

LAB 4: WORKING WITH ARRAY

MỤC TIÊU

Kết thúc bài thực hành này bạn có khả năng

- ✓ Làm việc với Mảng, Maps, Sets và các phương thức (map, filter, reduce, Magic of Chaining, find, findIndex, some and every, flat and flatMap,...)

NỘI DUNG

LAB4.1: (SECTION 11: WORKING WITH ARRAY > IMPLEMENTING LOGIN)

Tiếp tục với Bankist app. Xử lý sự kiện Login

Hướng dẫn thực hiện: Xem video (Implementing login) hướng dẫn thực hành trên Udemey

```
let currentAccount;
btnLogin.addEventListener('click', function (e) {
  // Prevent form from submitting
  e.preventDefault();

  currentAccount = accounts.find(
    acc => acc.username === inputLoginUsername.value
  );
  console.log(currentAccount);

  if (currentAccount?.pin === Number(inputLoginPin.value)) {
    // Display UI and message
    labelWelcome.textContent = `Welcome back, ${
      currentAccount.owner.split(' ')[0]
    }`;
    containerApp.style.opacity = 100;

    // Clear input fields
    inputLoginUsername.value = inputLoginPin.value = '';
    inputLoginPin.blur();

    // Update UI
    updateUI(currentAccount);
  }
});
```

```
});
```

LAB4.2: (SECTION 11: WORKING WITH ARRAY > IMPLEMENTING TRANSFERS)

Tiếp tục với Bankist app. Xử lý sự kiện Transfers

Hướng dẫn thực hiện: Xem video (Implementing transfers) hướng dẫn thực hành trên Udemy

```
btnTransfer.addEventListener('click', function (e) {
  e.preventDefault();
  const amount = Number(inputTransferAmount.value);
  const receiverAcc = accounts.find(
    acc => acc.username === inputTransferTo.value
  );
  inputTransferAmount.value = inputTransferTo.value = '';

  if (
    amount > 0 &&
    receiverAcc &&
    currentAccount.balance >= amount &&
    receiverAcc?.username !== currentAccount.username
  ) {
    // Doing the transfer
    currentAccount.movements.push(-amount);
    receiverAcc.movements.push(amount);

    // Update UI
    updateUI(currentAccount);
  }
});
```

LAB4.3: (SECTION 11: WORKING WITH ARRAY > ARRAY METHOD PRACTICE)

Thực hiện code theo ví dụ sau

```
//1.
const bankDepositSum = accounts
  .flatMap(acc => acc.movements)
  .filter(mov => mov > 0)
  .reduce((sum, cur) => sum + cur, 0);
```

```
console.log(bankDepositSum);

// 2.
// const numDeposits1000 = accounts
//   .flatMap(acc => acc.movements)
//   .filter(mov => mov >= 1000).length;

const numDeposits1000 = accounts
  .flatMap(acc => acc.movements)
  .reduce((count, cur) => (cur >= 1000 ? ++count : count), 0);

console.log(numDeposits1000);

// Prefixed ++ operator
let a = 10;
console.log(++a);
console.log(a);

// 3.
const { deposits, withdrawals } = accounts
  .flatMap(acc => acc.movements)
  .reduce(
    (sums, cur) => {
      // cur > 0 ? (sums.deposits += cur) : (sums.withdrawals += cur);
      sums[cur > 0 ? 'deposits' : 'withdrawals'] += cur;
      return sums;
    },
    { deposits: 0, withdrawals: 0 }
  );

console.log(deposits, withdrawals);

// 4.
// this is a nice title -> This Is a Nice Title
const convertTitleCase = function (title) {
  const capitalize = str => str[0].toUpperCase() + str.slice(1);

  const exceptions = ['a', 'an', 'and', 'the', 'but', 'or', 'on', 'in', 'with'];

  const titleCase = title
    .toLowerCase()
```

```
        .split(' ')
        .map(word => (exceptions.includes(word) ? word : capitalize(word)))
        .join(' ');

    return capitalize(titleCase);
};

console.log(convertTitleCase('this is a nice title'));
console.log(convertTitleCase('this is a LONG title but not too long'));
console.log(convertTitleCase('and here is another title with an EXAMPLE'));
```

--- Hết ---