



# For And While Loops And Do While

Status	In progress
Assign	
# Number	18

For And While السلام عليكم معاكم زياد شلبي وهنشرح النهارده الحلقات التكرارية او Loops

خليني في الاول اشرح ليك الحلقات التكرارية او اللوب بيعمل ايه كا كونسبت او مفهوم

الحلقات التكرارية دي مفهوم في البرمجة بنستخدمه عشان نعيد تنفيذ سلسلة من الأوامر عدد معين من المرات أو حتى يتحقق شرط معين. عمومًا، بنستخدم الحلقات التكرارية في البرمجة عشان ننفذ نفس المهمة تاني وتالت وأكثر، زي مثلاً معالجة قائمة بالعناصر أو تنفيذ عملية على كل عنصر في مصفوفة.

في "While" والحلقات الـ "For" بنقدر نستخدم نوعين من الحلقات التكرارية، الحلقات الـ بنحدد عدد معين من المرات اللي هنفذ فيها الحلقة، بينما في الحلقات الـ "For"، الحلقات الـ بنفذ الحلقة حتى يتحقق شرط معين، "While".

مثلاً، لو عايز تعمل برنامج يطبع أول ١٠ أعداد صحيحة، بنقدر نستخدم حلقة تكرارية من نوع عشان ننفذ العملية ١٠ مرات. ولو عايز تعمل برنامج يطبع كل الأعداد الصحيحة الزوجية "For"

عشان ننفذ العملية حتى يتم طباعة "While" بين ١ و ٢٠، بنقدر نستخدم حلقة تكرارية من نوع كل الأعداد الزوجية في هذا النطاق.

ده كان مفهوم النظري في الحلقات التكرارية تعاله نشوف الاكواد في الاول for خرينا ناخد ال for and while وتعاله نشوف بس اننا عندنا نوعين هما ال

## Instead of writing:

```
text += cars[0] + "<br>";
text += cars[1] + "<br>";
text += cars[2] + "<br>";
text += cars[3] + "<br>";
text += cars[4] + "<br>";
text += cars[5] + "<br>";
```

## You can write:

```
for (let i = 0; i < cars.length; i++) {
  text += cars[i] + "<br>";
}
```

# Different Kinds of Loops

JavaScript supports different kinds of loops:

- `for` - loops through a block of code a number of times
- `for/in` - loops through the properties of an object
- `for/of` - loops through the values of an iterable object
- `while` - loops through a block of code while a specified condition is true
- `do/while` - also loops through a block of code while a specified condition is true

## The For Loop

The `for` statement creates a loop with 3 optional expressions:

```
for (expression 1; expression 2; expression 3) {  
    // code block to be executed  
}
```

**Expression 1** is executed (one time) before the execution of the code block.

**Expression 2** defines the condition for executing the code block.

**Expression 3** is executed (every time) after the code block has been executed.

## Example

```
for (let i = 0; i < 5; i++) {  
    text += "The number is " + i + "<br>";  
}
```

## Example

```
let i = 2;  
let len = cars.length;  
let text = "";  
for (; i < len; i++) {  
    text += cars[i] + "<br>";  
}
```

وبعدھا نقول الفرق بينهم while كلو خلینا نخش علی ال loop کده ادیتک

## Syntax

```
while (condition) {  
    // code block to be executed  
}
```

## Example

```
while (i < 10) {  
    text += "The number is " + i;  
    i++;  
}
```

do while تعاله بقا نقول حاجه ثانيه اسمها ال

## Syntax

```
do {  
    // code block to be executed  
}  
while (condition);
```

## Example

```
do {  
    text += "The number is " + i;  
    i++;  
}  
while (i < 10);
```

## Comparing For and While

```
const cars = ["BMW", "Volvo", "Saab", "Ford"];  
let i = 0;  
let text = "";  
  
for (;cars[i];) {  
    text += cars[i];  
    i++;  
}
```

```
const cars = ["BMW", "Volvo", "Saab", "Ford"];  
let i = 0;  
let text = "";  
  
while (cars[i]) {  
    text += cars[i];  
    i++;  
}
```