Struktogramme festgelegt, dass das erste Zeichen den Index 0 hat⁵. Das wäre bei einer Implementierung eines Struktogramms in Snap! dann entsprechend anzupassen. Materialien zur Zeichenkettenverarbeitung mit Processing und zur Einführung von Struktogrammen sind im Materialpaket „Einstieg in die textbasierte Programmierung mit Processing“ enthalten.

Lizenz

Dieses Werk ist lizenziert unter einer [Creative Commons Namensnennung - Nicht kommerziell - Keine Bearbeitungen 4.0 International Lizenz](#).

Für die korrekte Ausführbarkeit der beiliegenden Quelltexte wird keine Garantie übernommen. Auch für Folgeschäden, die sich aus der Anwendung der Quelltexte für die Musterlösungen oder Beispiele des Leitfadens oder durch eventuelle fehlerhafte Angaben ergeben, wird keine Haftung oder juristische Verantwortung übernommen.

⁵ Niedersächsisches Kultusministerium (Hrsg.) (2025) *Ergänzende Hinweise zum Kerncurriculum Informatik für die gymnasiale Oberstufe am Gymnasium, an der Gesamtschule sowie für das Kolleg*.
<https://cuvo.nibis.de/index.php?p=download&upload=736> [Datum des Zugriffs: 13.08.2025]