

Suprasti kas yra currying ir partial application. Turėkit galvoje, kad lambda abstraction visada yra curried, t. y. Lambda funkcijos visada priima tik po vieną parametą. Funkcinėse programavimo kalbose keli argumentai funkcijai yra syntactic sugar. Curried funkcijos leidžia jas panaudoti kaip argumentus kitoms funkcijoms, pvz map ir filter, apie tai trumpai kitoje paskaitoje.

Naudodami Javascript, parašykite funkciją, kuri bet kuriai 2jų argumentų funkcijai atliktų *currying*.

t. y.

Funckcija curry2 turi atlikti:

jeigu turime funkciją $sum(a,b)$, iškviečiama kaip $sum(4,5)$, tai

```
let f2 = curry2(sum);
```

f2 iškviečiama kaip $f2(4)(5)$

Žinant, kad javascripte galima gauti visus argumentus kaip masyvą, bet to galima išimti iš masyvo elementus po vieną (žr. pop arba unshift) mutuoiant masyvą, pabandykite parašyti $curry_n$ bet kuriam parametru skaičiui (patys, ne kopijuodami)

Literatūra:

<https://medium.com/nexa-digital/currying-and-partial-application-composition-made-easier-30aadaa54b5f>