

練習問題の解答

2章

練習 2-1

識別子として有効なものは次のものです。

```
$someVariable  
_someVariable  
some_variable
```

以下のものは有効ではありません。

```
1Variable // 数字で始まっています  
som&#232;variable // 「&」などは使えません  
function // functionは予約語ですから識別子としては使えません  
.someVariable // 「.」は使えません  
some*variable // 「*」は使えません
```

練習 2-2

この本では以下のようにすることをおすすめしています(独自の方式を使ってもかまいませんが、一貫性を保つことが大切です)。

```
var someMonth; // キャメルケースにします  
function getTheCurrentMonth // 関数名は動詞を使います  
CURRENT_MONTH // すべて大文字にします  
var summerMonths // 配列には複数の項目が記憶されるので複数形にします。  
mylibraryDoSomethingFunctionName  
// ライブラリ名を先頭につけ、その後に動詞を用いた関数名にします
```

練習 2-3

次のように「!」の前に「\」を書く必要があります。

```
var someString = 'Who once said, "Only two things are infinite, the
universe and human stupidity, and I\'m not sure about the former."'
```

練習 2-4

問題の書き方が曖昧でしたが、"432.54" という文字列の整数部分を取り出し、それを 16 進数や 8 進数として解釈するのならば、次のようにすればよいでしょう。

```
var floatValue = 432.54;
var intValue = parseInt(floatValue);
var hexValue = parseInt(floatValue, 16); // 16 進数として解釈
var octValue = parseInt(floatValue, 8); // 8 進数として解釈
```

練習 2-5

まず、次の例のように引数に渡す段階で、宣言も値の代入も行っていない引数を指定した場合は、JavaScript のエラーになります。

```
function nullOrUndefined(someMonth) {
  ...
}

nullOrUndefined(abc);
```

次の例のように変数を宣言して、値が入っていない場合は、`undefined` との比較も `null` との比較も `true` になり、両者を区別することはできません。

```
function nullOrUndefined(someMonth) {
  if (someMonth == undefined) {
    ... // A
  }
  if (someMonth == null) {
    ... // B
  }
}

var abc;
nullOrUndefined(abc);
```

また、次の例のように変数を宣言して、`null` あるいは `undefined` の値を入れた場合も、`undefined` との比較も `null` との比較も `true` になり、両者を区別することはできません。

```
function nullOrUndefined(someMonth) {
  if (someMonth == undefined) {
    ... // A
  }
  if (someMonth == null) {
    ... // B
  }
}
```

```
    }  
}  
  
var abc = null;  
nullOrUndefined(abc);
```

3章

練習 3-1

次のようにすれば `resultOfComp` の値は 8 になります

```
var valA = 37;  
var valB = 3;  
var valC = 18;  
var resultOfComp = (valA - valB) % 3 / 2 * (4 + valC) - 3;
```

練習 3-2

たとえば次のようにします。

```
switch(val) {  
  case 'one' :  
  case 'two' :  
    result = 'OK';  
    break;  
  case 'three' :  
    result = 'OK2';  
    break;  
  default :  
    result = 'NONE';  
}
```

練習 3-3

次のような条件式を書けばよいでしょう。

```
if ((varOne == 33) && (varTwo <= 100) && (varThree > 0)) {  
  ...  
}
```

練習 3-4

たとえば、次のコードでよいでしょう(問題の「数字」は「整数」としておいた方がよかったですね)。

```
for (var i = 11; i < 20; i++) {  
  document.writeln(i + "<br />");  
}
```

練習 3-5

たとえば、次のコードでよいでしょう。

```
for (var i = 19; i > 10; i--) {  
    document.writeln(i + "<br />");  
}
```

4 章

練習 4-1

次のようにすれば、単語 "fun" が置換されます。

```
var str = "The fun of functions is that they are functional."  
var regexp = /\bfun\b/g;  
var result = str.replace(regexp, "power");
```

練習 4-2

次のコードのように日付に7を足せば大丈夫です。もちろん `getHours` を用いて `7*24` 時間を足しても同じことになります。

```
var dtNow = new Date();  
var date = dtNow.getDate();  
date += 7;  
dtNow.setDate(date);  
document.writeln(dtNow.toString());
```

練習 4-3

切り下げるには `Math.floor` を、切り上げるには関数 `Math.ceil` を用います。たとえば次のようなコードで確かめられます。

```
var a = 34.44;  
document.writeln(Math.floor(a) + "<br />");  
document.writeln(Math.ceil(a) + "<br />");
```

練習 4-4

次のようなコードでよいでしょう。

```
var str = "apple.orange-strawberry,lemon-.lime";  
var regexp = /[.,]/g;  
var result = str.replace(regexp, ',');  
var arrayValues = result.split(',');  
for (var i = 0; i < arrayValues.length; i++) {  
    document.writeln(arrayValues[i] + "<br />");  
}
```

