

Namenskonventionen

Neben den verbindlichen Regeln für Bezeichner gibt es in JavaScript (wie eigentlich in jeder Programmiersprache) Konventionen, die man bei Bezeichnern einhalten sollte. Die Bedeutung dieser Namenskonventionen kann gar nicht genug betont werden.

Die Einhaltung solcher Konventionen ist technisch nicht zwingend (was gerade Einsteigern den Nutzen nicht immer erkennen lässt), aber von sehr großem Vorteil. Denn es ist durchaus von Bedeutung, ob Sie in großen Projekten oder mit mehreren Personen zusammenarbeiten oder nur für sich alleine programmieren. Während eine Verletzung der Namenskonventionen im letzteren Fall nur geringe Probleme macht und im Zweifelsfall nur Sie selbst durch Ihre eigenen Codes nicht mehr durchblicken, kann dies große Projekte ganz zum Scheitern bringen oder zumindest Zeitabläufe verzögern und damit unnötige Kosten verursachen.

Deshalb die dringende Empfehlung, dass Sie Namenskonventionen zu 100 % einhalten sollten. Gerade in JavaScript, das durch seine Freiheiten zwar sehr schlank und flexibel ist, aber ohne strenge Einhaltung dieser Konventionen – gerade für Einsteiger – keinerlei Halt und Orientierung gibt. Mit der Beachtung der Namenskonventionen hingegen sind Syntaxstrukturen in JavaScript eindeutig festgelegt und Quellcode oft alleine aufgrund der gewählten Schreibweise vom Bezeichner eindeutig klar. Es lassen sich so beispielsweise durch eine vereinheitlichte und konsequente Wahl von Groß- und Kleinschreibung sehr wichtige Metainformationen festlegen.

Oder noch einmal anders ausgedrückt – beim Einstieg erscheint die strenge Beachtung der Namenskonventionen vielleicht lästig, aber es zahlt sich aus und erleichtert auf längere Sicht die Programmierung, das Verständnis und später die Pflege von Quellcode ungemein.

Die folgenden Konventionen haben sich in JavaScript etabliert. Beachten Sie, dass es in einigen Zusammenhängen andere Regeln geben kann und insbesondere Microsoft die üblichen JavaScript-Konventionen für sein Habitat an einigen Stellen verändert hat:

- ✓ Bezeichner von Eigenschaften, Variablen und Funktionen/Methoden in JavaScript beginnen immer mit einem Kleinbuchstaben. Allerdings kann es begründete Ausnahmesituationen geben, wo man explizit durch ein abweichendes Großschreiben bestimmte Metainformationen vermitteln will. Diese Abweichungen sollten aber sehr selten erfolgen und wirklich sehr gut gerechtfertigt sein.
- ✓ Bei längeren, zusammengesetzten Bezeichnern verwendet man die Camelnotation (auch Höckernotation genannt), bei der jedes neue Teilwort mit einem Großbuchstaben an das vorherige Teilwort angefügt wird. Alternativ kann man auch den Unterstrich zum Zusammenfügen verwenden, aber die Schreibweise ist in JavaScript eher unüblich. Nur bei Konstanten ist sie üblich, denn diese werden vollständig großgeschrieben und damit kann man nicht mit Groß- und Kleinschreibung logische Trennungen vornehmen.
- ✓ Deutsche Umlaute, im Prinzip erlaubte Sonderzeichen außer dem Unterstrich etc. unterbleiben beim Bezeichner.
- ✓ Klassen/Prototypen werden immer mit einem Großbuchstaben begonnen und dann kleingeschrieben. Es gilt weiter die Camelnotation.
- ✓ Klassen/Prototypen werden in Singularform notiert.
- ✓ Konstanten (und nur diese Elemente) werden vollständig großgeschrieben.