# Get  ⭐ Premium

**Yangshun Tay** 🔗
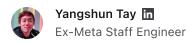Ex-Meta Staff Engineer          </> **JS** **TS**          🔥 Easy          🕐 10mins          ✅ 4.96k done

Before the optional chaining operator ( `?.` ) existed, it was sometimes troublesome to access deeply-nested properties in huge JavaScript objects when some of the intermediate properties might not be present.

```
1   const john = {
2     profile: {
3       name: { firstName: 'John', lastName: 'Doe' },
4       age: 20,
5       gender: 'Male',
6     },
7   };
8
9   const jane = {
10    profile: {
11      age: 19,
12      gender: 'Female',
13    },
14  };
15
16  function getFirstName(user) {
17    return user.profile.name.firstName;
18  }
```

Doing `getFirstName(john)` works but `getFirstName(jane)` will error because the `name` property doesn't exist for `jane.profile` .

## Lodash's `_.get` method

Lodash's `_.get` method was created as a solution for such use cases.

Let's write our own version as a `get` function. The function gets the value at `path` of `object` . If the resolved value is `undefined` , the `defaultValue` is returned in its place. The function signature is as such:

```
get(object, path, [defaultValue]);
```

- `object` : The object to query.
- `path` : The path of the property to get. It can be a string with `.` as the separator between fields, or an array of path strings.
- `defaultValue` : Optional parameter. The value returned if the resolved value is `undefined` .

## Examples

```
get(john, 'profile.name.firstName'); // 'John'
get(john, 'profile.gender'); // 'Male'
get(jane, 'profile.name.firstName'); // undefined
```

Arrays can also be accessed if numerical indices are provided.

```
get({ a: [{ b: { c: 3 } }] }, 'a.0.b.c'); // 3
```
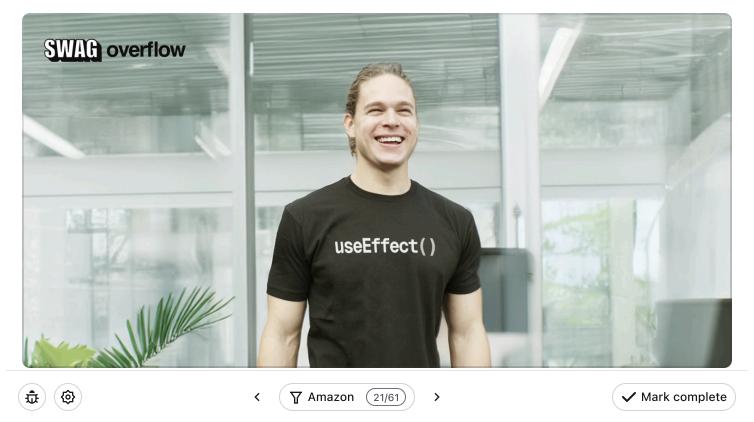
There's no need to support syntax resembling `get(object, 'a[0].b.c')` .

## Asked at these companies

Tiktok    Amazon    Tiktok

⚠ Coding and submission is not supported on mobile devices. Use a wider screen to practice solving this question within the editor.

View the solution

Amazon 21/61     ✓ Mark complete

## The useEffect() Tee — Mount, Update, Rerender, Repeat

Nothing screams React struggle like debugging infinite re-renders. This sleek black tee pays tribute to every front end dev's favorite hook — the one that either saves your app or breaks everything. Perfect for React engineers who live in the dependency array.

**Get yours now ->**