

## Aufgabe 1

Hier ein paar Anwendungen für Stacks:

- History-Funktion (zurück/vor) in einem Browser
- Undo-Funktion in einem Anwendungsprogramm
- Überprüfung von Klammerausdrücken
- Umwandlung von Infix- in Postfix-Ausdrücke
- Berechnung von Postfix-Ausdrücken

## Aufgabe 2

Last In - First Out

## Aufgabe 3

```
1 from stack import Stack
2
3 s = Stack()      # []
4 s.push(5)       # [5]
5 s.push(2)       # [5, 2]
6 s.push(4)       # [5, 2, 4]
7 a = s.pop()     # [5, 2] a=4
8 s.push(7)       # [5, 2, 7]
9 s.push(1)       # [5, 2, 7, 1]
10 b = s.peek()   # [5, 2, 7, 1] b=1
11 s.push(8)      # [5, 2, 7, 1, 8]
12
13 print(a)        # => 4
14 print(b)        # => 1
15 print(s.size()) # => 5
```

## Aufgabe 4

- (a) Gemeinsamkeit: beide Methoden geben das oberste Element des Stacks zurück.
- (b) Unterschied: `s.pop()` entfernt das oberste Element, `s.peek()` entfernt es nicht.  
*to peek: forschend blicken, spähen, gucken*

## Aufgabe 5

```
1 class Stack:
2
3     def __init__(self):
4         self.items = []
5
6     def push(self, item):
7         self.items.append(item)
8
9     def pop(self):
10        return self.items.pop()
11
12    def peek(self):
13        return self.items[-1]
14
15    def is_empty(self):
16        return self.items == []
17
18    def size(self):
19        return len(self.items)
```

## Aufgabe 6

(a) 1, 5, 4, 6, 3, 2, 9, 8, 0, 7  
0 ~~1~~ ~~2~~ ~~3~~ ~~4~~ ~~5~~ ~~6~~ 7 ~~8~~ ~~9~~  
1 5 4 6 3 2 9 8            False

(b) 1, 3, 2, 0, 8, 7, 6, 5, 4, 9  
~~0~~ ~~1~~ ~~2~~ ~~3~~ ~~4~~ ~~5~~ ~~6~~ ~~7~~ ~~8~~ ~~9~~  
1 3 2 0 8 7 6 5 4 9        True

(c) 1, 5, 6, 4, 7, 9, 8, 3, 0, 2  
0 ~~1~~ 2 ~~3~~ ~~4~~ ~~5~~ ~~6~~ ~~7~~ ~~8~~ ~~9~~  
1 5 6 4 7 9 8 3            False

(d) 0, 2, 3, 7, 9, 8, 6, 5, 1, 4  
~~0~~ 1 ~~2~~ ~~3~~ 4 ~~5~~ ~~6~~ ~~7~~ ~~8~~ ~~9~~  
0 2 3 7 9 8 6 5            False

## Aufgabe 7

Erzeuge einen leeren Stack.

Durchlaufe den Term zeichenweise.

Ist das Zeichen eine öffnende Klammer, lege sie dem Stack ab.

Ist das Zeichen eine schliessende Klammer, prüfe, ob der Stack leer ist. Wenn ja, ist die Klammerung falsch → Stop. Wenn nein entferne eine öffnende Klammer vom Stack.

Prüfe am Schluss, ob der Stack leer ist. Wenn ja, dann ist die Klammerung richtig. Wenn nein, dann ist sie falsch.