第 1 刷～第 3 刷 訂正情報

本書（第 1 刷～第 3 刷）の掲載内容に下記の誤りがございました。ご迷惑をおかけしましたことをお詫び申し上げます。

（項目 027）　P.077　JavaScript（027/main.js）　1～5 行目

【誤】

"/** テキストエリア */
let textarea = document.querySelector('.textarea');

/** 入力中の文字数 */
let string_num = document.querySelector('.string_num');"

【正】

"/** テキストエリア */
const textarea = document.querySelector('.textarea');

/** 入力中の文字数 */
const string_num = document.querySelector('.string_num');"

（項目 053）　P.129　本文　4 行目

【誤】

スプレッド演算子（...）を用いると、[...配列]のように指定することで、

【正】

スプレッド構文（...）を用いると、[...配列]のように指定することで、

（項目 064）　P.152　本文　1 行目

【誤】

スプレッド演算子（...）を用いると、配列のようなオブジェクト

【正】

スプレッド構文（...）を用いると、配列のようなオブジェクト

（項目 064）　P.153　本文　1 行目

【誤】

スプレッド演算子（...）を用いると、ArrayLike オブジェクトを配列に変換できます。

【正】

スプレッド構文（...）を用いると、ArrayLike オブジェクトを配列に変換できます。

（項目 064） P.154　コラム　1 行目

【誤】

Array.from()もスプレッド演算子（...）と同様に ArrayLike オブジェクトを配列に 変換できます。

【正】

Array.from()もスプレッド構文（...）と同様に ArrayLike オブジェクトを配列に 変換できます。

（項目 064）　P.154　コラム　4～6 行目

【誤】

スプレッド演算子（...）でも map()を組み合わせれば同等のことが実現できるので、ブラウザー互換の問題がない限りはスプレッド演算子を使うほうが手軽でしょう。

【正】

スプレッド構文（...）でも map()を組み合わせれば同等のことが実現できるので、ブラウザー互換の問題がない限りはスプレッド構文を使うほうが手軽でしょう。

（項目 069）　P.163　本文　1 行目

【誤】

スプレッド演算子（...）を用いると、次のような短いコードでオブジェクトの

【正】

スプレッド構文（...）を用いると、次のような短いコードでオブジェクトの

（項目 069）　P.164　本文　1 行目

【誤】

Object.assign()メソッドやスプレッド演算子を使うと、

【正】

Object.assign()メソッドやスプレッド構文を使うと、

（項目 069）　P.164　本文　5 行目

【誤】

次のコードでは、スプレッド演算子によってオブジェクトをコピーしています。

【正】

次のコードでは、スプレッド構文によってオブジェクトをコピーしています。

（項目 073）　P.170　利用シーン

【誤】

オブジェクトを深い階層まで編集不可能にするとき

【正】

オブジェクトを編集不可能にするとき

（項目 073）　P.170　本文　3～4 行目

【誤】

プロパティーの追加、削除、変更を禁止するには、Object.freeze()メソッドを用います。

【正】

プロパティーの追加、削除、変更を禁止するには、Object.freeze()メソッドを用います。
ただし、編集不可能になるのオブジェクト直属のプロパティで、深い階層までは編集不可能になりません。

（項目 178）　P.384　**注釈**　1～2 行目

【誤】

「メッセージを作成」箇所の処理は、三項演算子を用いています。「`真偽値 ？ 値 1： 値 2`」と記述することで、真偽値が `true`の場合は`値 1`、真偽値が`false`の場合は`値 2`が返ります。

【正】

「メッセージを作成」箇所の処理は、三項演算子を用いています。「真偽値 ？ 値 1： 値 2」と記述することで、真偽値が true の場合は値 1、真偽値が false の場合は値 2 が返ります。

（項目 209）　P.443　JavaScript（209/main.js）　7 行目

【誤】

```
const fileName = `mySvg.svg`;
```

【正】

```
const fileName = 'mySvg.svg';
```

（項目 273）　P.580　JavaScript　3 行目

【誤】

```
for (let value of array) {
```

【正】

for (const value of array) {

（項目 274）　P.583　JavaScript（2 つ目）　10 行目

【誤】

for (let value of myIterable) {

【正】

for (const value of myIterable) {

（項目 274）　P.585　JavaScript（274/sample2/main.js）　11 行目

【誤】

for (let value of range(2, 6)) {

【正】

for (const value of range(2, 6)) {

（項目 277）　P.595　JavaScript（1 つ目）　6～7 行目

【誤】

"// 20, 50, 120 が順番に出力
for (let key of keyList) {"

【正】

"// 20, 50, 120 が順番に出力
for (const key of keyList) {"

（項目 277）　P.595　JavaScript（1 つ目）　13～14 行目

【誤】

"// '鈴木', '田中', '高橋'が順番に出力
for (let value of valueList) {"

【正】

"// '鈴木', '田中', '高橋'が順番に出力
for (const value of valueList) {"

（項目 277）　P.595　JavaScript（1 つ目）　20～21 行目

【誤】

"// [20, '鈴木'], [50, '田中'], [120, '高橋']が順番に出力
for (let entry of entryList) {"

【正】

"// [20, '鈴木'], [50, '田中'], [120, '高橋']が順番に出力
for (const entry of entryList) {"

（項目 278）　P.597　JavaScript（3 つ目）　6 行目

【誤】

```
for (let value of valueList) {
```

【正】

```
for (const value of valueList) {
```