**Free Code Camp: Basic Data Structures Challenges**

**1. Use an Array to Store a Collection of Data**

let yourArray = [2, "two", true, 17, "blue"];

**2. Access an Array’s Contents Using Bracket Notation**

let myArray = ["a", "b", "c", "d"];

myArray[1] = "e";

console.log(myArray);

**3. Add Items to an Array with push() and unshift()**

function mixedNumbers(arr) {

arr.unshift('I', 2, 'three');

arr.push(7, 'VIII', 9);

return arr;

}

console.log(mixedNumbers(['IV', 5, 'six']));

**4. Remove Items from an Array with pop() and shift()**

function popShift(arr) {

let popped = arr.pop();

let shifted = arr.shift();

return [shifted, popped];

}

console.log(popShift(['challenge', 'is', 'not', 'complete']));

**5. Remove Items Using splice()**

function sumOfTen(arr) {

arr.splice(2,2);

return arr.reduce((a, b) => a + b);

}

console.log(sumOfTen([2, 5, 1, 5, 2, 1]));

**6. Add Items using splice()**

fucction htmlColorNames(arr) {

arr.splice(0, 2, "DarkSalmon", "BlanchedAlmond");

return arr;

}

console.log(htmlColorNames(['DarkGoldenRod', 'WhiteSmoke', 'LavenderBlush', 'PaleTurqoise', 'FireBrick']));

**7. Copy Array Items Using slice()**

slice()takes only 2 parameters — the first is the index at which to begin extraction, and the second is the index at which to stop extraction (extraction will occur up to, but not including the element at this index).

function forecast(arr) {

return arr.slice(2,4);

}

console.log(forecast(['cold', 'rainy', 'warm', 'sunny', 'cool', 'thunderstorms']));

**8. Copy an Array with the Spread Operator**

function copyMachine(arr, num) {

let newArr = [];

while (num >= 1) {

newArr.push([...arr]);

num--;

}

return newArr;

}

console.log(copyMachine([true, false, true], 2));

**9. Combine Arrays with the Spread Operator**

function spreadOut() {

let fragment = ['to', 'code'];

let sentence = ['learning', ...fragment, 'is', 'fun'];

return sentence;

}

console.log(spreadOut());

**10. Check For The Presence of an Element With indexOf()**

function quickCheck(arr, elem) {

if(arr.indexOf(elem) === -1) {

return false;

} else {

return true;

}

}

console.log(quickCheck(['squash', 'onions', 'shallots'], 'mushrooms'));

**11. Iterate Through All an Array’s Items Using For Loops**

function filteredArray(arr, elem) {

let newArr = [];

for (let i = 0; i < arr.length; i++) {

if (arr[i].indexOf(elem) === -1) {

newArr.push(arr[i]);

}

}

return newArr;

}

console.log(filteredArray([[3, 2, 3], [1, 6, 3], [3, 13, 26], [19, 3, 9]], 3));

**12. Create complex multi-dimensional arrays**

let myNestedArray = [

['unshift', false, 1, 2, 3, 'complex', 'nested'],

[['loop', 'shift', 'deep'], [6, 7, 1000, 'method']],

['concat', false, true, 'spread', 'array'],

[

[

['mutate', 1327.98, 'splice'], ['slice', 'push', 'deeper']

],

[

[

['iterate', 1.3849, 7, '8.4876', 'arbitrary', 'depth', 'deepest']

]

]

]

];