

①課題

Week2 Thursday Lesson 9 応用演習 ナマケモノタイム 2

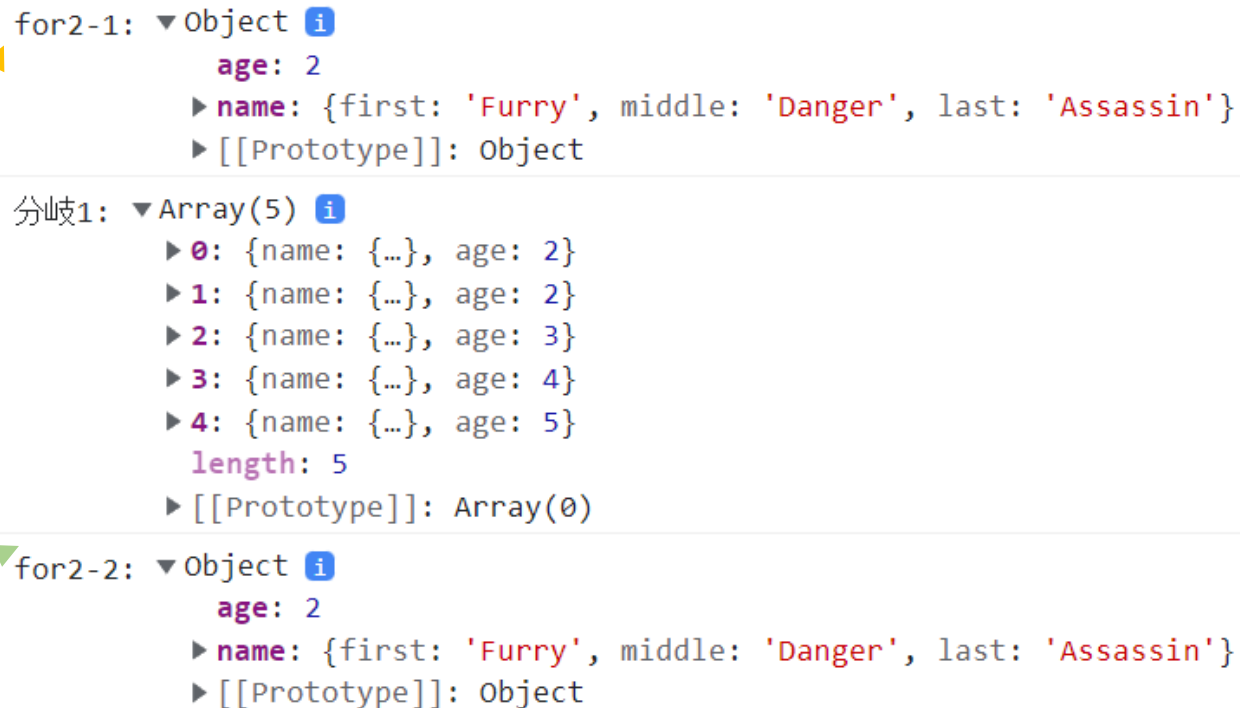
引数 : 要素にオブジェクトを持つ配列

返回值 : 与えられたすべてのナマケモノを若い順(ageの昇順) に並び替えた新たな配列。
同じ年齢(age) の場合は、元の順番のままにする。

作成・実行 ⇒ “Test FAILED. Keep trying!” (actualに正しい配列が入っていない)
⇒ console.log()を使って不具合を調査

```
function getSortedByAge(arrayOfSloths) {  
  let result = [];  
  console.log("スタート:", result);  
  for (const obj1 of arrayOfSloths) {  
    console.log("for1:", obj1)  
    if (result[0] === undefined) {  
      console.log("for2-1:", obj1);  
      result[0] = obj1;  
      console.log("分岐1:", result);  
      console.log("for2-2:", obj1);  
    } else {  

```



for2-1: ▼ Object **i**

- age: 2
- ▶ name: {first: 'Furry', middle: 'Danger', last: 'Assassin'}
- ▶ [[Prototype]]: Object

分岐1: ▼ Array(5) **i**

- ▶ 0: {name: {...}, age: 2}
- ▶ 1: {name: {...}, age: 2}
- ▶ 2: {name: {...}, age: 3}
- ▶ 3: {name: {...}, age: 4}
- ▶ 4: {name: {...}, age: 5}
- length: 5
- ▶ [[Prototype]]: Array(0)

for2-2: ▼ Object **i**

- age: 2
- ▶ name: {first: 'Furry', middle: 'Danger', last: 'Assassin'}
- ▶ [[Prototype]]: Object

空の配列に要素を一つ追加すると大量の要素が追加

②原因

helpに相談！！ ⇒ console.logの仕様で正しくない結果が表示されていると判明

引数

obj1 ... objN

出力する JavaScript オブジェクトのリスト。各オブジェクトの文字列表現が記述順で出力されます。Chrome や Firefox の比較的新しいバージョンを使っているなら注意が必要です。これらのブラウザーで記録されるのはオブジェクトへの参照です。そのため、console.log() を呼び出した時点でのオブジェクトの「値」が表示されるのではなく、内容を見るために開いた時点での値が表示されます。

```
console.log("表示確認1");  
let array = [];  
console.log(array);  
array.push(1);  
console.log(array);  
array.push(2);  
console.log(array);
```

