

P1: students array demo using map, filter, find, reduce

The screenshot shows a browser developer tools console with the URL `127.0.0.1:5501/tku-javascript-learn/...`. The code in `app.js` uses several array methods to manipulate a `students` array:

- `updateStudents`: A callback function that maps each student to a new object where `student.role` is set to 'student'.
- `highScores`: A filtered array of students with scores greater than or equal to 50.
- `specificID`: A found object representing the student with `id === 1`.
- `averageScore`: A reduced array of scores to calculate the average.
- `survey`: A reduced array of favorite courses to calculate the count.

The console output shows the resulting arrays and objects, with specific elements highlighted by red boxes.

P2: getRandomUser three times

The screenshot shows a browser developer tools console with the URL `127.0.0.1:5501/tku-javascript-learn/...`. The code in `script.js` uses the `getRandomUser` function to add three users to the DOM:

- `addData`: A function that pushes user objects to a global `data` array.
- `getRandomUser`: An asynchronous function that fetches a random user from the `'https://randomuser.me/api'` API.
- The `getRandomUser` function is called three times sequentially.

The console output shows the `data` array being populated with three user objects, with specific elements highlighted by red boxes.

P3: add 6 users to the DOM

tku-javascript-learn > billy-school > w06 > theme > `script.js` > `addData`

```

15 //           return `<div class="person"><strong>${item.name}</strong> ${item.money}</div>
16 //         )
17 //       .join('');
18 //
19 //   addUserBtn.addEventListener('click', () => {
20 //     main.innerHTML = `<h2><strong>Person</strong> Wealth</h2> ${html}`;
21 //   });
22 // }

23 const updateDOM = () => {
24   const tempData = data
25   .map(item => {
26     return `<div class="person"><strong>${item.name}</strong> ${item.money}</div>`;
27   })
28   .join('');
29   console.log('tempData', tempData);
30   main.innerHTML = `<h2><strong>Person</strong> Wealth</h2> ${tempData}`;
31 }
32 }

33 const addData = obj => {
34   data.push(obj);
35   console.log('data', data);
36   updateDOM();
37 }

38 }

39 function formatMoney(number) {
40   return '$' + number.toFixed(2).replace(/\d{1}(\d{3})+\./g, '$&,');
41 }

42 }

43 const getRandomUser = async () => {
44   const res = await fetch('https://randomuser.me/api');
45   const data = await res.json();
46   console.log('random user data', data);
47   const user = data.results[0];
48   const newUser = {
49     name: `${user.name.first} ${user.name.last}`,
50     money: Math.floor(Math.random() * 1000000000),
51   }
52 }

```

OM Array Methods

Person	Wealth
Daniela Santana	82748398702
Quirio Caldeira	37714458272
Melvin Smith	64637971954
Emile Wilson	46802672098
Anjana Sponselee	67298656649
Juho Savela	9937235848

Console output:

```

37714458272 <div><strong>Daniela Santana</strong></div>
82748398702 </div><div><strong>Quirio Caldeira</strong></div>
37714458272 </div><div><strong>Melvin Smith</strong> 64637971954</div>
46802672098 </div><div><strong>Emile Wilson</strong> 46802672098</div>
67298656649 </div><div><strong>Anjana Sponselee</strong> 67298656649</div>
9937235848 </div><div><strong>Juho Savela</strong> 9937235848</div>

```

P4: add 6 users first, then filter condition set to > 30000000

const config = {
 numRandom: 6,
 ratio: 1.5,
 showCondition: 2000000,
};

window.addEventListener('DOMContentLoaded', () => {
 getRandomUser();
});

const updateDOM = (provideData = data) => {
 let tempData = provideData
 .map(item => {
 return `<div class="person">\${item.name} \${item.money}</div>`;
 })
 .join('');
 console.log('tempData', tempData);
 main.innerHTML = `<h2>Person Wealth</h2> \${tempData}`;
};

const addData = obj => {
 data.push(obj);
 console.log('data', data);
 updateDOM();
};

function formatMoney(number) {
 return '\$' + number.toFixed(2).replace(/\d{1}(\d{3})+\./g, '\$&,');
}

const getRandomUser = async n => {
 for (let i = 0; i < n; i++) {
 const res = await fetch('https://randomuser.me/api');
 const data = await res.json();
 console.log('random user data', data);
 const user = data.results[0];
 const newUser = {
 name: `\${user.name.first} \${user.name.last}`,
 }
 addData(newUser);
 }
}

Person

Person	Wealth
Robin Mercier	\$669274
Vicente Dominguez	\$456835
Eliab Almeida	\$513666
Romane Garcia	\$1192629
Julius Ollila	\$240935
Diane Reyes	\$76151

Console output:

```

random user data > Object
data ▶ Array(6) ▶
  ▷ 0: {name: 'Robin Mercier', money: 669274}
  ▷ 1: {name: 'Vicente Dominguez', money: 456835}
  ▷ 2: {name: 'Eliab Almeida', money: 513666}
  ▷ 3: {name: 'Romane Garcia', money: 1192629}
  ▷ 4: {name: 'Julius Ollila', money: 240935}
  ▷ 5: {name: 'Diane Reyes', money: 76151}
  length: 6
  [[Prototype]]: Array(0)

tempData <div><strong>Robin Mercier</strong> $669274</div><div><strong>Vicente Dominguez</strong> $456835</div><div><strong>Eliab Almeida</strong> $513666</div><div><strong>Romane Garcia</strong> $1192629</div><div><strong>Julius Ollila</strong> $240935</div><div><strong>Diane Reyes</strong> $76151</div>

```

P5: use config for three buttons

```
const config = {  
  numRandom: 5,  
  ratio: 1.5,  
  showCondition: 2000000,  
};
```

DOM Array Methods

Add User (5) 🧑

Raise 1.5 Money 💰

Show > 2000000 💰

Sort by Richest ↓

Calculate entire Wealth 📊

Person	Wealth
Joris Roche	\$2391923671.0808945
Kyro Arkenbout	\$2109863572.1004982
Milton Stewart	\$517822279.03492355
Erhardt Pelz	\$2904954624.6407433
Bernard Andrews	\$449097350.82951736
Pedro Crespo	\$61356159.29154968
Judy Simmons	\$1443284085.8806114
Lotty Beer	\$460123902.1464615
Lucas Silva	\$42283293.54785156
Lucy Johnson	\$11505443.923828125
Sonia Murray	\$112930405.18945312
Aubree Park	\$23527854.922851562
Esat Körmükçü	\$106666367.32617188
Suzanne Reeves	\$97726556.3203125