



Resurse suplimentare
pe care recomand să le parcurgi

Destructuring Assignment

Destructuring allows you to extract values from arrays or properties from objects into distinct variables with a concise syntax.

Destructurarea Array-urilor

```
const rgb = [255, 100, 80]

// Fără destructuring
const red = rgb[0]
const green = rgb[1]
const blue = rgb[2]

// Cu destructuring
const [red, green, blue] = rgb

console.log(red, green, blue) // 255 100 80
```

Skipping Values & Default Values

```
const coordinates = [12, 8]

const [x, , z = 0] = coordinates
// x = 12, z = 0 (valoare implicită)
```

Destructurarea Obiectelor



```
const user = {
  id: 42,
  name: 'Ana',
  age: 27,
  location: {
    city: 'București',
    country: 'RO',
  },
}

// Fără destructuring
const userName = user.name
const city = user.location.city

// Cu destructuring
const {
  name: fullName,
  location: { city },
} = user

console.log(fullName, city) // Ana București
```

Renaming & Default Values

```
const { role = 'guest', age: userAge } = user
console.log(role, userAge) // guest 27
```

Nested Destructuring în React (props/state)

```
const Profile = ({
  user: {
    name,
    location: { city },
  },
}) => <p>`${name} din ${city}`</p>
```

Operatorul Spread (...)

Spread operatorul „desface” un array sau un obiect, oferind o modalitate simplă de copiere superficială (shallow copy) și de combinare a colecțiilor.



Copiere și Combinare Array-uri

```
const numbers = [1, 2, 3]
const moreNumbers = [...numbers, 4, 5] // [1,2,3,4,5]
```

Copiere și Extindere Obiecte

```
const product = { name: 'Laptop', price: 2500 }
const updated = { ...product, price: 2300, stock: 5 }
```

Actualizarea Immutabilă a State-ului în React

```
const [todos, setTodos] = useState([
  { id: 1, text: 'Learn React', completed: false },
])

const toggleTodo = (id) => {
  setTodos((prev) =>
    prev.map((todo) =>
      todo.id === id ? { ...todo, completed: !todo.completed } : todo
    )
  )
}
```

Operatorul Rest (...)

Rest operatorul colectează elementele rămase într-un array sau obiect.

Parametrii Variadici

```
const sum = (...numbers) => numbers.reduce((acc, n) => acc + n, 0)
console.log(sum(1, 2, 3, 4)) // 10
```

Rest în Destructuring Obiecte

```
const { id, ...userInfo } = user
console.log(id) // 42
console.log(userInfo) // { name: 'Ana', age: 27, location: { ... } }
```



Combinăția Spread & Rest

```
// Clonare adâncime superficială + modificare proprietate  
const newUser = { ...user, age: 28 }  
  
// Eliminarea unei proprietăți  
const { age, ...userWithoutAge } = newUser
```

Studii de Caz în React

1. Merge props: `const props = { ...defaultProps, ...customProps }`
2. Pass-through props: `{...rest}` într-un component wrapper
3. Formulare: actualizarea câmpurilor `setForm({ ...form, [name]: value })`