

*variable* eine Variable mit Wert 1. Schreibe eine Funktion, die bei Aufruf die Variable *my\_variable* um 1 erhöht.

```

1 variable =
1 variable =
1 my_variable +
1 variable)
1 variable =
1 function() :
1 local_variable =
5 function() print(global_variable) print(local_variable)

```

```

1 variable =
1 function() :
1 print(global_variable)
1 function() print(global_variable)

```

```

1 glob-
1 ale
1 variable =
1 function() :
1 global_variable =
5 print(global_variable)
1 function() print(global_variable)

```

```

1 variable =
1 function() :
1 print(global_variable) global_variable =
5 function() print(global_variable)

```

*variable* eine globale Variable ist, oder eine lokale Variable, die zu fllig den gleichen Namen wie eine globale Variable trgt.

```

1 variable =
1 variable =
1 my_variable +
1 variable)

```

*variable* rechts vom Gleichheitszeichen steht, denkt Python, dass es sich um die globale Variable *my\_variable* handelt. Da *my\_a*

```

1 variable =
1 variable =
1 increment(my_variable) print(my_variable)

```

*Scope*

```

1 global
1 Scope
1 lo-
1 cal
1 Scope
1 list =
1 [1, 2, 3]
1 list.append(item)
1 list)

```

*list* nicht beschrieben wird, sondern nur das referenzierte Objekt verändert wird, erkennt Python dies nicht als Schreibzugriff.

```

1 file.txt") as my_file :
1 my_text =
1 my_file.read() print(my_text)

```

*Dateiob-  
jekt*

```

1 file gespeichert (der Variablenname ist beliebig) Die Methode read() liest den Text –
1 Inhalt der Datei, sodass er in einer Variable gespeichert werden kann. Sobald der eingerckte Block verlassen wird, wird die Datei
1 file.txt") as my_file :
1 my_lines =
1 my_file.readlines() for line in my_lines :
1 print(f"Theline reads :
1 line")

```

*Liste* – eine geordnete, unveränderliche Sammlung von Elementen. Jedes Element kann ein beliebiges Python-Objekt sein.