

ソフトウェア開発 第 14 回目授業

平野 照比古

2016/1/15

jQuery ファイルの違い

- jQuery のソースファイルには jQuery-<Version No>.js と jQuery-<Version No>.min.js の 2 種類が存在
- jQuery-<Version No>.min.js は jQuery-<Version No>.js からコメントや、空白を取り除き、変数名を短いものに置き換えるという操作を行って、ファイルサイズを小さくしたもの

jQuery のソースコード (1)

```
/*!  
 * jQuery JavaScript Library v1.11.3  
 * http://jquery.com/  
 *  
 * Includes Sizzle.js  
 * http://sizzlejs.com/  
 *  
 * Copyright 2005, 2014 jQuery Foundation, Inc. and other contributors  
 * Released under the MIT license  
 * http://jquery.org/license  
 *  
 * Date: 2015-04-28T16:19Z  
 */  
}
```

jQuery のソースコード (2)

```
(function( global, factory ) {  
  if ( typeof module === "object" && typeof module.exports === "object" ) {  
    // For CommonJS and CommonJS-like environments where a proper window is present,  
    // execute the factory and get jQuery  
    // For environments that do not inherently possess a window with a document  
    // (such as Node.js), expose a jQuery-making factory as module.exports  
    // This accentuates the need for the creation of a real window  
    // e.g. var jQuery = require("jquery")(window);  
    // See ticket #14549 for more info  
    module.exports = global.document ?  
      factory( global, true ) :  
      function( w ) {  
        if ( !w.document ) {  
          throw new Error( "jQuery requires a window with a document" );  
        }  
        return factory( w );  
      };  
  } else {  
    factory( global );  
  }  
  // Pass this if window is not defined yet  
(typeof window !== "undefined" ? window : this, function( window, noGlobal ) {
```

jQuery のソースコード (3)

```
// Can't do this because several apps including ASP.NET trace
// the stack via arguments.caller.callee and Firefox dies if
// you try to trace through "use strict" call chains. (#13335)
// Support: Firefox 18+
//
var deletedIds = [];
var slice = deletedIds.slice;
var concat = deletedIds.concat;
var push = deletedIds.push;
var indexOf = deletedIds.indexOf;
var class2type = {};
var toString = class2type.toString;
var hasOwn = class2type.hasOwnProperty;
var support = {};
```

jQuery のソースコード (tiny)

```
var
  version = "1.11.3",
  // Define a local copy of jQuery
  jQuery = function( selector, context ) {
    // The jQuery object is actually just the init constructor 'enhanced'
    // Need init if jQuery is called (just allow error to be thrown if not
    return new jQuery.fn.init( selector, context );
  },
  // Support: Android<4.1, IE<9
  // Make sure we trim BOM and NBSP
  rtrim = /^[\s\uFEFF\xA0]+|[\s\uFEFF\xA0]+$|$/g,
  // Matches dashed string for camelizing
  rmsPrefix = /^-ms-/ ,
  rdashAlpha = /-([\da-z])/gi,
  // Used by jQuery.camelCase as callback to replace()
  fcamelCase = function( all, letter ) {
    return letter.toUpperCase();
  };
```

jQuery 1.12.2.js のソース

- このリストでは詳細なコメントが付いていて比較的読みやすい。
- コメントを見るとクロスブラウザ対策が行われていることも見て取れる。
- ソースコードの最小化のテクニックとしてグローバルなオブジェクトを短い変数名で置き換える
- 冒頭の定義された関数の引数に、関数を渡している。その関数の仮引数は window となっている。これにより、開発時はわかりやすい変数名が使えるメリットがある。

短縮化のリスト

```

/*! jQuery v1.11.3 | (c) 2005, 2015 jQuery Foundation, Inc. | jquery.org/license */
!function(a,b){"object"===typeof module&&"object"===typeof
  module.exports?module.exports=a.document?b(a,!0):function(a){if(!a.document)throw
  new Error("jQuery requires a window with a document");
  return b(a)}:b(a)}
  ("undefined"!==typeof window?window:this,function(a,b){
    var c=[],d=c.slice,e=c.concat,f=c.push,g=c.indexOf,h={},i=h.toString,
    j=h.hasOwnProperty,k={},l="1.11.3",
    m=function(a,b){return new m.fn.init(a,b)},
    n=/^[\s\uFEFF\xA0]+|[\s\uFEFF\xA0]+$/g,o=/^-ms-/ ,
    p=/-([\da-z])/gi,q=function(a,b){return b.toUpperCase()};

