

# إظهار وإخفاء الباسورد



عايزه اول ما اضغط على زر password يظهر الارقام مره ثانيه يختفي

تقدر تحقق ده باستخدام JavaScript من خلال تغيير نوع الإدخال بين `password` و `text` عند الضغط على الزر. إليك كود بسيط يحقق ذلك:

## ◆ كود HTML + JavaScript:

html

```
<!DOCTYPE html>
<html lang="ar">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>إظهار وإخفاء الباسورد</title>
</head>
<body>

  <input type="password" id="password" placeholder="أدخل كلمة المرور">
  <button id="togglePassword">👁️</button>

  <script>
    const passwordInput = document.getElementById("password");
    const toggleButton = document.getElementById("togglePassword");

    toggleButton.addEventListener("click", function () {
      if (passwordInput.type === "password") {
        passwordInput.type = "text"; // إظهار الباسورد
      } else {
        passwordInput.type = "password"; // إخفاء الباسورد
      }
    });
  </script>

</body>
</html>
```

## ◆ شرح الكود:

- عند الضغط على الزر، يتحقق الكود من نوع الإدخال:
  - إذا كان `password`، يتم تغييره إلى `text` ليظهر النص.
  - وإذا كان `text`، يتم تغييره إلى `password` ليتم إخفاؤه.
- يمكنك استبدال أيقونة الزر `👁️` بأيقونة أخرى تناسب التصميم.

💡 إذا كنت تستخدم Bootstrap، يمكنك تحسين المظهر بإضافة أيقونات. هل تريد مثال باستخدام Bootstrap؟ 😊



في تحديثات في جافا سكريبت في domand event

نعم، هناك العديد من التحديثات التي يمكنك حلها باستخدام DOM و Events في JavaScript لممارسة مهاراتك. إليك بعض التحديثات الممتعة التي يمكنك تجربتها:

## 🔥 تحديثات DOM & Events في JavaScript

### 1 تحديث تغيير لون الخلفية عند الضغط على زر

**المطلوب:** عند النقر على زر، يتغير لون خلفية الصفحة إلى لون عشوائي.  
**التلميح:** استخدم `Math.random()` لإنشاء لون عشوائي.

### 2 تحديث عداد النقرات

**المطلوب:** إنشاء زر عند الضغط عليه، يزيد العداد كل مرة بمقدار 1 ويعرض العدد داخل عنصر `p`.  
**التلميح:** استخدم `addEventListener("click", function)` لتحديث العداد.

### 3 تحديث إظهار وإخفاء عنصر

**المطلوب:** لديك `div` يحتوي على نص، ويوجد زر عند النقر عليه يتم إخفاء النص، وعند النقر مرة أخرى يتم إظهاره.  
**التلميح:** استخدم `style.display = "none"` لإخفاء العنصر، و `style.display = "block"` لإظهاره.

### 4 تحديث تغيير النص عند المرور بالماوس

**المطلوب:** عند مرور الماوس على عنصر `h1`، يتم تغيير النص إلى "مرحبًا!"، وعند إخراج الماوس يعود للنص الأصلي.  
**التلميح:** استخدم الأحداث `onmouseover` و `onmouseout`.

### 5 تحديث عداد تنازلي باستخدام DOM

**المطلوب:** قم بإنشاء عداد يبدأ من 10 وينقص كل ثانية حتى يصل إلى 0، ثم يعرض رسالة "انتهى الوقت!" بدل الرقم.  
**التلميح:** استخدم `setInterval()` لتحديث العدد.

### 6 تحديث إدخال النص وعرضه مباشرة

**المطلوب:** أنشئ `input` ومربع `div`. عند كتابة أي شيء في الـ `input`، يظهر مباشرة في `div`.  
**التلميح:** استخدم `input.addEventListener("input", function(e) { ... })` لالتقاط القيم المدخلة.

## 7 تحدي التحقق من قوة كلمة المرور

💡 **المطلوب:** عند إدخال كلمة مرور في `input`، يتم تغيير لون الإطار:

- **ضعيف** إذا كانت أقل من 6 أحرف.
- **متوسط** إذا كانت بين 6-10.
- **قوي** إذا كانت أكثر من 10 أحرف.

🔥 **التلميح:** استخدم `input.addEventListener("input", function() { ... })` و `classList.add`.

