

## Aufgabe 1

Hier ein paar Anwendungen für Stacks:

- History-Funktion (zurück/vor) in einem Browser
- Undo-Funktion in einem Anwendungsprogramm
- Überprüfung von Klammerausdrücken
- 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 7

f	r	o	m		s	t	a	c	k		i	m	p	o	r	t		S	t	a	c	k								
d	e	f		c	h	e	c	k	(	s	t	r	i	n	g	)	:													
		s	=	S	t	a	c	k	(	)																				
		f	o	r		c		i	n		s	t	r	i	n	g	:													
				i	f		c	=	=	'	(	'	:																	
							s	.	p	u	s	h	(	c	)															
				i	f		c	=	=	'	)	'	:																	
							i	f		s	.	i	s	_	e	m	p	t	y	(	)									
										r	e	t	u	r	n		F	a	l	s	e									
										s	.	p	o	p	(	)														
										r	e	t	u	r	n		s	.	i	s	_	e	m	p	t	y	(	)		

### Aufgabe 8

8 1 2 + - 6 \* 9 7 - /

8	1	2	3	5	6	30	9	7	2	15		
			+	-		*		-	/			

Resultat: 15