



# Destrutturazione

# DESTRUTTURAZIONE - ARRAY

---

- Con il costrutto di destrutturazione è possibile assegnare a delle variabili valori di array e oggetti.

```
let myArray = [1, 2, 3];  
let [a, b, c] = myArray;
```

- in caso di variabili ri-assegnabili (`let`) è possibile fare una assegnazione separata dalla dichiarazione:

```
let a, b;  
[a, b] = [1, 2];
```

# DESTRUTTURAZIONE - ARRAY

---

- Si può assegnare un valore di default alle variabili, che sostituisce solo i valori `undefined`:

```
const arr = [1, undefined];  
const [a = 5, b = 7, c = 9] = arr;  
a; // 1 - arr[0] non è undefined, il default viene scartato  
b; // 7 - arr[1] è undefined, quindi usa il default  
c; // 9 - arr non ha un terzo valore, usa il default
```

# DESTRUTTURAZIONE - ARRAY

---

- un uso raro, ma usato spesso come esempio della destrutturazione, è il *variable swap*, ovvero scambiare i valori tra variabili.

```
let a = 1;  
let b = 3;  
  
[a, b] = [b, a];  
  
a; // 3  
b; // 1
```

anzichè dover passare per una variabile intermedia

```
let a = 1;  
let b = 3;  
let c = a;  
a = b;  
b = c;
```

# DESTRUTTURAZIONE - ARRAY

---

- è possibile ignorare dei valori ritornati lasciando degli “spazi vuoti”

```
const arr = [1, 2, 3];  
const [a, , b] = arr;  
a; // 1  
b; // 3
```

**Sebbene possibile, vi sconsigliamo questo costrutto.** Il motivo sta nel fatto che la doppia virgola può sembrare un errore di battitura.

Vi consigliamo in questi casi di utilizzare l'approccio tradizionale:

```
const arr = [1, 2, 3];  
const a = arr[0];  
const b = arr[1];  
a; // 1  
b; // 3
```

# DESTRUTTURAZIONE - ARRAY

---

- è possibile assegnare la parte rimanente dell'array con la sintassi `...`, il **Rest operator**.

```
const arr = [1, 2, 3, 4, 5];  
const [first, ...rest] = arr;  
first; // 1  
rest; // [2, 3, 4, 5]
```

Nel caso non ci siano elementi a sufficienza nell'array, verrà assegnato un array vuoto:

```
const arr = [1, 2];  
const [first, second, ...rest] = arr;  
first; // 1  
second; // 2  
rest; // []
```

Attenzione, l'assegnazione “rest” **deve assolutamente** essere l'ultima:

```
const [first, ...mid, last] = arr;  
// SyntaxError!!
```

# DESTRUTTURAZIONE - OGGETTI

---

- Esiste la destrutturazione anche per gli oggetti:

```
const obj = { a: "foo", b: "bar" };  
const { a, b } = obj;  
a; // "foo"
```

- è possibile fare una assegnazione separata dalla dichiarazione

```
const obj = { a: 1, b: 2 };  
let a, b;  
  
({ a, b } = obj);
```

## ATTENZIONE

le parentesi in questo caso sono necessarie altrimenti { viene interpretato come inizio di un Block Statement!

# DESTRUTTURAZIONE - OGGETTI

---

- è possibile fare una assegnazione con nuovi nomi

```
const obj = { a: 1, b: 2 };  
const { a, b: differentName } = obj;  
a; // 1  
differentName; // 2  
b; // ReferenceError!! La variabile non viene creata
```

È un costrutto potenzialmente confusionario, se possibile evitatelo. Non è raro incontrarlo in codice React (durante la estrazione delle props di un componente).



# DESTRUTTURAZIONE - OGGETTI

---

- è possibile assegnare valori di default alle variabili

```
const obj = { a: 1, b: undefined };  
let { a = 5, b = 7, c = 9 } = obj;  
a; // 1 - obj.a non è undefined, il default viene scartato  
b; // 7 - obj.b è undefined, quindi usa il default  
c; // 9 - obj non ha una proprietà "c", usa il default
```

# DESTRUTTURAZIONE - OGGETTI

---

DOMANDA: come correggiamo il codice sottostante in modo che scriva sulla console «Ciao Mario Rossi»?

Attenzione, potete sostituire solo il commento `/* ? */`!

```
function saluta(/* ? */) {  
  console.log(`Ciao ${nome} ${cognome}!`);  
}  
  
let persona = {  
  nome: "Mario",  
  cognome: "Rossi"  
};  
  
saluta(persona);
```

- `nome, cognome`
- `[ nome, cognome ]`
- `{ nome, cognome }`
- Nessuna delle precedenti
- Non è possibile, dobbiamo modificare di più il codice