```


短縮化の方法

- 各関数内で変数名は 1 文字から始めている。
- 関数の仮引数も a から付け直している。
- JavaScript の固有の関数は当然のことながら変換されていない。
- このライブラリーは一つの関数を定義して、その場で実行している。
14 行目の `function()` の前に (がついている。
- 短縮化されたコードではこの部分が `!function()` となっている。関数オブジェクトを演算の対象とすることで実行する。
- そのほかにもキーワード `true` の代わりに `!0` としている。
- `if(){}else{}` 構文は `?:` に置き換えている。

JavaScript ファイルの短縮化ツール

JavaScript の短縮化ツールとして、Google が提供する Closure Compiler を紹介する。このサービスは次のように説明されている。

The Closure Compiler is a tool for making JavaScript download and run faster. It is a true compiler for JavaScript. Instead of compiling from a source language to machine code, it compiles from JavaScript to better JavaScript. It parses your JavaScript, analyzes it, removes dead code and rewrites and minimizes what's left. It also checks syntax, variable references, and types, and warns about common JavaScript pitfalls.

Closure Compiler の使い方

使い方については次のように書かれている。

You can use the Closure Compiler as:

- *An open source Java application that you can run from the command line.*
- *A simple web application.*
- *A RESTful API.*

To get started with the compiler, see "How do I start" below.

ここでは 2 番目にある Web アプリケーションで行う。

Closure Compiler の Web サイト

サイトは <http://closure-compiler.appspot.com/home>

The screenshot shows the Closure Compiler web interface in the Opera browser. The address bar shows the URL <http://closure-compiler.appspot.com/home>. The page title is "Closure Compiler". The interface includes a "Add a URL:" field with a dropdown and an "Add" button. Below this, there are radio buttons for "Optimization:" (Whitespace only, Simple, Advanced) and checkboxes for "Formatting:" (Pretty print, Print input delimiter). A "Compile" button and a "Reset" link are also present. The main code area contains the following code:

```
// ==ClosureCompiler==
// @compilation_level SIMPLE_OPTIMIZATIONS
// @output_file_name default.js
// ==/ClosureCompiler==

// ADD YOUR CODE HERE
function hello(name) {
  alert('Hello, ' + name);
}
hello('New user');
```

On the right side, there are sections for "Original Size:", "Compiled Size:", and a table with buttons for "Compiled Code", "Warnings", "Errors", and "POST data".

短縮化の例

使用するファイルは9回目の授業で用いた event.js

Opera

Closure Compiler Service

closure-compiler.appspot.com/home#code%3Dwindow.onload%2520%253D%2520function()%2520%2

Closure Compiler

Add a URL: Add

Example: <http://www.example.com/bigfile.js>

Optimization: ☐ Whitespace only ☒ Simple ☐ Advanced

[Which optimization is right for my code?](#)

Formatting: ☐ Pretty print ☐ Print input delimiter

```
Squares.children[0].style.background = "red";
Squares.children[1].style.background = "yellow";
Squares.children[2].style.background = "blue";
Select.style.fontSize = "30px";
Squares.addEventListener("click",function(E){
  Inputs[0].value = E.clientX;
  Inputs[1].value = E.clientY;
  Inputs[2].value = E.pageX;
  Inputs[3].value = E.pageY;
  Inputs[4].value = window.pageXOffset;
  Inputs[5].value = window.pageYOffset;
  var R = E.target.getBoundingClientRect();
  Inputs[6].value = E.pageX-R.left;
  Inputs[7].value = E.pageY-R.top;
  ColorName.value = E.target.style.background;
  lastClicked = E.target;
},false);
Select.addEventListener("change", function(){
  lastClicked.style.background = Select.value; },false);
```

Compilation was a success!

Original Size: 544 bytes gzipped (1.53KB uncompressed)

Compiled Size: 466 bytes gzipped (1.06KB uncompressed)

Saved 14.34% off the gzipped size (31.06% without gzip)

The code may also be accessed at [default.js](#).

