

## SVG 資料第 7 回目 (その 3) 配列のメソッドの利用

メディア専門ユニット I(SVG)

2016/6/6

- ▶ 通常、配列の各要素について操作をするためには for 文などのループで処理を行う。
- ▶ JavaScript では for 文の代わりになるメソッドがいくつか定義されている。それぞれのメソッドは引数に関数が必要とする。
  - ▶ 引数に渡される関数には最大で 3 つの引数を渡される。
  - ▶ 一つ目の引数は配列の要素
  - ▶ 2 つ目の引数は配列のインデックス
  - ▶ 3 つ目の引数はメソッドが適用される配列自身である。これを利用すると元来の配列の要素を変更できる。

`forEach(func)`: 引数に関数を

## 配列のメソッド-forEach(func)

第 7 回目 (その 3)

メディア専門ユニット I(SVG)

配列のメソッド

- ▶ 与えられた関数を配列の各要素に対して実行
- ▶ 与えられた関数の戻り値は無視される。
- ▶ 途中でループの処理を中断できない。
- ▶ 要素の値が undefined のところは実行されない。

## 配列のメソッド-forEach(func) 実行例

第 7 回目 (その 3)

メディア専門ユニット I(SVG)

配列のメソッド

```
>A = [1,2,3,4];
(4) [1, 2, 3, 4]
>A[5]=5;
5
>A;
(6) [1, 2, 3, 4, undefined × 1, 5]
>A.forEach(function(V,i,Vs){
  Vs[i] = V*V;
  console.log(V*V);
});
1
4
9
16
25
undefined
>A;
[1, 4, 9, 16, undefined × 1, 25]
>undefined*undefined
NaN
```

- ▶ 配列の各要素を 2 乗している。
- ▶ 元の配列に代入しているので、実行後、要素の値が変化する。
- ▶ undefined の値のところは実行されていない。

### map(func)

配列の各要素に対して引数の関数を実行し、その戻り値で新しい配列を作成する。

```
>A=[1,2,3]
(3) [1, 2, 3]
>A[5]=5
5
>A;
(6) [1, 2, 3, undefined × 2, 5]
>>A.map(function(V){return V*V;});
(6) [1, 4, 9, undefined × 2, 25]
>A;
(6) [1, 2, 3, undefined × 2, 5]
```

- ▶ 要素を 2 乗した配列を作成している。
- ▶ undefined の要素はそのまま undefined
- ▶ 元の配列は変化していない。

# 配列のメソッド-map(func) 実行例

第 7 回目 (その 3)

メディア専門ユニット I(SVG)

配列のメソッド

▶ `every(func)`

配列の各要素に対して引数の関数を実行し、その戻り値がすべて `true` のとき、`true` を返す。どこかで `false` になると実行はそこで打ち切られ、`false` を返す。

▶ `some(func)`

配列の各要素に対して引数の関数を実行し、その戻り値がすべて `false` のとき、`false` を返す。どこかで `true` になると実行はそこで打ち切られ、`true` を返す。

▶ `filter(func)`

配列の各要素に対して引数の関数を実行し、その戻り値が `true` になるものだけを集めた新しい配列を返す。