

# Shallow Copy vs. Deep Copy

---

# Shallow copy

---

طريقه عمل ال Shalow copy هي انها بتنسخ فقط اول ليفل من Object

زى المتغيرات العاديه التي تحمل ارقام او حروف او حتى جمل

وممكن استعمال هذه الخاصيه عن طريق ال assign operator او ال `Object.assign()`



# مثال

---

```
const obj = { name: 'Version 1', additionalInfo: { version: 1 } };
```

```
const shallowCopy1 = { ...obj };
```

```
const shallowCopy2 = Object.assign({}, obj);
```

```
shallowCopy1.name = 'Version 2';
```

```
shallowCopy1.additionalInfo.version = 2;
```

## مثال

---

```
shallowCopy2.name = 'Version 2';  
shallowCopy2.additionalInfo.version = 2;  
console.log(obj); // { name: 'Version 1', additionalInfo: {  
  version: 2 } }  
console.log(shallowCopy1); // { name: 'Version 2',  
  additionalInfo: { version: 2 } }  
console.log(shallowCopy2); // { name: 'Version 2',  
  additionalInfo: { version: 2 } }
```

---

وكما يتضح من المثال السابق انهم عندما حدثنا  
خاصيه في المستوى الأول تم تحديث  
الـ Object الأصلي



# Deep copy

---

طريقه عمل ال deep copy هي انها بتنسخ كل ليفل من Object

arrays المتغيرات العاديه التي تحمل ارقام او حروف او حتى جمل او حتى ال

وممكن استعمال هذه الخاصيه عن طريق ال `JSON.parse() + JSON.stringify()`

# مثال

---

```
const obj = { name: 'Version 1', additionalInfo: { version: 1 } };
const deepCopy = JSON.parse(JSON.stringify(obj));
deepCopy.name = 'Version 2';
deepCopy.additionalInfo.version = 2;

console.log(obj); // { name: 'Version 1', additionalInfo: {
version: 1 } }
console.log(deepCopy); // { name: 'Version 2', additionalInfo: {
version: 2 } }
```

---

وكما يتضح من المثال السابق انهم عندما حدثنا خاصيه في المستوى الأول لم يتم تحديث  
الـ Object الأصلي