🔧 هل تريد حل أحد هذه التحديات معًا؟ أي تحدي تود البدء به؟ 😊



التحدي الاول: `let prenants = document.getElementById('prenents');`

```
prenants.addEventListener('click', function () {
  let randomColor = `#${Math.floor(Math.random() * 16777215).toString(16)}`;
  prenants.style.backgroundColor = randomColor;
  console.log(randomColor);
});
```

الكود يحتوي على خطأ بسيط في `toString(16)`، حيث أن `toString` يجب أن تكون بحروف صغيرة بالكامل (`toString` وليس `tostring`).  
إليك الكود الصحيح مع بعض التحسينات:

✅ **الكود المصحح:**

javascript

```
let prenants = document.getElementById("prenents");

prenants.addEventListener("click", function () {
  let randomColor = `#${Math.floor(Math.random() * 16777215).toString(16)}`;
  prenants.style.backgroundColor = randomColor;
  console.log(randomColor);
});
```

🔥 **تحسين إضافي (إضافة زر لتغيير الخلفية)**

إذا كنت تريد تطبيقه على خلفية الصفحة بالكامل أو زر معين، استخدم الكود التالي:

html

```
<!DOCTYPE html>
<html lang="ar">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>تغيير لون الخلفية</title>
```

```

</head>
<body>

  <button id="prenants">غيّر اللون 🎨</button>

  <script>
    let prenants = document.getElementById("prenants");

    prenants.addEventListener("click", function () {
      let randomColor = `#${Math.floor(Math.random() * 16777215).toString(16)}`;
      document.body.style.backgroundColor = randomColor;
      console.log(randomColor);
    });
  </script>

</body>
</html>

```

## 🎯 ماذا أضفنا؟

- ✓ أصلحنا الخطأ في `toString(16)`.
- ✓ أضفنا زر لتغيير الخلفية بدلاً من عنصر غير محدد.
- ✓ جعلنا اللون يغير الخلفية الكاملة للصفحة بدلاً من زر فقط.
- ♦ جربه الآن، وإذا أردت تحسينات أخرى، أخبرني! 🚀



```

التحدي 2-(`btn`) document.getElementById(btn);let
;let p = document.getElementById(`p`)
;let clickCount = 0

} () btn.addEventListener(`click`, function
; ++clickCount
;p.textContent = clickCount
;({

```

الكود الخاص بك صحيح ويعمل بدون مشاكل! 🎯  
لكن سأقدم لك بعض التحسينات والإضافات لجعله أكثر احترافية.

## ✓ الكود المحسّن مع HTML

```

html

<!DOCTYPE html>
<html lang="ar">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>عداد النقرات</title>
  <style>
    body {
      text-align: center;
      font-family: Arial, sans-serif;
      margin-top: 50px;
    }
    button {

```

```
padding: 10px 20px;
font-size: 18px;
cursor: pointer;
}
p {
font-size: 24px;
font-weight: bold;
}
</style>
</head>
<body>

<button id="btn">اضغط هنا 🚀</button>
<p id="p">0</p>

<script>
let btn = document.getElementById("btn");
let p = document.getElementById("p");
let clickCount = 0;

btn.addEventListener("click", function () {
clickCount++;
p.textContent = clickCount;
});
</script>

</body>
</html>
```

## 🎯 ما الجديد في هذا الكود؟

- ✓ أضفنا تنسيق CSS لجعل الزر والنص أكثر وضوحًا.
- ✓ تأكدنا من أن العداد يبدأ من 0 ويزيد عند كل نقرة.
- ✓ استخدمنا نصًا عربيًا ليكون أكثر وضوحًا للمستخدمين.

♦ **تحسين إضافي:** إذا كنت تريد زرًا لإعادة تعيين العداد إلى الصفر، يمكن إضافة التالي:

html

```
<button id="reset">إعادة التعيين 🔄</button>
```

ثم أضف الكود التالي داخل `<script>`:

javascript

```
let resetBtn = document.getElementById("reset");
resetBtn.addEventListener("click", function () {
clickCount = 0;
p.textContent = clickCount;
});
```

🔥 جرب الكود الآن وأخبرني إن كنت تحتاج إلى أي تعديلات إضافية! 🚀



```
التحدي(`prents` = document.getElementById(`prents`);
let p = document.getElementById(`p`);
let btn = document.getElementById(`btn`);
```

```

} () btn.addEventListener(`click`, function
} if (prenents.style.display === "none")
; "prenents.style.display = "block
; btn.textContent = "lubna
} else {
; "prenents.display = "none
; btn.textContent = "lubna
{
;({

```

هناك خطأ بسيط في الكود الخاص بك، وهو أنك كتبت `prenents.display = "none"` بينما الصحيح هو:

javascript

```

prenents.style.display = "none";

```

يجب إضافة `style.` عند تغيير `display`.