「-」でも区切ることにするのならば、次のように「-」も加えます。ただし、そうすると「-.」のところで空白ができるので、その場合は印刷しないようにしてみました。

```
var str = "apple.orange-strawberry,lemon-.lime";
var regexp = /[-.]/g;
var result = str.replace(regexp, ',');
var arrayValues = result.split(',');
for (var i = 0; i < arrayValues.length; i++) {
    if (arrayValues[i] != "") {
        document.writeln(arrayValues[i] + "<br />");
    }
}
```

5章

練習 5-1

宣言的関数、無名関数、リテラル関数の3種類です。それぞれの性質については、本文を参照してください。

練習 5-2

引数として渡された変数がオブジェクト(配列を含む)ならば、関数で変更すると呼び出した側で利用するときも変わっています。あるいは、関数の戻り値として値を受け取って、それを変数に代入することにより、その変数の値を変更することができます。また、大域変数(グローバル変数)を使うこともできます。

練習 5-3

たとえば、次のように functionC を定義し、その下のように呼び出します。

```
function functionC(objectA, functionB) {
    functionB(objectA);
}

var func = new Function('x', 'alert(x)');
functionC('こんにちは', func);
```

6章

練習 6-1

次の3つです。

- インラインモデル—— onload など、タグの属性として指定します
- JavaScript コードブロックに window.onload などのハンドラを記述
- DOM Level2 のイベントを使う—— addEventListener や attachEvent を利用します

練習 6-2

DOM Level 0 のイベント処理システムを使う場合、window オブジェクトの Event オブジェクトにアクセスするか、関数として渡された Event オブジェクトにアクセスします。DOM Level 2 の場合、Event オブジェクトは常に関数に渡されます。Event オブジェクトからは、プロパティの `screenX` と `screenY` にアクセスします。

練習 6-3

IE のアプローチだけが異なるため、2つの方法で処理する必要があります。イベントオブジェクトに関してメソッド `stopPropagation` がサポートされているかどうかをテストして、サポートされていればそれを呼び出し、そうでなければプロパティ `cancelBubble` を `true` にします。

練習 6-4

次のようにします。

```
if (window.addEventListener) {
    window.addEventListener("load", functionCall, false);
} else if (window.attachEvent) {
    window.attachEvent("onload", functionCall);
}
```

練習 6-5

キーボード関連のイベントについては説明しませんでした。典型的な方法としては、次のコードのように `keydown` イベントを捕捉しそのイベントの `which` プロパティから Unicode のキーにアクセスします。

```
if (document.addEventListener) {
    document.addEventListener("keydown", getKey, true);
} else if (document.attachEvent) {
    document.attachEvent("onkeydown", getKey);
}

function getKey(evt) {
    alert(evt.which);
}
```

7 章

練習 7-1

DOM Level 0 のイベントを利用している場合、イベントハンドラから `false` を返せば、送信はキャンセルされます。DOM Level 2 を使っている場合、IE では `cancelBubble` を `true` に設定し、ほかのブラウザ用には `preventDefault` を呼び出します。

練習 7-2

`blur` イベントは、フィールドがフォーカスを失ったときに発せられます。このタイミングでテキ

ストフィールドをチェックして妥当性を確認するのがよいでしょう。

練習 7-3

選択肢は Options という配列に保存されています。したがって、新しい配列要素を追加すると新しい選択肢を追加することができます。加える要素は Option オブジェクトでなければなりません。たとえば、次のようにします。

```
opts[opts.length] = new Option(" 選択肢 4", "opt4");
```

練習 7-4

たとえば、次のようにします。

```
var rgEx = /^[^0-9]*$/g;  
var OK = rgEx.exec(document.someForm.address.value);
```

練習 7-5

まず、各ラジオボタンにイベントハンドラ関数を割り当てておく必要があります。

```
document.someForm.radiogroup[0].onclick = handleClick;  
document.someForm.radiogroup[1].onclick = handleClick;
```

関数 handleClick で、あるラジオボタンがチェックされているかを確認し、チェックされていればテキストフィールドを無効にするといった操作を行います。たとえば、送信ボタンを無効にするには次のようにします。

```
function handleClick() {  
    if (document.someForm.radiogroup[1].checked) {  
        document.someForm.submit.disabled=true;  
    } else {  
        document.someForm.submit.disabled=false;  
    }  
}
```

ボタンの数が多いのならば、ループを使ってチェックするとよいでしょう。

8 章

練習 8-1

次のような方法が考えられます。

- クッキーを使う
- Flash など、プラグインを使う
- たとえば、マウスの右クリックでファイルを保存してもらう
- リンクをクリックするとファイルをダウンロードするようにする

練習 8-2

クッキー名、値、クッキーの有効期限、クッキーのパス

練習 8-3

有効期限を指定しなければ、ブラウザを閉じたときに破棄されます。

練習 8-4

害になるもの、あるいは害になる可能性のあるものはすべて取り除くのが一番安全な方法です。ただし、どのような利用者を対象としているのかによって、害になる可能性のある項目は変わってしまうため、どのようなデータを取り除けばよいかを一律に規定することはできません。

練習 8-5

この解答に「正解」はありませんが、たとえば以下のような用途にクッキーを利用することができます。

- ユーザー名や e メールアドレスなどログイン情報の保存
- ユーザーの好みの記憶。たとえば、表示色、文字の大きさなど
- ショッピングカートに入っているものの記憶
- 前回の訪問日
- 検索した単語などの履歴

