

DYNAMISCHE DATENSTRUKTUREN: STAPELSPEICHER (STACK)



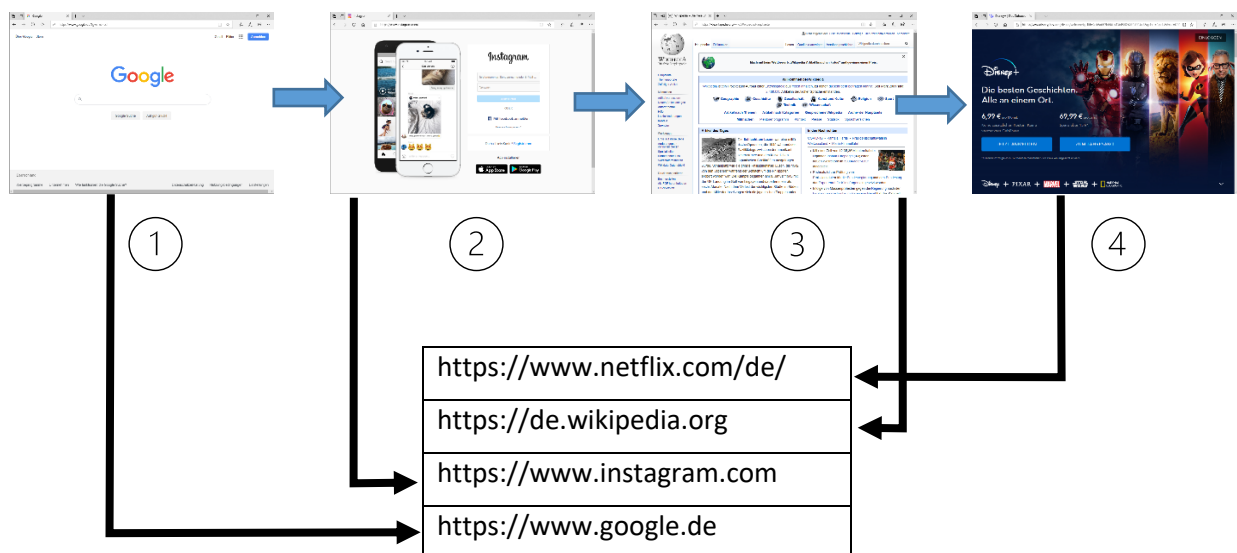
Manchmal ist es bei digitalen Geräten nötig, Daten zu speichern, mit denen man sich später wieder beschäftigen möchte. Diese Daten wirft man einfach auf einen *Stapel*. Das letzte hinzugefügte Element landet also immer ganz oben auf dem Stapel. Das können Sie sich vorstellen wie in der Abbildung dargestellt.

Der Nachteil eines solchen Stapels ist, dass man immer nur direkt an das oberste Element herankommt. Manchmal möchte man aber genau das: Man will auf das Element zugreifen, das als letztes gespeichert wurde!

Wo braucht man einen Stapelspeicher?

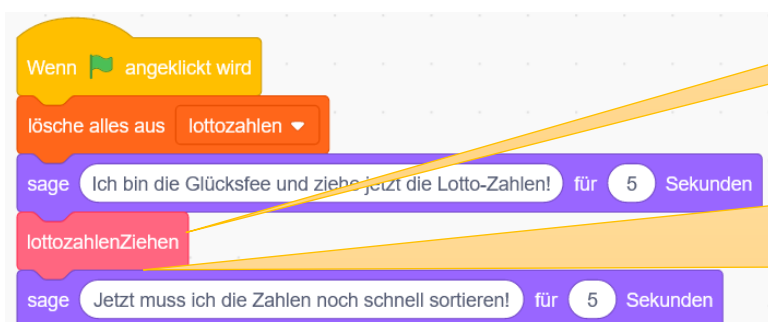
Beispiel 1: Browser

Das offensichtlichste Beispiel für einen Stapelspeicher finden Sie in jedem Browser. Die besuchten Webseiten wandern in einen Stapelspeicher. Mit dem Zurück-Button kann man dann einfach zur letzten besuchten Seite springen.



Beispiel 2: Unterprogramme

Wir rufen in einem Programm ein Unterprogramm auf: Dann muss der Prozessor nach Abarbeitung des Unterprogramms wieder zum ursprünglichen Programm zurückkehren. Diese sogenannte Rücksprungadresse merkt er sich in einem Stapelspeicher!



Wir rufen hier ein Unterprogramm auf!

Im Stapelspeicher wird die Rücksprungadresse des aufrufenden Programms gespeichert! Weil das Unterprogramm weitere Unterprogramme aufrufen kann, kann der Stapelspeicher ganz schön voll werden!



Was man über den Stapelspeicher wissen sollte:

- Der Stapelspeicher ist eine **dynamische Datenstruktur**. Das heißt, dass sich seine Größe jederzeit verändern kann.
- Man kann immer nur direkt auf das oberste Element zugreifen. Dieses Prinzip wird auch **LIFO** genannt (**Last In First Out**).
- Häufig wird für den Stapelspeicher auch der Begriff **Stack** verwendet.