## الكود المصحح مع تحسينات

html

```

<!DOCTYPE html>
<html lang="ar">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>إظهار وإخفاء العنصر</title>
  <style>
    #prenents {
      padding: 20px;
      background-color: lightblue;
      text-align: center;
      font-size: 20px;
      font-weight: bold;
      margin: 20px auto;
      width: 200px;
      border-radius: 8px;
    }
    button {
      padding: 10px 20px;
      font-size: 18px;
      cursor: pointer;
    }
  </style>
</head>
<body>

  <div id="prenents">هذا العنصر يمكن إخفاؤه</div>
  <button id="btn">إخفاء</button>

  <script>
    let prenents = document.getElementById("prenents");
    let btn = document.getElementById("btn");

    btn.addEventListener("click", function () {
      if (prenents.style.display === "none") {
        prenents.style.display = "block";
        btn.textContent = "إخفاء";
      } else {
        prenents.style.display = "none";
        btn.textContent = "إظهار";
      }
    });
  </script>

```

```

    }
  });
</script>

</body>
</html>

```

## 🎯 ما الجديد في الكود؟

- ✅ **إصلاح الخطأ:** الآن الكود يعمل بشكل صحيح عند النقر على الزر.
- ✅ **تحسين مظهر العنصر** باستخدام CSS لجعل التجربة أجمل.
- ✅ **تغيير نص الزر** بحيث يصبح "إظهار" عند إخفاء العنصر، و "إخفاء" عند ظهوره.
- 🔥 **جرب الكود الآن وأخبرني إن كنت تحتاج أي تحسينات إضافية!**



كود اول ما اضغط على زر لا يخفى شيء

إذا كنت تريد عند الضغط على الزر ألا يختفي العنصر في البداية، ولكن بعد الضغط مرة أخرى يتم الإخفاء، يمكنك التأكد من أن العنصر مرئي عند بدء التشغيل، ثم التحكم في إظهاره وإخفائه عند النقر على الزر.

## ✅ الكود المطلوب:

```

html

<!DOCTYPE html>
<html lang="ar">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>إظهار وإخفاء العنصر</title>
  <style>
    #prenents {
      padding: 20px;
      background-color: lightblue;
      text-align: center;
      font-size: 20px;
      font-weight: bold;
      margin: 20px auto;
      width: 200px;
      border-radius: 8px;
    }
    button {
      padding: 10px 20px;
      font-size: 18px;
      cursor: pointer;
    }
  </style>
</head>
<body>

  <div id="prenents">هذا العنصر يمكن إخفاؤه</div>
  <button id="btn">إخفاء</button>

  <script>
    let prenents = document.getElementById("prenents");
    let btn = document.getElementById("btn");

```

```
btn.addEventListener("click", function () {
  if (prents.style.display === "" || prents.style.display === "block") {
    prents.style.display = "none";
    btn.textContent = "إظهار";
  } else {
    prents.style.display = "block";
    btn.textContent = "إخفاء";
  }
});
</script>

</body>
</html>
```

## 🎯 كيف يعمل هذا الكود؟

1. في البداية، العنصر ظاهر لأنه لم يتم تعيين `display` في CSS.
2. عند الضغط على الزر لأول مرة، يتم إخفاء العنصر.
3. عند الضغط مرة أخرى، يتم إظهاره مجددًا.
4. يتم تغيير نص الزر بين "إظهار" و "إخفاء".

💡 بهذه الطريقة، عند تحميل الصفحة لأول مرة لا يختفي أي شيء حتى تضغط على الزر! 🚀  
 🧐 جربه وأخبرني إن كنت تحتاج إلى أي تعديلات إضافية! 😊



مفیش تغییر حصل

يبدو أنك تحتاج إلى التأكد من أن الكود يتعامل مع القيمة الافتراضية لخاصية `display` بشكل صحيح. في البداية، إذا لم يتم تعيين `display` في CSS، فإن `prents.style.display` سيكون سلسلة فارغة `""`، ولهذا السبب قد لا يعمل الكود كما تتوقع.

