



CleanCode

Aurel, Joél, Noé, Eric



Weshalb CleanCode?



Bessere Lesbarkeit und
Verständlichkeit



Weniger Fehler und leichte



Erhöhte Wartbarkeit und
Erweiterbarkeit
Fehlersuche

Wichtigste Prinzipien



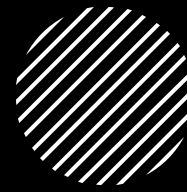
KLARHEIT UND
LESBARKEIT



SOLID-PRINZIPIEN



DRY



Klarheit und Lesbarkeit

- Namen sollten den Zweck klar beschreiben
- Keine Abkürzungen, die für Verwirrung sorgen

Schlechtes Beispiel:

```
def c(a, b):  
    return a * b * 3.14159  
  
r = 5  
h = 10  
v = c(r, h)  
print(v)
```

Gutes Beispiel:

```
def calculate_cylinder_volume(radius, height):  
    PI = 3.14159  
    return radius * radius * height * PI  
  
radius = 5  
height = 10  
cylinder_volume = calculate_cylinder_volume(radius, height)  
print(cylinder_volume)
```



SOLID- Prinzipien

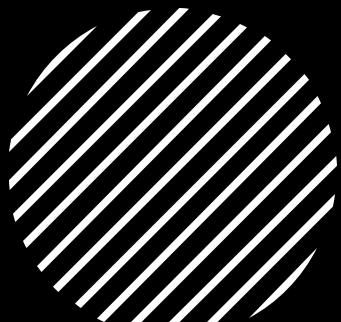


- **Single Responsibility:** Eine Klasse, eine Aufgabe
- **Open/Closed:** Erweiterbar, aber nicht veränderbar
- **Liskov Substitution:** Austauschbarkeit ohne Fehler
- **Interface Segregation:** Kleine, spezifische Interfaces
- **Dependency Inversion:** Abstraktionen statt Details



DRY (Don't Repeat Yourself)

- Vermeide Redundanzen
- Halte Code an einer Stelle konsistent und wartbar





Noch Fragen?

