

مختصر كتابي: For, forIn, forOf and while loops

أنواع الحلقات التكرارية

for, for...in, for...of, while

**for**

```
for (let i = 0; i < 10; i++)
```

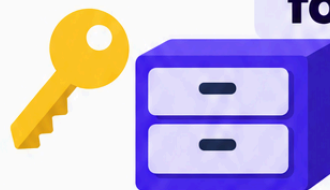
البداية

الشرط

الزيادة

for...of

تمر على القيم

for...in

تمر على خصائص الكائن

**while**

كرر طالما الشرط متحقق

انتبه للحلقة اللانهائية ∞

do



while

do...while

تنفذ مرة واحدة على الأقل



غير العداد دائما داخل الحلقة

مختصر كتابي: For, forIn, forOf and while loops

ما أنواع الحلقات التكرارية في JavaScript؟

في هذا الدرس نتعرف على أشكال أخرى من **الحلقات التكرارية (Loops)**: `for` و `for...in` و `for...of` و `while` و `do...while`. قد تكون أقل شيوعًا من الـ `methods` التي عرفناها حتى الآن، لكنها **ما زالت مستخدمة** وستقابلها حتمًا في المشاريع التي ستقرأها أو تعمل عليها.

حلقة `for`

أشهر الأنواع وأقدمها في JavaScript. تبدأ بالكلمة `for`، وبين القوسين ثلاثة أجزاء يفصل بينها `;`:

```
for (let i = 0; i < 10; i++) {
  console.log(i); // 0, 1, 2 ... 9
}
```

1. **القيمة المبدئية (Initial expression)**: `let i = 0`، أي نبدأ من الصفر. نستخدم `let` وليس

`const` لأن قيمة `i` ستتغير في كل مرة.

2. **الشرط (Condition)**: `i < 10`، يُفحص **قبل** كل دورة، وطالما تحقق تُنفَّذ الأوامر التي بالداخل. لو

كتب `i <= 10` ستطبع من 0 إلى 10.

3. **الزيادة (Increment)**: `i++` تعني `i = i + 1`. عندما تصل `i` إلى 10 لا يتحقق الشرط فتتوقف

الحلقة.

لو حذفنا `i++` ستبقى `i` صفرًا للأبد وتدخل في **حلقة لا نهائية (Infinite loop)**، فيعلق الـ `tab` أو المتصفح كله.

مثال: جمع أرقام مصفوفة. نمشي على الـ `indexes` حتى `numbers.length - 1` لأن الـ `length` يبدأ العد من 1 بينما الـ `index` يبدأ من 0:

```
const numbers = [1, 2, 3, 4, 5];
let count = 0;

for (let i = 0; i <= numbers.length - 1; i++) {
  count += numbers[i];
}

console.log(count); // 15
```

- لو استخدمت `<` بدل `<=` مع `numbers.length - 1` فلن تمر الحلقة على آخر عنصر، ويكون الناتج 10 بدل 15.
- الزيادة لا يلزم أن تكون بواحد: مع `i += 2` تمر الحلقة على الـ 0 indexes و 2 و 4 فقط، ولو طبعت `count` داخلها ستظهر القيم 1 ثم 4 ثم 9. لكن الشائع هو `i++` لأنك غالبًا تريد المرور على المصفوفة كلها.

حلقة for...in للمرور على الـ Object

نستخدم `for...in` للمرور بسرعة على **خصائص الكائن (Object properties)**. في كل دورة يُوضع اسم الخاصية (الـ key) في المتغير، لذلك نستخدم `const`، فهو متغير جديد في كل دورة (block scope):

```
const person = { name: "Mohamed", age: 10 };

for (const key in person) {
  console.log(key);           // name, age
  console.log(person[key]); // Mohamed, 10
}

// نفس الفكرة باستخدام Object.keys
Object.keys(person).map((key) => console.log(key)); // name, age
```

- سمّ المتغير `key` وليس `i`، لأنه ليس index بل اسم خاصية.
- للوصول إلى القيمة نكتب `person[key]` بقوسين مربعين، فهي ليست استدعاء function.
- الطريقتان صحيحتان: `Object.keys` تحتاج تحويل الخصائص إلى array ثم المرور عليها، بينما `for...in` خطوة واحدة ويراهها كثيرون أسهل في القراءة. لكن الأشيع في الاستخدام هو `Object.keys` و `Object.values` و `Object.entries`.

