

Cài đặt Linked List - Remove

```
// 12-06: Linked List
// - removeHead()
// - removeTail()
// - removeAtPosition(position)
function removeHead() {
  if (head !== null) head = head.next;

  return head;
}

function removeTail() {
  if (head === null) return null;

  // has only 1 item
  if (head.next === null) {
    head = null;
    return head;
  }

  let secondLast = head;
  while (secondLast.next.next !== null) {
    secondLast = secondLast.next;
  }

  secondLast.next = null;
  return head;
}
```

```
function removeAtPosition(position) {  
  if (position <= 0 || head == null) return removeHead();  
  
  // has only 1 item  
  if (head.next == null) {  
    head = null;  
    return head;  
  }  
  
  let prev = head;  
  let curr = head;  
  let i = 0;  
  
  while (curr.next !== null && i < position) {  
    prev = curr;  
    curr = curr.next;  
    i++;  
  }  
  
  prev.next = curr.next;  
  return head;  
}
```

Unit Test

```
describe('removeHead()', () => {
  it('should return null if list is empty', () => {
    const numberLinkedList = createLinkedList();
    const head = numberLinkedList.removeHead();

    expect(head).toBeNull();
  });

  it('should return null if list has one item', () => {
    const numberLinkedList = createLinkedList();
    numberLinkedList.insertHead(5);
    const head = numberLinkedList.removeHead();

    expect(head).toBeNull();
  });

  it('should return one item if list has two items', () => {
    // 5 -> 10
    const numberLinkedList = createLinkedList();
    numberLinkedList.insertHead(10);
    numberLinkedList.insertHead(5);
    const head = numberLinkedList.removeHead();

    expect(head).toEqual({ data: 10, next: null });
  });
});
```

```
describe('removeTail()', () => {
  it('should return null if list is empty', () => {
    const numberLinkedList = createLinkedList();
    const head = numberLinkedList.removeTail();

    expect(head).toBeNull();
  });

  it('should return null if list has one item', () => {
    const numberLinkedList = createLinkedList();
    numberLinkedList.insertHead(5);
    const head = numberLinkedList.removeTail();

    expect(head).toBeNull();
  });

  it('should return one item if list has two items', () => {
    // 5 -> 10
    const numberLinkedList = createLinkedList();
    numberLinkedList.insertHead(10);
    numberLinkedList.insertHead(5);
    const head = numberLinkedList.removeTail();

    expect(head).toEqual({ data: 5, next: null });
  });
});
```

```
describe('removeAtPosition(position)', () => {
  it('should return null if list is empty', () => {
    const numberLinkedList = createLinkedList();
    const head = numberLinkedList.removeAtPosition(0);

    expect(head).toBeNull();
  });

  it('should return null if list has one item', () => {
    const numberLinkedList = createLinkedList();
    numberLinkedList.insertHead(5);
    const head = numberLinkedList.removeAtPosition(1);

    expect(head).toBeNull();
  });

  it('should remove correctly item at 0 <= position < n - 1', () => {
    // 0 -> 5 -> 10
    const numberLinkedList = createLinkedList();
    numberLinkedList.insertHead(10);
    numberLinkedList.insertHead(5);
    numberLinkedList.insertHead(0);
    const head = numberLinkedList.removeAtPosition(1);

    expect(head).toEqual({ data: 0, next: { data: 10, next: null } });
  });
});
```

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>