コンソール
表示結果

表示確認1

▶ []

▶ [1]

▶ (2) [1, 2]



▶ を開く

全ての中身が同じ

▼ [] ⓘ

0: 1

1: 2

length: 2

▶ [[Prototype]]: Array(0)

▼ [1] ⓘ

0: 1

1: 2

length: 2

▶ [[Prototype]]: Array(0)

▼ (2) [1, 2] ⓘ

0: 1

1: 2

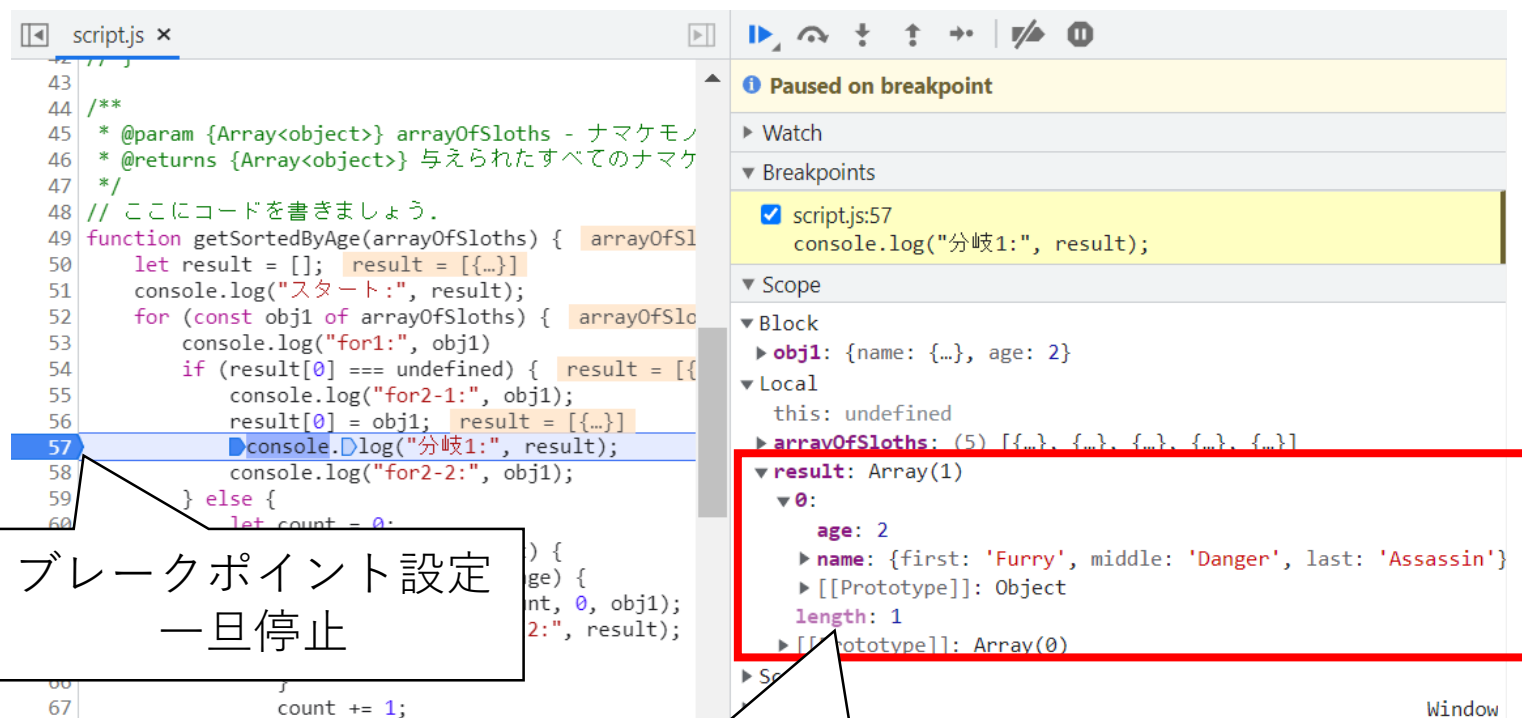
length: 2

▶ [[Prototype]]: Array(0)

Console.logの仕様により、オブジェクトの最終結果のみ表示

③対応策

Chrome : developer tool の sources タブの × ブレークポイント の活用



ブレークポイント設定
一旦停止

正しく入力されている
ことを確認

+ α の気づき

```
function getSortedByAge(arrayOfSloths) {
  let result = [];
  console.log("スタート:", result);
  for (const obj1 of arrayOfSloths) {
    console.log("for1:", obj1);
    if (result[0] === undefined) {
      console.log("for2-1:", obj1);
      result[0] = obj1;
      console.log("分岐1:", result);
      console.log("for2-2:", obj1);
    } else {
      let count = 0;
      for (const obj2 of result) {
```

1行ずつ実行することで
イメージ通りの分岐に
進んでいるかを簡単に確認

sources タブのブレークポイントを活動し、実態を正しく把握