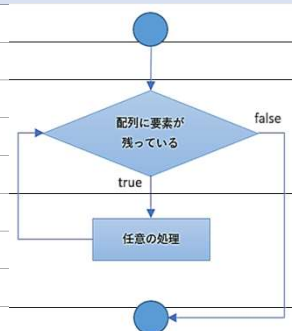


## ■教育実施記録帳

|        |  |         |   |
|--------|--|---------|---|
| 管理No.  | FND27  | Day8    | Page1                                     |
| 実施内容   | for.of ループ、TDD レビュー、HTML + CSS & レイアウト   |         |   |
| 場所     | ライオン   | 設備・工機   |   |
| 実施日    | 2024/5/13  | 実施時間    | 9:00 ~ 12:00                              |
| 実施者    | Toraさん   | 受講者     |   |
|        | Masaさん   |         |   |
| 作成者    | 本田   | 作成日(開始) | 2024/5/13                                 |
|        |  | 作成日(完了) | 2024/5/13                                 |
| <Time> | <Contents>   |         |   |
|        | ★.indexOf()メソッド  |         |   |
|        | [構文] <code>arr.indexOf( 検索する要素 , 検索開始位置 )</code>   |         |   |
|        | [利用] 配列の要素の位置を先頭から検索   |         |   |
|        | [戻り値] 見つかった要素位置 (インデックス) 、要素が見つからない場合は「-1」を返す  |         |   |
|        | ・検索開始位置は省略可能   |         |   |
|        | // 配列の作成   |         |   |
|        | <code>const arrTreats = ["apple", "banana", "chocolate", "doughnut"];</code>             |         |   |
|        | // 例1 : 要素の位置を先頭から検索   |         |   |
|        | <code>console.log(arrTreats.indexOf("doughnut")); // ⇒ 3</code>                          |         |   |
|        | // 要素がない場合   |         |   |
|        | <code>console.log(arrTreats.indexOf("ramen")); // ⇒ -1</code>                            |         |   |
|        | // ウォーミングアップ復習 : 末尾に追加した要素を取得する  |         |   |
|        | <code>arrTreats.push("takoyaki");</code>   |         |   |
|        | <code>console.log(arrTreats[arrTreats.length - 1]); // ⇒ "takoyaki"</code>               |         |   |
|        | // 分解すると   |         |   |
|        | <code>console.log(arrTreats.length); // ⇒ 5</code>                                       |         | //ここまですと要素番号                              |
|        | <code>console.log(arrTreats[5 - 1]); // ⇒ takoyaki</code>                                |         | //ここでインデックス値に                             |
|        | // 検索開始位置を指定した場合   |         |   |
|        | <code>console.log(arrTreats.indexOf("chocolate", 0)); // ⇒ 2</code>                      |         |   |
|        | <code>console.log(arrTreats.indexOf("chocolate", 1)); // ⇒ 2</code>                      |         |   |
|        | <code>console.log(arrTreats.indexOf("chocolate", 2)); // ⇒ 2</code>                      |         |   |
|        | <code>console.log(arrTreats.indexOf("chocolate", 3)); // ⇒ -1</code>                     |         |   |
|        | <code>const arrTreats = ["apple", "banana", "chocolate", "doughnut", "takoyaki"];</code> |         |   |
|        | インデックス ⇒ [ 0 , 1 , 2 , 3 , 4 ];  |         |   |
|        |  |         |   |
|        | ☆.lastIndexOf()メソッド : 配列の要素の位置を末尾から検索  |         |   |
|        | ★for...of命令  |         |   |
|        | [構文] <code>for ( 変数 of 配列 ) {</code>   |         |   |
|        | ...繰り返し処理...   |         |   |
|        | }  |         |   |
|        | [例] 配列の値を1つずつ変数へ代入して繰り返し処理をする  |         |   |
|        | [戻り値] 繰り返し処理が終わるまで、処理毎の値を変数へ都度代入   |         |   |
|        | // 配列の作成   |         |   |
|        | <code>const colors = ["red", "green", "blue"];</code>                                    |         |   |
|        | // 例   |         |   |
|        | <code>for (const color of colors) .</code>   |         | //colors配列に取出していない内容があったらcolor変数に内容の一つ取出し |
|        | <code>console.log(color);</code>   |         | //コンソールに表示。                               |
|        | }  |         | //colors配列に取出していない内容が無くなったら処理終了           |
|        | // 戻り値   |         |   |
|        | red // 1回目の処理の結果 ... 配列に要素残っている  |         |   |
|        | green // 2回目の処理の結果 ... 配列に要素残っている  |         |   |
|        | blue // 3回目の処理の結果 ... 配列に要素残っていないから処理終了  |         |   |



