

第 1 刷 訂正情報

本書（第 1 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。

P.60 本文 2～3 行目

【誤】

for 文のインデックスが偶数のときのみログを出力します。

【正】

for 文のインデックスが**奇数**のときのみログを出力します。

P.67 JavaScript(022/main.js) 4 行目

【誤】

```
document.querySelector('.result4').innerHTML = Math.round(6.24);
```

【正】

```
document.querySelector('.result4').innerHTML = Math.trunc(6.24);
```

P.69 JavaScript(023/main.js) 10 行目

【誤】

```
// 0~360 の間のランダムな数を取得する
```

【正】

```
// 0~359 の間のランダムな数を取得する
```

P.77 Syntax 2 行目

【誤】

```
[...文字列].length
```

【正】

```
Array.from(文字列).length
```

P.78 コラム 8~11 行目

【誤】

サロゲートペアを一文字としてカウントするには、次のように... (Spread Operator) を用います。

■ JavaScript

```
[...'𐀀'].length; // 1  
[...'𐀀𐀀'].length; // 1
```

【正】

サロゲートペアを **含めて** 一文字としてカウントするには、次のように `Array.from()` を用います。

■ JavaScript

```
Array.from('𐀀').length; // 1  
Array.from('𐀀𐀀').length; // 1
```

※絵文字にスキントーンが含まれている場合など、一部の文字では一文字としてカウントできない場合もあるのでご注意ください。

P.105 JavaScript(1つ目)4~7行目

【誤】

```
/va*/.test('JavaScript'); // 「v」の後「a」が0回以上続くかどうか。true  
/鈴木.*郎/.test('鈴木一郎'); // 「鈴木」で始まり「郎」で終わるかどうか。true。  
/¥d*-¥d*-¥d*/.test('090-1234-5678'); // 「数字-数字-数字」という形かどうか。true
```

【正】

```
/Java.*/.test('JavaScript'); // 「Java」の後「a」が0回以上続くかどうか。true  
/鈴木.*郎/.test('鈴木一郎'); // 「鈴木郎」または「鈴木」と「郎」の間に文字が含まれるかどうか。true。  
/¥d+-¥d+-¥d+/.test('090-1234-5678'); // 「数字-数字-数字」という形かどうか。true
```

P.116 JavaScript 13行目

【誤】

```
decodeURI(  
  
'http%3A%2F%2Fexample.com%2F%E5%8F%AF%E6%84%9B%E3%81%84%E7%8C%AB%E3%81%AE%20%E3%83%9A%E3%83%BC%E3%82%B8.html'  
);  
// http://example.com/可愛い猫の ページ.html
```

【正】

```
decodeURIComponent(  
  
'http%3A%2F%2Fexample.com%2F%E5%8F%AF%E6%84%9B%E3%81%84%E7%8C%AB%E3%81%AE%20%E3%83%9A%E3%83%BC%E3%82%B8.html'  
);  
// http://example.com/可愛い猫の ページ.html
```

P.137 本文 8 行目

【誤】

[1, 2, 3, 4, 5]という数値が入った配列を

【正】

[1, 2, 3, **3**, 4, 5]という数値が入った配列を

P.164 本文 6 行目以降

【誤】

コピー元の name プロパティを John に変更してますが、コピー先の同プロパティも John になっていることがわかります。これがシャローコピーの挙動となります。

■ JavaScript

```
// コピー元のオブジェクト
const object3 = {
  id: 1,
  name: '鈴木'
};

// オブジェクトのコピー
const copiedObject3 = { ...object3 };

// 元オブジェクトの name プロパティを更新する
object3.name = 'John';

// コピー先の name プロパティも更新される
console.log(copiedObject3.name); // 結果: 'John'
```

【正】

コピー元の `members` プロパティ内の配列に注目してください。1 番目の要素を `John` に変更しています。すると、コピー先の `members` プロパティの配列も変更されていることがわかります。これがシャローコピーの挙動となります。

■ JavaScript

```
// コピー元のオブジェクト
const object3 = {
  id: 1,
  members: [ '鈴木', '田中', '高橋' ]
};

// オブジェクトのコピー
const copiedObject3 = { ...object3 };

// 元オブジェクトの members プロパティ内の配列を更新する
object3.members[0] = 'John';

// コピー先の members プロパティの配列も更新される
console.log(copiedObject3.members[0]); // 結果: 'John'
```