

# Das KISS-Prinzip

- ✓ Halten Sie sich zu 100 % an Namenskonventionen.
- ✓ Allgemein sollte jede Form von Ausnahme im Programmierstil vermieden werden.  
Programmieren Sie „erwartbar“. Dazu zählt beispielsweise, dass Sie sich an gängige Notationen halten, die die meisten anderen Programmierer verwenden. So wird etwa die Zählvariable einer Schleife `i` lauten. Eine darin verschachtelte Schleife mit einer Zählvariablen wird diese `j` nennen, eine wieder darin verschachtelte `k` usw.
- ✓ Alle Zählvariablen von Schleifen werden mit dem Wert `0` initialisiert und um den Wert `1` erhöht.
- ✓ Variablen werden grundsätzlich initialisiert.  
Ausnahme: Sie wollen explizit einen undefinierten Zustand für eine gewisse Logik nutzen. Aber dann wird ja in JavaScript explizit mittels `var` oder `let` die „Initialisierung“ auf `undefined` vorgenommen.
- ✓ Globale Variablen werden – soweit es geht – vermieden.
- ✓ Vermeiden Sie es, den Datentyp einer Variable zu ändern.
- ✓ In Entscheidungsstrukturen, Schleifen etc. wird immer eine Blockanweisung notiert – auch wenn da nur eine einzige Anweisung notwendig ist und man auf die Notation eines Blocks verzichten könnte.
- ✓ Alle Anweisungen werden mit einem Semikolon beendet, auch wenn man darauf verzichten kann. Wenige Ausnahmen wie der Inkrement-Teil der `for`-Schleife sind aber zugelassen.
- ✓ Wenn Sie öffnende und schließende geschweifte Klammern benötigen, notieren Sie diese immer an der gleichen Stelle. Wenn Sie etwa einen Block mit einer geschweiften Klammer eröffnen, steht diese Klammer entweder in einer neuen Zeile oder direkt am Ende der vorherigen Zeile. Es ist – sofern es keine Richtlinien in einem Projekt gibt – gleichgültig, welche Variante Sie wählen. Aber nutzen Sie diese konsequent und ohne Ausnahme.
- ✓ Wiederholen Sie sich nicht. Das Konzept hat auch einen eigenen Namen (DRY – Don't repeat yourself). Erstellen und/oder nutzen Sie (getestete) Strukturen (Funktionen, Objekte, Methoden, etc.).