## ■教育実施記録帳

|        |  |       |              |           |         |           |
|--------|--|-------|--------------|-----------|---------|-----------|
| 管理No.  | FND27  |       | Day8         |           | Page1   |           |
| 実施内容   | for..of ループ、TDD レビュー、HTML + CSS &レイアウト                         |       |              |           |         |           |
| 場所     | 5F   | 設備・工機 |              |           |         |           |
| 実施日    | 2024/5/13  | 実施時間  | 9:00 ~ 12:00 |           |         |           |
| 実施者    | Toraさん   | 受講者   |              |           |         |           |
|        | Masaさん   |       |              |           |         |           |
| 作成者    | 本田   |       | 作成日(開始)      | 2024/5/13 | 作成日(完了) | 2024/5/13 |
| <Time> | <Contents>   |       |              |           |         |           |
|        | ★テストコード、プロダクションコードの中で配列の繰り返し処理を行うには？                           |       |              |           |         |           |
|        | // 関数にしたテストを書く   |       |              |           |         |           |
|        | function test(actual, expected) {                              |       |              |           |         |           |
|        | if (JSON.stringify(actual) === (JSON.stringify(expected))) {   |       |              |           |         |           |
|        | console.log("Yay! Test PASSED.");                              |       |              |           |         |           |
|        | } else {   |       |              |           |         |           |
|        | console.error("Test FAILED. Keep trying");                     |       |              |           |         |           |
|        | console.log("expected:",expected);                             |       |              |           |         |           |
|        | console.log(" actual:",actual);                                |       |              |           |         |           |
|        | }  |       |              |           |         |           |
|        | }  |       |              |           |         |           |
|        | // JSONブロックを書く   |       |              |           |         |           |
|        | /**  |       |              |           |         |           |
|        | * @param {Array<number>} arrNums //数値型の要素を持つ配列                 |       |              |           |         |           |
|        | * @returns {number} sumResalt //与えられた配列のすべての数字の合計              |       |              |           |         |           |
|        | * 関数sumArrayは数値型の要素を持つ配列を1個取り、                                 |       |              |           |         |           |
|        | * 与えられた配列のすべての数字の合計を数値型データで返す                                  |       |              |           |         |           |
|        | */   |       |              |           |         |           |
|        | // テストコードを書く   |       |              |           |         |           |
|        | test(sumArray([1, 2, 3, 4]),10);                               |       |              |           |         |           |
|        | ・求める期待値に使用さる配列の内容と、処理結果の実際値が指定される                              |       |              |           |         |           |
|        | // プロダクションコードを書く   |       |              |           |         |           |
|        | function sumArray(arrNums) {                                   |       |              |           |         |           |
|        | ・const sumArray = [1, 2, 3, 4] //[]部分を再利用するために関数にしている          |       |              |           |         |           |
|        | // 処理結果の初期化  |       |              |           |         |           |
|        | let sumResalt = 0; //合計値用変数の初期値決定                              |       |              |           |         |           |
|        | // 何かする  |       |              |           |         |           |
|        | for (const arrNum of arrNums) { // [1, 2, 3, 4]から1つずつarrNumに代入 |       |              |           |         |           |
|        | sumResalt = sumResalt + arrNum //合計値用変数にarrNumの値を加算            |       |              |           |         |           |
|        | console.log("sumResalt:", sumResalt) //現段階でのsumResaltの値を表示     |       |              |           |         |           |
|        | } //上記3点を配列末尾まで繰り返す  |       |              |           |         |           |
|        | // 処理結果を返す   |       |              |           |         |           |
|        | return sumResalt;  |       |              |           |         |           |
|        | ・test(sumArray([1, 2, 3, 4])の返り値がsumResaltになり、,実際値10);との比較となる  |       |              |           |         |           |
|        | ・実行時のコンソールの表示  |       |              |           |         |           |
|        | sumResalt : 1  |       |              |           |         |           |
|        | sumResalt : 3  |       |              |           |         |           |
|        | sumResalt : 6  |       |              |           |         |           |
|        | sumResalt : 10   |       |              |           |         |           |
|        | Yay! Test PASSED.  |       |              |           |         |           |
|        | ☆「結果の初期化、何かする、結果を返す」をパターンとして覚える                                |       |              |           |         |           |

## ■教育実施記録帳

[illegible]