## ✅ إليك الكود الصحيح مع التأكد من القيم الافتراضية:

```
html

<!DOCTYPE html>
<html lang="ar">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>إظهار وإخفاء العنصر</title>
  <style>
    #prents {
      padding: 20px;
      background-color: lightblue;
      text-align: center;
      font-size: 20px;
      font-weight: bold;
      margin: 20px auto;
      width: 200px;
      border-radius: 8px;
    }
    button {
```



```
padding: 10px 20px;
font-size: 18px;
cursor: pointer;
}
</style>
</head>
<body>

<div id="prenants">هذا العنصر يمكن إخفاؤه</div>
<button id="btn">إخفاء</button>

<script>
let prenants = document.getElementById("prenants");
let btn = document.getElementById("btn");

btn.addEventListener("click", function () {
  if (prenants.style.display === "none") {
    prenants.style.display = "block";
    btn.textContent = "إخفاء";
  } else {
    prenants.style.display = "none";
    btn.textContent = "إظهار";
  }
});
</script>

</body>
</html>
```

## 🔥 ما الجديد في هذا الكود؟

✓ تأكدنا أن `display` يتم التحقق منه بالشكل الصحيح.

✓ عند النقر على الزر لأول مرة، يتم إخفاء العنصر، وعند النقر مرة أخرى، يظهر مجدداً.

✓ تم تعديل منطق `if` بحيث يعمل بسلسلة دون مشاكل.

💡 إذا لم يعمل الكود معك، تأكد من عدم وجود أي `display: none` في ملفات الـ CSS الخاصة بك تؤثر على العنصر عند التحميل.

🚀 جربه الآن وأخبرني إذا كنت تحتاج لأي تحسينات! 😊



```
4-let title = document.getElementById(`title`);
let originalText = title.textContent;
title.addEventListener(`mouseover`, function () {
  title.textContent = "hello";
  title.style.color = `red`;
});
title.addEventListener(`mouseout`, function () {
  title.textContent = originalText;
  title.style.color = `blue`;
});
```

الكود الذي كتبته صحيح ويعمل بشكل جيد! 🎯  
لكن سأضيف لك بعض التحسينات البسيطة لجعل التجربة أكثر احترافية.

html

```

<!DOCTYPE html>
<html lang="ar">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>تغيير النص عند مرور الماوس</title>
  <style>
    #title {
      font-size: 24px;
      font-weight: bold;
      color: blue;
      text-align: center;
      margin-top: 50px;
      transition: color 0.3s ease-in-out;
    }
  </style>
</head>
<body>

  <h1 id="title">مرر الماوس هنا</h1>

  <script>
    let title = document.getElementById("title");
    let originalText = title.textContent;

    title.addEventListener("mouseover", function () {
      title.textContent = "Hello!";
      title.style.color = "red";
    });

    title.addEventListener("mouseout", function () {
      title.textContent = originalText;
      title.style.color = "blue";
    });
  </script>

</body>
</html>

```

## ما التحسينات التي أضفناها؟ 🔥

- ✓ أضفنا CSS لتنسيق النص وتطبيق تأثير انتقال سلس عند تغيير اللون.
  - ✓ تأكدنا من أن النص الأصلي يعود عند خروج الماوس.
  - ✓ جعلنا النص متناسبًا في الوسط مع لون أزرق افتراضي.
- ♦ جربه الآن وأخبرني إن كنت تريد أي تحسينات إضافية! 🚀



```
5-let prent = document.getElementById(`prent`);
```

```

let hideCount = 10;
function text() {
  hideCount --;
  prent.style.visibility = `hidden`;
}

```

```
setTimeout(() => {
  prent.style.visibility = `visible`;
}, 1000);

if (hideCount <= 0) {
  prent.removeEventListener(`click`, text());
  alert(`!! انتهى الوقت`);
}
}

prent.addEventListener(`click`, text());
```

## أخطاء في الكود:

1. `removeEventListenerEvent`` خطأ → الصحيح هو `removeEventListener``.
2. `text`()` داخل `addEventListener`` → لا تضع الأقواس `()`` لأنها ستنفذ الدالة مباشرة عند تحميل الصفحة بدلاً من انتظار النقر.
3. العدد `hideCount`` لا يُعرض للمستخدم، من الأفضل إضافته داخل العنصر `prent``.
4. التأثير المرئي ليس سلساً، يمكن تحسينه باستخدام `opacity`` بدلاً من `visibility``.

