

**Merke 19.3** Ableitung von Summen und Vielfachen

Die Ableitung der Summe von Funktionen ist die Summe der Ableitungen:

$$(f(x) + g(x))' = f'(x) + g'(x) \quad \text{oder kurz} \quad (f + g)' = f' + g'$$

Die Ableitung eines (konstanten) Vielfachen einer Funktion ist das entsprechende Vielfache der Ableitung:

$$(a \cdot f(x))' = a f'(x) \quad \text{oder kurz} \quad (a \cdot f)' = a \cdot f' \quad \text{für beliebiges } a \in \mathbb{R}.$$

✂ **Aufgabe 19.9** Leiten Sie mit Hilfe der bereits bekannten Ableitungsregeln nach x ab:

a) $a(x) = x^{42}$

b) $b(x) = 5x^4$

c) $c(x) = x^3 + x^2$

d) $d(x) = 7$

e) $e(x) = x$

f) $f(x) = x \cdot x$

g) $g(x) = 1 + x + \frac{1}{2}x^2 + \frac{1}{6}x^3$

h) $h(x) = \frac{1}{x}$

19.5 Ableitung von Exponentialfunktionen

Im folgenden soll die Herleitung der Ableitung von Exponentialfunktionen skizziert werden.

Im ersten Schritt betrachten wir die konkrete Funktion $f(x) = 2^x$ und untersuchen deren Ableitung mit Hilfe des Differenzenquotienten, um zu zeigen, dass die Ableitung bis auf eine multiplikative Konstante die Funktion selbst ist:



Die genau gleichen Umformungen sind gültig, wenn man die Basis 2 durch eine allgemeine Basis positive Basis $a \neq 1$ ersetzt. Es gilt also für $g(x) = a^x$:

$$g'(x) = (a^x)' = a^x \cdot g'(0) \quad \text{wobei } g'(0) \in \mathbb{R}$$

Die Zahl $g'(0)$ direkt als Grenzwert zu bestimmen, übersteigt hier unsere Möglichkeiten. Anstatt $g'(0)$ für eine gegebene Basis a zu bestimmen, suchen wir eine Basis a so, dass $g'(0) = 1$ gilt (d.h. die Ableitung bei 0 möglichst einfach ist). Dies bedeutet, dass die Ableitung der Exponentialfunktion mit dieser Basis die Funktion selbst ist.



Listing 1: Python-Code zur Annäherung der speziellen Basis für Ableitung 1 bei $x = 0$.

```

1 from math import e
2 a_max = 3
3 a_min = 2
4 h = 1e-6
5 genauigkeit = 1e-8
6
7 def differenzenquotient(a):
8     return (a**h - 1) / h
9
10 while a_max - a_min > genauigkeit:
11     a = (a_max + a_min) / 2
12     if differenzenquotient(a) < 1:
13         a_min = a
14     else:
15         a_max = a
16     print(f"{a}")
17 print(f"{e}")
18 print(f"Fehler: {e - a}")

```