

Webプログラマのための JavaScript 言語の改良

柏倉 歩[†] 大谷 真[‡]

湘南工科大学[†] 湘南工科大学[‡]

1. はじめに

JavaScript はクラスレスのオブジェクト指向言語であるとともに、第 1 級関数をサポートした関数型言語である。グローバル変数や変数スコープの問題、セミコロン挿入や==の一貫性上の問題などが指摘されているものの、ECMAScript5 による標準化とその実装も進み充実した言語に成長している。フロントエンドプログラミングを中心に世界的に広く使われており、最近では Node.js にみられるようにサーバプログラミングにも使われ始めている。JavaScript の素晴らしい点は、その技術的特徴だけでなく、これまで一部の専門家しか使っていなかった第 1 級関数を使った関数型プログラミングを一般プログラマが普通に使うまでに大衆化させた点にある。実際に一般の Web プログラムが何の違和感なく関数オブジェクトを変数に代入したり関数の引数に指定している。

本研究ではこれまで JavaScript ではオウン変数を使うのが難しいという問題点に関して、JavaScript の柔軟さを活かすことのできるプロトタイプベースプログラミングで解決するという立場で新たな文法と通常の JavaScript に変換するプリプロセッサの開発を行ってきた。この方法により、オブジェクトを利用した方法のようにオブジェクトを自分で管理する必要やオウン変数がパブリックになってしまうような問題もなく、簡易にオウン変数を使えるようにすることができた[1]。

ここで、このオウン変数を複数の関数間で共有できないかという課題が浮上した。ローカルで関数呼び出しを超えて値を持続するというオウン変数を複数の関数間で共有して使用したい場合が出てくる。このような機能を JavaScript で実現するにはクロージャ機能を入れ子にし、関数を返す関数を返すというような方法で解決することができる。しかし、クロージャ機能を利用することは多くの Web プログラマには技術的に難しいため、通常はグローバル変数を宣言することで済ませていることが多い。

本研究では現在上記の問題点を解決するため、一般 Web プログラマにも分かりやすい形でオウン変数を複数の関数間で共有して使用できるようにシンタックスと実現方法を考案中である。

2. これまでの結果：オウン変数機能

我々は既に[1]において JavaScript 拡張文法オウンブロックを導入し、オウン変数をクロージャに変換するアルゴリズムを考案している。

Improvement of JavaScript language for Web programmers

[†]Ayumu Kashiwakura, Shonan Institute of Technology

[‡]Makoto Oya, Shonan Institute of Technology

2.1 オウン変数

オウン変数とは、ALGOL60 で使用された用語であり、C 言語のスタティック変数に類似している。つまり、関数呼び出しを超えて値の持続する変数のことである。これは従来の JavaScript には存在しない。

2.2 オウン変数機能の追加

通常に関数定義に新たな文法オウンブロックを追加し、"関数を生成する関数"という複雑な概念を使用せず、オウン変数が定義できるようにした。使い易さを考え、関数呼び出しは通常の JavaScript と同じとした。拡張文法を図 1 に示す。

```
function 関数名([仮引数リスト]) {
  @own
  var オウン変数宣言 1;
  var オウン変数宣言 2;
  ...
  @end
  文
}
```

オウンブロック
と呼ぶ

図 1 オウンブロックの追加

オウンブロック内で宣言された変数はローカルスコープにあるものの、関数が返った後も持続し、次の関数呼び出し時に値が引き継がれる。なお、オウンブロックには変数以外に関数、オブジェクトを宣言することもできる。

3. オウン変数の共有化の提案

3.1 オウン変数の共有化

複数の関数間でオウン変数を共有して使用したい場合がある。クラスベース言語の場合は同じクラスのインスタンスを作ることによって対処できるが、クラスベースでないオブジェクト指向言語である JavaScript では一般的にプロトタイプ継承で対処する。しかし、後述のような問題もあり、Web プログラマに分かり易い手段はない。そこで共有オウン変数という新たな概念を提案する。

3.2 共有化の有用性

他の関数で使用しているローカル変数の値そのものを共有し、使用できることが望まれている。ある関数が使用しているオウン変数を別の関数が使用するというものである。クリック処理関数内や Ajax コールバック関数内で使用しているオウン変数を別の関数からアクセスし、その値を条件分岐や関数引数などに利用するなど、いくつか考えられる。また、値そのものの共有ではなく、コードの再利用としての共有化ということも考えられる。これは同じ変数・関数を複数回記述するのではなく、どこか 1 ヶ所に書いたものを短い文で使用できるようになればコードの単純化になる。