Compiled Code	Warnings
Errors	POST data

```
window.onload=function(){var
b=document.getElementById("Squares"),e=
document.getElementById("select"),g=document.
getElementById("colorName"),f=document.
getElementById("radio"),h=document.g
getElementById("Set"),c=document.getElemen
tsByClassName("click"),d=b.children[1];l
.children[0].style.background="red";b.cl
ildren[1].style.background="yellow";b.cl
ildren[2].style.background="blue";e.styl
e.fontSize="30px";b.addEventListener("c
```

短縮化の結果 (1)

画面の右のほうで、元のファイルの大きさが 1.53KB であったのに対し、短縮化の結果が 1.06KB となったことがわかる。

短縮化の詳細

```
window.onload=function(){var b=document.getElementById("Squares"),
    e=document.getElementById("select"),
    g=document.getElementById("colorName"),
    f=document.getElementById("radio"),
    h=document.getElementById("Set"),
    c=document.getElementsByClassName("click"),
    d=b.children[1];
    b.children[0].style.background="red";
    b.children[1].style.background="yellow";
    b.children[2].style.background="blue";
    e.style.fontSize="30px";
```

短縮化の結果 (2)

```
b.addEventListener("click",function(a){
  c[0].value=a.clientX;
  c[1].value=a.clientY;
  c[2].value=a.pageX;
  c[3].value=a.pageY;
  c[4].value=window.pageXOffset;
  c[5].value=window.pageYOffset;
  var b=a.target.getBoundingClientRect();
  c[6].value=a.pageX-b.left;
  c[7].value=a.pageY-b.top;
  g.value=a.target.style.background;
  d=a.target
},!1);
```


短縮化の結果 (3)

```
e.addEventListener("change",function(){
    d.style.background=e.value},!1);
f.addEventListener("click",function(a){
    alert(a.target.tagName);
    "DIV"===a.target.tagName&&(a.target.firstChild.checked=!0);
    console.log("----"+f.value);
    d.style.background=f.querySelector("input:checked").value
},!1);
h.addEventListener("click",function(){
    d.style.background=g.value},!1)};
```

- if 文であったものが論理式の&&で置き換えられている。
- document.getElementById などそのまま短縮化、共通化されていない。

document.getElementById を減らす

単純に関数を置き換えただけではうまくいかない。

Opera

Closure Compiler Service

closure-compiler.appspot.com/home#code%3Dwindow.onload%2520%253D%2520function()%2520%257B%2520

Closure Compiler

Add a URL:

Example: <http://www.example.com/bigfile.js>

Optimization: ☐ Whitespace only ☒ Simple ☐ Advanced

[Which optimization is right for my code?](#)

Formatting: ☐ Pretty print ☐ Print input delimiter

[Reset](#)

```
// ==ClosureCompiler==
// @output_file_name default.js
// @compilation_level SIMPLE_OPTIMIZATIONS
// ==/ClosureCompiler==

window.onload = function() {
  function getElm(N){
    return document.getElementById(N);
  }
  var Squares = getElm("Squares");
  var Select = getElm("select");
  var ColorName = getElm("colorName");
  var Radio = getElm("radio");
  var Set = getElm("Set");
  var Inputs = document.getElementsByClassName("click");
  var Squareschildren = Squares.children;
  var lastClicked = Squareschildren[1];
  function SetBackground(Pos, Color) {
    Squareschildren[Pos].style.background = Color;
  }
  var v = "value";
  SetBackground(0, "red");
}
```

Compilation was a success!

Original Size: 605 bytes gzipped (1.57KB uncompressed)

Compiled Size: 469 bytes gzipped (1.04KB uncompressed)

Saved 22.48% off the gzipped size (33.92% without gzip)

The code may also be accessed at [default.js](#).

Compiled Code	Warnings
Errors	POST data

```
window.onload=function(){var
g=document.getElementById("Squares"),e=document.getElementById("select"),h=document.getElementById("colorName"),f=document.getElementById("radio"),k=document.getElementById("Set"),b=document.getElementsByClassName("click"),c=g.children,d=c[1];c[0].style.background="red";c[1].style.background="yellow";c[2].style.background="blue";e.style.fontSize="30px";g.addEventListener("click",function(a){b[0].value=a.clientX;b[1].value=a.clientY;b[2].value=a.pageX;b[3].value=a.pageY;b[4].value=window.pageYOffset;b[5].value=window.pageYOffset;var c=a.target.getBoundingClientRect();b[6].
```

