

# Cài đặt Linked List - Some and Every

---

```
// 12-07: Linked List
// - some(isValidFn)
// - every(isValidFn)
function some(isValidFn) {
  if (head == null) return false;

  let current = head;
  while (current != null) {
    if (isValidFn(current.data)) return true;

    // move to the next item
    current = current.next;
  }

  return false;
}

function every(isValidFn) {
  if (head == null) return false;

  let current = head;
  while (current != null) {
    if (!isValidFn(current.data)) return false;

    // move to the next item
    current = current.next;
  }

  return true;
}
```

## Unit Test

```
describe('some(isValidFn)', () => {
  it('should return false if list is empty', () => {
    const numberLinkedList = createLinkedList();
    expect(numberLinkedList.some((x) => x > 0)).toBe(false);
  });

  it('should return false if not existed in list', () => {
    // 1 -> 2 -> 3
    const numberLinkedList = createLinkedList();
    numberLinkedList.insertHead(3);
    numberLinkedList.insertHead(2);
    numberLinkedList.insertHead(1);

    expect(numberLinkedList.some((x) => x > 3)).toBe(false);
    expect(numberLinkedList.some((x) => x % 5 === 0)).toBe(false);
    expect(numberLinkedList.some((x) => x < 0)).toBe(false);
  });

  it('should return true if existed in list', () => {
    const numberLinkedList = createLinkedList();
    numberLinkedList.insertHead(3);
    numberLinkedList.insertHead(2);
    numberLinkedList.insertHead(1);

    expect(numberLinkedList.some((x) => x % 2 === 0)).toBe(true);
    expect(numberLinkedList.some((x) => x > 1)).toBe(true);
    expect(numberLinkedList.some((x) => x < 3)).toBe(true);
  });
});
```

```
describe('every(isValidFn)', () => {
  it('should return false if list is empty', () => {
    const numberLinkedList = createLinkedList();
    expect(numberLinkedList.every((x) => 0)).toBe(false);
  });

  it('should return false if not all nodes match condition', () => {
    const numberLinkedList = createLinkedList();
    numberLinkedList.insertHead(15);
    numberLinkedList.insertHead(10);
    numberLinkedList.insertHead(5);

    expect(numberLinkedList.every((x) => x % 3 === 0)).toBe(false);
    expect(numberLinkedList.every((x) => x <= 10)).toBe(false);
  });

  it('should return true if all nodes match condition', () => {
    const numberLinkedList = createLinkedList();
    numberLinkedList.insertHead(15);
    numberLinkedList.insertHead(10);
    numberLinkedList.insertHead(5);

    expect(numberLinkedList.every((x) => x % 5 === 0)).toBe(true);
    expect(numberLinkedList.every((x) => x > 1)).toBe(true);
  });
});
```

---

## Khoá học Javascript cho người mới bắt đầu 2021 🎉

- Tác giả: **Hậu Nguyễn** - Founder Easy Frontend
- Khoá học chỉ được published trên Udemy, không thông qua trung gian.
- Khoá học không bán dạng videos upload trên Google Drive hay bất cứ hình thức nào tương tự.
- Khoá học có nhóm discord để hỗ trợ trong quá trình học tập.

☎️ Liên hệ tác giả để được hỗ trợ:

- ✅ Facebook: <https://www.facebook.com/nvhauesmn/>
- ✅ Fanpage: <https://www.facebook.com/learn.easyfrontend>
- ✅ Youtube Channel: <https://www.youtube.com/easyfrontend>