

Cài đặt Linked List - Insert

```
// 12-05:
// - insertTail(newData)
// - insertBeforePosition(newData, position)
function createLinkedList() {
  let head = null;

  function getTail() {
    // implemented from previous video
  }

  function insertHead(newData) {
    // implemented from previous video
  }

  function insertTail(newData) {
    // your code here ...
  }

  function insertBeforePosition(newData, position) {
    // your code here ...
  }

  return {
    getTail,
    insertHead,
    insertTail,
    insertBeforePosition,
  };
}
```

```
function insertTail(newData) {
  const newNode = {
    data: newData,
    next: null,
  };

  if (head == null) {
    head = newNode;
  } else {
    const tail = getTail();
    tail.next = newNode;
  }

  return head;
}
```

```
function insertBeforePosition(newData, position) {
  if (position <= 0 || head == null) {
    insertHead(newData);
    return head;
  }

  let prev = head;
  let curr = head;
  let i = 0;

  while (curr != null && i < position) {
    prev = curr;
    curr = curr.next;
    i++;
  }

  // found node to insert
  const newNode = {
    data: newData,
    next: curr,
  };
  prev.next = newNode;

  return head;
}
```

Unit Test

```
describe('insertTail(data)', () => {  
  it('should return head with one node if list is empty', () => {  
    const numberLinkedList = createLinkedList();  
    const head = numberLinkedList.insertTail(5);  
  
    expect(head).toEqual({ data: 5, next: null });  
  });  
  
  it('should return head with two node if list has one node', () => {  
    const numberLinkedList = createLinkedList();  
    numberLinkedList.insertTail(5);  
    const head = numberLinkedList.insertTail(10);  
  
    expect(head).toEqual({ data: 5, next: { data: 10, next: null } });  
  });  
});
```

```
describe('insertBeforePosition(data, position)', () => {
  it('should insert head if position <= 0', () => {
    expect(
      createLinkedList().insertBeforePosition(5, -1)
    ).toEqual({ data: 5, next: null });

    expect(
      createLinkedList().insertBeforePosition(5, 0)
    ).toEqual({ data: 5, next: null });
  });

  it('should insert tail if position >= size', () => {
    const numberLinkedList = createLinkedList();
    numberLinkedList.insertBeforePosition(5, -1);
    const head = numberLinkedList.insertBeforePosition(10, 5);

    expect(head).toEqual({ data: 5, next: { data: 10, next: null } });
  });

  it('should insert before position if 0 <= position < size', () => {
    const numberLinkedList = createLinkedList();
    numberLinkedList.insertBeforePosition(5, 1);
    numberLinkedList.insertBeforePosition(10, 2);
    numberLinkedList.insertBeforePosition(15, 3);
    const head = numberLinkedList.insertBeforePosition(0, 1);

    expect(head).toEqual({
      data: 5,
      next: { data: 0, next: { data: 10, next: { data: 15, 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>