3.3 考えられる解決方法

まず考えられる方法であり、多くの Web プログラマーが使用している方法としては、グローバル変数を利用する方法である(図 2)。

```
var sharedVar = 0;
function X() {
  sharedVar += 1;
}
function Y() {
  alert(sharedVar);
}
X(); Y();
```

図 2 グローバル変数による解決

次に考えられる方法は、プロトタイプ継承を利用する方法がある(図 3)。この方法であれば、値そのものを共有することが可能である。

```
function shareConst() {}
shareConst.prototype.sharedVar = 0;
var share = new shareConst();
function X() {
  share.sharedVar += 1;
}
function Y() {
  alert(share.sharedVar);
}
X(); Y();
```

図 3 プロトタイプ継承による解決

そして、クラスインスタンスを利用する方法も考えられる(図 4)。これはコード再利用を可能とする方法となる。

```
function shareConst() {
  this.sharedVar = 0;
}
var share = new shareConst();
function X() {
  share.sharedVar += 1;
}
function Y() {
  alert(share.sharedVar);
}
X(); Y();
```

図 4 クラスインスタンスによる解決

最後にクロージャを利用する方法が挙げられる(図 5)。

```
function shareFunc() {
  var sharedVar = 0;
  function X() {
    sharedVar += 1;
  }
  function Y() {
    alert(sharedVar);
  }
  return [X, Y];
}
var share = shareFunc();
var X = share[0](); var Y = share[1]();
X(); Y();
```

図 5 クロージャによる解決

3.4 解決法の利点・欠点

まず、グローバル変数による解決はプログラムの保守性や拡張性などに問題があるため、適切ではない。

次にプロトタイプ継承による解決にはプロトタイプに割り当てられたインスタンスを利用しているため、オウ

ン変数以外の共有して欲しくない変数の値までアクセス可能になってしまうという問題が存在する。そして、クラスインスタンスによる解決では変数の共有化という問題は解決されるものの、共有された変数がローカルではなくパブリックになる問題が存在する。最後に、クロージャによる解決ではローカルな変数・関数の共有化という問題を解決することが可能である。クロージャによる解決であれば、値そのものの共有とコード再利用、両方が可能であるのではないかと考える。

3.5 有力と考えられる解決法

以上のことより、本研究で有力と考えられる解決法はクロージャを利用する解決方法である。

そして本研究で提案するのは 2.2 の仕様を拡張し、「@own」の後にオウンブロック名を書けるようし、その名前を他の関数のオウンブロック内で書くと名前付きオウンブロック内の変数をローカルなオウン変数として共有して使用できるようにするという考え方である。

例えば図 6 のようなシンタックスを考えている。

```
function 関数名 1([仮引数リスト]) {
  @own(オウンブロック名)
  var オウン変数宣言 1;
  var オウン変数宣言 2;
  ...
  @end
  文
}
function 関数名 2([仮引数リスト]) {
  @own(オウンブロック名)
  ...
}
```

図 6 提案シンタックス

ただし、現状では 1 つの関数が複数のオウンブロックを共有することは JavaScript の言語仕様では解決不能であるものと考えている。これは、複数のオウンブロックを共有できるとなると、関数 X がオウンブロック A と B を共有し、関数 Y がオウンブロック B と C を共有するなどのことが可能になってしまう。これを実現できるコーディングを現在のところ本研究では見つけられていないためである。

また、あるオウンブロックが別のオウンブロックを共有して利用することも望まれるが、このような場合どうするのか、など問題が存在していることも確かである。

4. 考察と今後の課題

本研究では一般の Web プログラマーでも、容易に宣言し扱えるようになったオウン変数を複数の関数間で共有して使用できるようにすることを提案した。JavaScript におけるオウン変数の追加と共有化によりグローバル変数を利用せず、オウン変数を簡易に利用でき、その変数を複数関数間で共有できる。

これを実現する手法の実装が今後の課題である。

参考文献

[1] 柏倉,大谷,JavaScript へのオウン変数機能と関数クロージャ機能の追加,FIT2015, pp.109-112,2015