مختصر كتابي

Destructuring, Spread, and Immutability



كيف تصنع نسخة معدلة من كائن دون أن يتغير الأصل؟

تخيل ورقة مكتوبًا فيها اسم وجبة ورقمها، وتريد نسخة للغداء برقم مختلف دون أن تتغير ورقة الفطار. إذا كتبت أن متغيرًا جديدًا يساوي الموجود، فأنت تمسك الكائن نفسه لا نسخة منه. وهذه النقطة ستحتاجها كثيرًا مع الحالة (State) في React.

```
const original = { name: "فول", calories: 250 };
const changed = { ...original, calories: 280 };
export default function App() {
  const { name, calories } = changed;
  return (
    <main dir="rtl">
      <p>الأصل: {original.name} {original.calories}</p>
      <p>النسخة: {name} {calories}</p>
    </main>
  );
}
```

الأقواس المعقوفة في السطر الثاني تصنع كائنًا جديدًا، وعلامة **النشر** (Spread) `...original` تنقل حقول الأصل إليه. كأنك كتبت `name` و `calories` القديمين بيدك، ثم كتبت `calories` مرة أخرى بقيمة `280`. آخر كتابة هي التي تحدد قيمة الحقل، ولذلك تعرض المعاينة الأصل `فول 250` والنسخة `فول 280`، مع أن الاسم لم يُكتب في تعريف النسخة. ولو عكست الترتيب لعاد الرقم القديم.

في `App` نستخدم **التفكيك** (Destructuring) لنأخذ `name` و `calories` من `changed` في متغيرين. هذا قراءة للحقول لتسهيل العرض، وليس عملية النسخ نفسها؛ لو حذفته وكتبت `changed.name` في الفقرة لبقيت النتيجة كما هي. الذي حمى الأصل هو إنشاء كائن جديد.

أما لو كتبت `const changed = original` ثم غيّرت `changed.name`، فإن الاسمين يشيران إلى الكائن نفسه، فيتغير سطر الأصل أيضًا. لهذا لا نكتب على `original` مباشرة، بل نصنع كائنًا جديدًا، كي تستطيع الواجهة لاحقًا أن تفرق بين البيانات القديمة والجديدة.

- التفكيك يأخذ حقولًا من كائن ويضع كل حقل في متغير.
- النشر ينقل الحقول إلى كائن جديد، والقيمة المكتوبة أخيرًا هي التي تبقى.
- الإسناد المباشر لا ينسخ؛ المتغيران يشيران إلى الكائن نفسه.
- هذه نسخة سطحية (Shallow Copy): الكائن الداخلي يبقى مشتركًا ما لم تنسخه هو أيضًا.
- `const` تمنع ربط الاسم بقيمة أخرى، لكنها لا تمنع الكتابة داخل الكائن.

جرّب في المحرر تحت الدرس: اصنع نسخة اسمها `عدس` ورقمها `300`، وتأكد أن الأصل ما زال `فول 250`.

الخلاصة: انسخ الحقول إلى كائن جديد ثم اكتب التغيير بعدها، ولا تكتب على الأصل، فيبقى القديم والجديد كلاهما واضحًا.