

Aufgabe 1:

- a) wahr: `complex(0,1) * complex(0,1) = (-1+0j)`
- b) wahr
- c) falsch: `{0, 1, 2, 3, 5, (1, 2, 3)}`
- d) wahr
- e) falsch (nur b)
- f) wahr
- g) falsch, nur Platzhalter
- h) falsch, nur 2 Nachstellen
- i) wahr
- j) wahr

Aufgabe 2:

1, 4, 6 und 7 entsprechen den von Python empfohlenen Namenskonventionen.

Anmerkung: 1 und 7 sind dennoch nicht zu empfehlen: 1 ist kurz und nichtssagend, 7 ist lang und somit erscheint der Status als guter Programmierstil fragwürdig.

Aufgabe 3:

1. `8/5 => 1.6 ist ein float`
2. `i=17+2`
`i%3`
`print(i)`
Hier ist der Fehler, dass zwei Operatoren hintereinander stehen: `/%`
Um den Fehler zu beheben, müsste man einen davon entfernen.
z.B. `"\"` entfernt => **19 ist ein int**
3. `"ja"*2+"ne"*2`
=> **jajanene ist ein str**
4. `a=3`
`a=a+4`
Variablennamen in der `print()` funktion ist falsch, richtig ist `a`
`print(a) => 7 ist ein int`
5. `1.0/0`
Hier wird durch 0 geteilt: `ZeroDivisionError: float division by zero.`
0 mit 1 ersetzt da ansonsten mit 0 geteilt wird:
`print(1.0/1) => 1.0 ist ein float`
6. `z="2.0"`
`11/int(z)`
Führt zu: `invalid literal for int() with base 10: '2.0'`
Da 2.0 ein float und kein int ist. Man müsste zu float parsen:
`z="2.0"`
`print(11/float(z)) => 5.5 ist ein float`
7. `str(7//2) => 3 ist ein int`
8. `3.0**(5-2) => 27.0 ist ein float`