Speichern und Lesen

Mit dem Befehl **PUSH()** können Daten in den Stapelspeicher gelegt werden.

Beispiel:

PUSH("sein!")
PUSH("dir")
PUSH("mit")
PUSH("Macht")
PUSH("die")
PUSH("Möge")



Möge
die
Macht
mit
dir
sein!

Wer jetzt den Befehl **POP()** ausführt, erhält das oberste Element!

Beispiel:

POP()
POP()
POP()



Rückgabe:
Möge die Macht

Der Stapelspeicher
sieht jetzt so aus:

mit
dir
sein!

Aufgaben:

(Quelle aller Aufgaben:

<https://www.schule-bw.de/faecher-und-schularten/mathematisch-naturwissenschaftliche-faecher/informatik/material/materialien-zum-neuen-bildungsplan-informatik-an-den-nichtgewerblichen-beruflichen-gymnasien> - Überarbeitung: Kiltbau / Metz)



1. Ein Textverarbeitungsprogramm speichert jedes getippte Zeichen in einem Stapelspeicher. Bei der Betätigung der Backspace-Taste wird das zuletzt erfasste Zeichen wieder aus dem Stapelspeicher entfernt. Die Backspace-Taste verschiebt den Cursor (=Schreibmarke) um eine Position nach links und entfernt das dort stehende Zeichen.

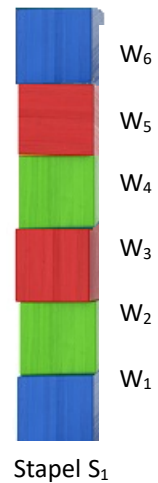


Folgende Operationen wurden durchgeführt:

1 PUSH("A")	2 PUSH("I")	3 PUSH("h")	4 POP()	5 PUSH("g")
6 PUSH("h")	7 POP()	8 PUSH("o")	9 PUSH("r")	10 PUSH("y")
11 PUSH("h")	12 POP()	13 POP()	14 PUSH("i")	15 PUSH("t")
16 PUSH("m")	17 PUSH("h")	18 POP()	19 POP()	20 PUSH("h")
21 PUSH("m")	22 PUSH("i")	23 PUSH("g")	24 POP()	25 PUSH("k")

Analysieren Sie die Abfolge der Operationen und benennen Sie das Wort, das geschrieben wurde.

2. In einem Stapel S_1 befinden sich sechs Würfel ($W_1, W_2, W_3, W_4, W_5, W_6$) in den Farben blau, rot und grün (siehe Abb.).
- Die Würfel sollen mit den Operationen $PUSH()$ und $POP()$ in einen Stapel S_{geordnet} umgeschichtet werden, so dass sich die roten Würfel (W_3 und W_5) unten, die grünen Würfel (W_2 und W_4) in der Mitte und die blauen Würfel (W_1 und W_6) oben im Stapel S_{geordnet} befinden.
- Zur Zwischenablage der Würfel steht Ihnen ein Stapel S_{Ablage} zur Verfügung.
- Notieren Sie in der nachfolgenden Tabelle die Operationen, die notwendig sind, um den geordneten Stapel zu erhalten.



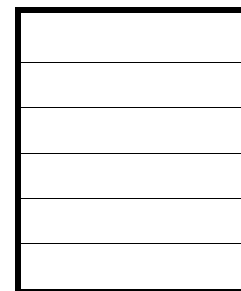
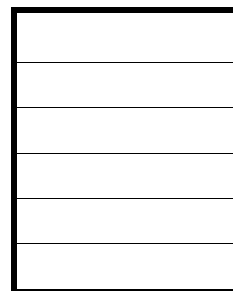
Notation für die Operationen:

S_1 : $POP()$

S_{geordnet} : $PUSH(Element)$

S_{Ablage} : $POP() / PUSH(Element)$

Stapelspeicher:



Sie können die Operationen hier notieren:

Nr.	Operation
1	
2	
3	
4	
5	
6	

Nr.	Operation
7	
8	
9	
10	
11	
12	

Nr.	Operation
13	
14	
15	
16	



Wissen für Nerds: Stack Overflow / Pufferüberlauf

Im Hintergrund verwenden viele Programme und auch das Betriebssystem Stapelspeicher (= Stack). Ein Stack kann zwar theoretisch immer größer werden, in der Praxis hat man aber natürlich nur einen begrenzten Arbeitsspeicher zur Verfügung. Dem Stack wird also meist nur ein begrenzter Speicherbereich zugewiesen.

Damit ist der Stack ein beliebtes Angriffsziel von Hackern. Wenn es den Hackern gelingt, zu viele Daten im Stack zu speichern, werden alte Daten überschrieben. Damit ist es möglich, die Rücksprungadressen von aufrufenden Programmen zu verändern. Die Rücksprungadressen kann man dann so ändern, dass eingeschleuster Schad-Code ausgeführt wird.

Wer es genauer wissen will:
https://de.wikipedia.org/wiki/Return_into_libc