حلقة for...of للمرور على القيم

تعمل `for...of` مع الأشياء **القابلة للتكرار (Iterable)**، أي التي لها `length` ويمكن المرور عليها، مثل الـ array و الـ string و الـ Map و الـ NodeList. وهي تعطيك كل item مباشرة، تمامًا مثل `forEach`:

```
const numbers = [1, 2, 3, 4, 5];

for (const item of numbers) {
  console.log(item); // 1, 2, 3, 4, 5
}

numbers.forEach((item) => console.log(item)); // 1, 2, 3, 4, 5
```

حلقة while وحلقة do...while

`while` تعني: **طالما الشرط متحقق، كرر**. مثل أن تقول لطفل: طالما المنبه لم يرن اجلس أمام التلفزيون، وعندما يرن أغلقه وقم. أو في البنك: طالما رقمك لم يظهر على الشاشة ابق في مكانك.

```
let count2 = 0;

while (count2 < 5) {
  console.log(`${count2} from while`);
  count2++; // لا تنس هذا السطر
}
```

السطر `count2++` يبدو بسيطًا لكنه مهم جدًا: بدونَه يظل `count2` صفرًا، والصفّر دائمًا أصغر من 5، فتدخل في حلقة لا نهائية ويعلق الـ tab ولا يستجيب حتى زر الإغلاق، فتضطر لإغلاقه بالقوة (kill).

أما `do...while` فهي ابنة عم `while`: نكتب الأوامر داخل `do` أولًا، ثم الشرط بعد `while`:

```
let count3 = 0;

do {
  console.log(`${count3} from do while`);
  count3++;
} while (count3 < 0);
// مرة واحدة رغم أن الشرط خاطئ (from do while يطبع: 0)

let count4 = 0;

while (count4 < 0) {
  console.log(`${count4} from while`); // لا يُطبع شيء
  count4++;
}
```

الفرق بسيط لكنه مهم، وقد يُسأل عنه في المقابلات (interview): في `while` يُفحص الشرط **أولًا** ثم تُنفَّذ الأوامر، أما في `do...while` فتُنفَّذ الأوامر **أولًا** ثم يُفحص الشرط، فيُضمن تنفيذ الكود **مرة واحدة على الأقل**.

الخلاصة

- `for`: قيمة مبدئية ثم شرط ثم زيادة، ونستخدم فيها `let` لأن العداد يتغير.

- `for...in` تمر على خصائص ال object، و `for...of` تمر على قيم ال array وال string وغيرها من ال iterables.
- `while` تكرر طالما الشرط متحقق، ولا بد أن تغيّر المتغير داخلها حتى لا تقع في حلقة لا نهائية.
- `do...while` تنفّذ الكود مرة واحدة على الأقل لأنها تفحص الشرط بعد التنفيذ.

مصطلحات الدرس

المصطلح	بالإنجليزية	المعنى
الحلقة التكرارية	Loop	تكرار تنفيذ مجموعة أوامر أكثر من مرة
القيمة المبدئية	Initial expression	المتغير الذي تبدأ به حلقة for مثل <code>let i = 0</code>
الزيادة	Increment	تغيير العداد في كل دورة مثل <code>i++</code>
الحلقة اللانهائية	Infinite loop	حلقة لا يتوقف شرطها أبدًا فيعلّق المتصفح
قابل للتكرار	Iterable	شيء يمكن المرور على عناصره مثل ال array وال string