

4. **コンセンサスありきの省略記法**

\$記号は一種のクセであるので、jQueryにおける\$記号は、ともすればプログラマに不快感を与えるものになりかねない。また、他のライブラリの\$記号の定義とバッティングして面倒なことになるリスクもはらんでいる。

しかしながらjQueryでは、jQueryを意味する\$記号が好まれて使われている。なぜか？

その答えは、「jQueryとはそういうものだ」という確かなコンセンサスがそこにあるからだ。なので、プロジェクトにjQueryを持ち込んだ時点で、\$を汚染するようなコードやライブラリを組み込んではいけない。これで\$記号の定義バッティングの問題も解決だ。

5. **コーディングスタイル = 思想**

jQueryにおける省略としての\$記号から見えてくるのは、コーディングスタイルとはすなわち思想であるということだ。

コーディングスタイルの比較は、プログラマの間の宗教戦争の定番ネタだ。あなたもプログラマなら、気に食わないスタイルのコードを見てストレスを感じたことがあるだろう。

たとえば、わたしは以下のようなコードを好んで書く。

[TK: ここにソースコードを]

\$("#name")と何度も書くのがブサイク&指がこむら返りしそうなので、\$nameというオブジェクトにしてしまうわけだ。そして、テキストボックスの内容はnameというオブジェクトにする。

このコードは、嫌いな人には嫌いなコードだと思う。\$記号を変数の先頭に付与することで、これはDOMオブジェクトであることを明示しているわけであるが、一種のハンガリアン記法とみなすこともできるからだ。

一方、\$記号を使わずに等価なコードを書くと以下のようになる。

[TK: ここにソースコードを]

先に\$記号を使ったコードを見てみると、\$記号を使わないコードが冗長でクドく思えないだろうか？\$記号を使ったコードのほうがスッキリしていて読みやすく感じないだろうか？それは、両スタイルの比較を通して、あなたとわたしの間でコーディングスタイルのコンセンサスができているからだ。つまり、先頭に\$記号のついている変数はDOMオブジェクトを表しているよという暗黙の同意だ。

6. **設定より規約**

以上の考え方は、Ruby on Railsの原理である「設定より規約」に似ている。コーディング規約をしっかりと決め、背景にある考え方をプログラマ同士で共有することで、コーディング時のストレスを減らすことができる。

規約という「縛り」があるからには、そこには必ず理由がある。すべての場合に対応できる規約を決めることは現実的ではないが、プログラマ同士、ある程度のコンセンサスをとっておくことは大切である。