# undefsafe

Simple *function* for retrieving deep object properties without getting "Cannot read property 'X' of undefined"

Can also be used to safely set deep values.

## Usage

var object = {

a: {

b: {

c: 1,

d: [1,2,3],

e: 'remy'

}

}

};

console.log(undefsafe(object, 'a.b.e')); // "remy"

console.log(undefsafe(object, 'a.b.not.found')); // undefined

Demo: <https://jsbin.com/eroqame/3/edit?js,console>

## Setting

var object = {

a: {

b: [1,2,3]

}

};

// modified object

var res = undefsafe(object, 'a.b.0', 10);

console.log(object); // { a: { b: [10, 2, 3] } }

console.log(res); // 1 - previous value

## Star rules in paths

As of 1.2.0, undefsafe supports a \* in the path if you want to search all of the properties (or array elements) for a particular element.

The function will only return a single result, either the 3rd argument validation value, or the first positive match. For example, the following github data:

const githubData = {

commits: [{

modified: [

"one",

"two"

]

}, /\* ... \*/ ]

};

// first modified file found in the first commit

console.log(undefsafe(githubData, 'commits.\*.modified.0'));

// returns `two` or undefined if not found

console.log(undefsafe(githubData, 'commits.\*.modified.\*', 'two'));