解説

ラップした関数は消えていて、もとの `document.getElementById` に戻っている。

これを避けるためには関数を定義してその場で実行する関数の引数に `document` と `"getElementById"` を渡すことで解決できる。

ソースコードの書き直し (1)

```
var Squares, Select, ColorName, Radio, Set;  
(function(document,getElementById){  
    Squares    = document[getElementById]("Squares");  
    Select      = document[getElementById]("select");  
    ColorName   = document[getElementById]("colorName");  
    Radio       = document[getElementById]("radio");  
    Set         = document[getElementById]("Set");  
})(document,"getElementById");
```

ソースコードの書き直し (2)

Opera

Closure Compiler Service

closure-compiler.appspot.com/home#code%3Dwindow.onload%2520%253D%2520function()%2520%257B%2520

Closure Compiler

Add a URL: Add

Example: <http://www.example.com/bigfile.js>

Optimization: ☐ Whitespace only ☒ Simple ☐ Advanced

[Which optimization is right for my code?](#)

Formatting: ☐ Pretty print ☐ Print input delimiter

```

var Inputs = document.getElementsByClassName("click");
var lastClicked = Squares.children[1];
Squares.children[0].style.background = "red";
Squares.children[1].style.background = "yellow";
Squares.children[2].style.background = "blue";
Select.style.fontSize = "30px";
Squares.addEventListener("click",function(E){
  Inputs[0].value = E.clientX;
  Inputs[1].value = E.clientY;
  Inputs[2].value = E.pageX;
  Inputs[3].value = E.pageY;
  Inputs[4].value = window.pageXOffset;
  Inputs[5].value = window.pageYOffset;
  var R = E.target.getBoundingClientRect();
  Inputs[6].value = E.pageX-R.left;
  Inputs[7].value = E.pageY-R.top;
  ColorName.value = E.target.style.background;
  lastClicked = E.target;
},false);
Select.addEventListener("change", function(){
  lastClicked.style.background = Select.value; },false);
Radio.addEventListener("click", function(E){
  alert(E.target.tagName);
  if(E.target.tagName === "DIV") {
    E.target.firstChild.checked = true;
  }
});

```

Compilation was a success!

Original Size: 589 bytes gzipped (1.65KB uncompressed)

Compiled Size: 486 bytes gzipped (1.02KB uncompressed)

Saved 17.49% off the gzipped size (38.18% without gzip)

The code may also be accessed at [default.js](#).

Compiled Code	Warnings
Errors	POST data

```

window.onload=function(){var c,d,g,e,h;
(function(a,b){c=a[b]("Squares");d=a[b]
("select");g=a[b]("colorName");e=a[b]
("radio");h=a[b]("Set");})(document
,"getElementById");var
b=document.getElementsByClassName("click
"),f=c.children[1];c.children[0].style.b
ackground="red";c.children[1].style.back
ground="yellow";c.children[2].style.back
ground="blue";d.style.fontSize="30px";c.
addEventListener("click",function(a){b[0
].value=a.clientX;b[1].value=a.clientY;b
[2].value=a.pageX;b[3].value=a.pageY;b[4
].value=window.pageXOffset;
b[5].value=window.pageYOffset;var
c=a.target.getBoundingClientRect();b[6].
value=a.pageX-c.left;b[7].value=a.pageY-
c.top;g.value=a.target.style.background;
f=a.target;f.addEventListener("chan
ge",function(){f.style.background=valu

```

ソースコードの書き直し (3)

コンパイル後の結果が 1.02KB とわずかに小さくなっていることがわかる。
短縮化を効率よくするためには元のプログラムも考えて作らないといけない。

注意

この書き直しで `document.getElementById` だけを渡せば十分と思われるかもしれないが、これは関数オブジェクトとなり、関数内での `this` が `document` にならないので、実行時にエラーが生ずる。
与えられた関数を別のオブジェクトのメソッドとして実行するために、`call` メソッドを用いる。

```
(function(document,getElementById){  
    Squares    = getElementById.call(document,"Squares");  
    Select     = getElementById.call(document,"select");  
    ColorName  = getElementById.call(document,"colorName");  
    Radio      = getElementById.call(document,"radio");  
    Set        = getElementById.call(document,"Set");  
})(document,document.getElementById);
```