9 章

練習 9-1

メソッド `prompt` を使って、ダイアログを開きます。

練習 9-2

次のようにします。

```
setTimeout(callFunction, 3000, paramA, paramB);
```

練習 9-3

ブラウザにロードされるページを変更するには `location` オブジェクトが利用できます。 `href` プロパティにページの URL を指定します。

練習 9-4

`navigator` オブジェクトを使います。

練習 9-5

次のコードでよいでしょう。

```
var newWindow = window.open("help.html", "",
    "width=200,height=200,toolbar=no,status=no");
```


10章

練習 10-1

以下の属性です。

```
id, title, lang, dir, className
```

練習 10-2

次のようにします。

```
var elems = document.getElementsByName('elemName');
for (var i = 0; i < elems.length; i++) {
    alert(elems[i].tagName);
}
```

練習 10-3

次のようにします。

```
var children = nd.childNodes;
for (var i = 0; i < children.length; i++) {
    alert(children[i].nodeType);
}
divs = document.getElementsByTagName('div');
for (var i = 0; i < divs.length; i++) {
    alert(divs[i].id);
}
var elem = document.getElementById("elem1");
var children = elem.childNodes;
var child = elem.getElementsByTagName('h1')[0];
var p = document.createElement("p");
var txt = document.createTextNode("hello");
p.appendChild(txt);
elem.replaceChild(p, child);
```

練習 10-4

次のコードでよいでしょう。

```
divs = document.getElementsByTagName('div');
for (var i = 0; i < divs.length; i++) {
    alert(divs[i].id);
}
```

練習 10-5

次のコードで可能です。

```
var elem = document.getElementById("elem1");
```

```
var children = elem.childNodes;
var child = elem.getElementsByTagName('h1')[0];
var p = document.createElement("p");
var txt = document.createTextNode("hello");
p.appendChild(txt);
elem.replaceChild(p,child);
```

11章

練習 11-1

prototype プロパティを使います。次のように定義します。

```
Number.prototype.triple = function () {
    var nm = this.valueOf() * 3;
    return nm;
}
var num = new Number(3.0);
alert(num.triple());
```

練習 11-2

varを使ってメンバを宣言します。隠蔽することにより、外部からのアクセスを制御でき、予測していない操作などをされないようにできます。

練習 11-3

throw 文を使って例外を起こします。呼び出す側では、これを try...catch で捕らえます。

```
if (typeof value != "number") {
    throw "NotANumber";
}
```

練習 11-4

イベントオブジェクトと違い、モデルの違いだけではないからです。プロパティの違いとプロパティに代入された値の違いがあります。

練習 11-5

次のようなコードを用います。

```
function Control() {
    var state = 'on';
    var background = '#fff';
    this.changeState = function() {
        if (state == 'on') {
            state = 'off';
            background = '#000';
        } else
            state = 'on';
    }
}
```

```
        background = '#fff';
    }
    this.getState = function() {
        return state;
    }
    this.getColor = function() {
        return background;
    }
}
```

12章

練習 12-1

258ページのメモにあるように、たとえスタイルシートでstyleプロパティに値を設定したとしても、JavaScriptや要素のインラインstyle属性で直接、そのプロパティに対して値を設定しなければ、スクリプト経由でそのプロパティにアクセスしたときの値は空あるいは未定義になります。

練習 12-2

フォントサイズと行の高さは一度に変更可能です。次のようにします。

```
obj.style.font="14pt/16pt";
obj.style.color="#f00";
```

練習 12-3

DIVブロック内にさらに別のブロックがあり、その中でスタイルが設定されていれば、変更できない可能性があります。

練習 12-4

widthあるいはheightを0にするのがひとつの方法です。visibilityをhiddenに設定する方法もあります。また、他の要素を上から描いて隠してしまうこともできます。

13章

練習 13-1

XMLHttpRequest.openの第3引数をfalseにして呼び出すと、同期的な(レスポンスを待つ)リクエストになります。

練習 13-2

XMLHttpRequestのプロパティonReadyStateChangeを介してコールバック関数を割り当てます。

練習 13-3

XMLHttpRequest オブジェクトのプロパティ readyStateは完了時には4になっています。

このとき、サービスのリクエストが成功すると、リクエストのオブジェクトのHTTP statusプロパティは200になっています。

練習 13-4

HTMLの場合、フォーマットせずにページに追加することができます。XMLの場合XSLTを使ってフォーマットできます。JSONの場合 eval 関数を使って以降の処理に備えることができます。

14章

練習 14-1

まず、DOM Level 2 のイベントハンドラを使っていることを確認してください。window.onload=function の形式の DOM Level 0 のイベントハンドラを使っていると、Dojo ライブラリがイベントに割り当てたイベントハンドラを上書きしてしまっている可能性があります。

練習 14-2

document.getElementById の代わりに使うことができ、複数の要素を引数に指定すると、それぞれの値を配列にして返してくれます。