



### Stacks und Queues

Die Listen, die wir ja schon kennen, können in Python auch die Funktion von zwei weiteren Datenstrukturen annehmen: **Stack** und **Queue**.

Ein **Stack** ist wie ein Stapel an Papieren aufgebaut: Es wird immer nur etwas oben drauf gelegt und nur das oberste Element eins nach dem anderen heruntergenommen. Dieses Prinzip nennt man **LIFO** (last in, first out).

Das darauflegen des neuen Elements *e* geht in Python mit `lst.append(e)`<sup>1</sup>, das herunternehmen mit `e = lst.pop()`.

Die **Queue** ist wie die Schlange vor dem Kaffeeautomaten: Neue Elemente werden hinten angehängt und es wird immer nur das vorderste Element bearbeitet. Dieses Prinzip nennt man **FIFO** (first in, first out).

Anhängen (*enqueue*) können wir das neue Element *e* wie gewohnt mit `lst.append(e)`, aber herunternehmen (*dequeue*) müssen wir das oberste Element mit `e = lst.pop(0)`.



### Aufgabe 1: Anwendungsfälle

Überlegen Sie sich gemeinsam Anwendungsfälle für die beiden Datenstrukturen...

- a) ...im Alltag
- b) ...im Kontext des Informatikunterrichts

---

<sup>1</sup>Eigentlich bezeichnet man das darauflegen auf einen Stack als *push*. Da wir aber eine Liste verwenden, nutzen wir dafür *append*.

## **Aufgabe 2: Browserverlauf**

Schreiben Sie ein Programm, das den Browserverlauf eines Nutzers aufzeichnet. Gehen Sie dazu wie folgt vor:

- Der Nutzer gibt die zuletzt aufgerufene URL ein – diese wird gespeichert.
- Gibt der Nutzer `back` ein, soll die letzte URL gelöscht werden und die nun aktuelle URL ausgegeben werden.
- Die Nutzereingabe `quit` beendet das Programm.
- Bei der Eingabe von `clear` soll der Browserverlauf gelöscht werden.

## **Aufgabe 3: Druckaufträge**

Schreiben Sie ein Programm, das nach und nach Druckaufträge bearbeitet. Gehen Sie dazu wie folgt vor:

- Der Nutzer gibt den Pfad zu einer Datei ein, die er drucken möchte. Speichern Sie den Pfad ab.
- Gibt der Nutzer `print` ein, sollen alle Druckaufträge nacheinander abgearbeitet werden. Geben Sie dazu z. B. aus: `Drucke Datei '/home/bug/Documents/Klassenliste.pdf'...`
- Die Nutzereingabe `quit` beendet das Programm.
- Bei der Eingabe von `clear` soll die Druckerwarteschlange geleert werden.

## **Bonusaufgabe 4: Klammerung korrekt?**

Schreiben Sie ein Programm, das für einen gegebenen String (z. B. durch eine Nutzereingabe) überprüft, ob die Klammerung korrekt ist. Beispiel:

- `"((3+5)*2)+(7*2)"` – richtig
- `"((5+2)*2()-5("` – falsch

Nutzen Sie eine geeignete Datenstruktur zur Lösung der Aufgabe. (*Tipp*: Sie wird auf diesem Arbeitsblatt erwähnt 😊)