## الكود المصحح والمحسن

```
html

<!DOCTYPE html>
<html lang="ar">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>اختفاء وظهور العنصر</title>
  <style>
    #prent {
      padding: 20px;
      background-color: lightblue;
      text-align: center;
      font-size: 20px;
      font-weight: bold;
      margin: 20px auto;
      width: 200px;
      border-radius: 8px;
      cursor: pointer;
      transition: opacity 0.5s ease-in-out;
    }
  </style>
</head>
<body>

  <div id="prent">(10) اضغط علي</div>

  <script>
    let prent = document.getElementById("prent");
    let hideCount = 10;

    function text() {
      if (hideCount > 0) {
        hideCount--;
      }
    }
  </script>
```

```
prent.style.opacity = "0";

setTimeout(() => {
  prent.style.opacity = "1";
  prent.textContent = `اضغط عليّ (${hideCount})`;
}, 1000);
}

if (hideCount <= 0) {
  prent.removeEventListener("click", text);
  alert("انتهى الوقت!");
}

prent.addEventListener("click", text);
</script>

</body>
</html>
```

## ما التحسينات في هذا الكود؟ 🔥

- ✓ إصلاح الأخطاء (`removeEventListenerEvent` → `removeEventListener`)
  - ✓ استخدام تأثير `opacity` بدلاً من `visibility` لانتقال سلس
  - ✓ إظهار عدد النقرات المتبقية داخل العنصر نفسه
  - ✓ إيقاف الكود عند الوصول إلى 0 وعدم السماح بالنقر بعد ذلك
- ♦ جربه الآن، وإذا أردت أي تعديلات، أخبرني! 🚀



```
6-let prent = document.getElementById(`prent`); //div
let textInput = document.getElementById(`textInput`); //input
textInput.addEventListener(`click`, function () {
  prent.textContent = textInput.value || "";
});
```

## أخطاء في الكود: ●

1. استخدام `click` مع `input`
  - الحدث `click` لا يعمل بشكل صحيح هنا، لأننا نريد التحديث عند إدخال النص وليس عند النقر فقط.
  - الأفضل استخدام `input` بدلاً من `click`.
2. عدم التحديث عند الكتابة
  - النص داخل `div` يتم تحديثه فقط عند النقر، لكنه لا يتغير أثناء الكتابة.
  - الأفضل أن يتم التحديث لحظيًا عند إدخال أي حرف.

## الكود المصحح والمحسن ✓

```

<!DOCTYPE html>
<html lang="ar">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>نقل النص من الإدخال إلى العنصر</title>
  <style>
    #prent {
      padding: 20px;
      background-color: lightblue;
      text-align: center;
      font-size: 20px;
      font-weight: bold;
      margin: 20px auto;
      width: 300px;
      border-radius: 8px;
    }
    input {
      display: block;
      margin: 20px auto;
      padding: 10px;
      font-size: 18px;
      width: 280px;
      text-align: center;
    }
  </style>
</head>
<body>

  <input type="text" id="textInput" placeholder="اكتب هنا...">
  <div id="prent">سيظهر النص هنا</div>

  <script>
    let prent = document.getElementById("prent");
    let textInput = document.getElementById("textInput");

    textInput.addEventListener("input", function () {
      prent.textContent = textInput.value.trim() || "سيظهر النص هنا";
    });
  </script>

</body>
</html>

```

## 🔥 ما التحسينات في هذا الكود؟

- ✓ استخدام `input` بدلاً من `click` لتحديث النص أثناء الكتابة
  - ✓ إضافة `trim()` لإزالة الفراغات غير الضرورية
  - ✓ إعادة النص الأصلي إذا كان الحقل فارغاً
  - ✓ تحسين المظهر العام باستخدام CSS
- ♦ جرب الكود الآن وأخبرني إن كنت تريد أي تحسينات إضافية! 🚀



```

7-let togglePassword = document.getElementById(`togglePassword`);
togglePassword.addEventListener(`input`, function () {
  let length = togglePassword.value.length;
  if (length < 6) {
    togglePassword.style.color = `red`;
  }
});

```

```

} else if (length <= 10) {
  togglePassword.style.color = `yellow`;
} else {
  togglePassword.style.color = `green`;
}
});

```

## ● أخطاء في الكود:

### 1. استخدام `input` مع الزر `togglePassword`

- يبدو أن `togglePassword` ليس حقل إدخال (input)، بل زر (button) أو عنصر آخر.
- في هذه الحالة، يجب التأكد من أننا نستخدم حقل إدخال كلمة المرور بدلاً منه.

### 2. ألوان النص لن تظهر في `input[type="password"]`

- لأن `password` يخفي النص، لن ترى اللون يتغير، لذا من الأفضل تغيير لون إطار الحقل (border) بدلاً من النص نفسه.

## ✓ الكود المصحح والمحسن

html

```

<!DOCTYPE html>
<html lang="ar">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>تغيير لون الحقل حسب طول الباسورد</title>
  <style>
    input {
      display: block;
      margin: 20px auto;
      padding: 10px;
      font-size: 18px;
      width: 280px;
      text-align: center;
      border: 2px solid gray;
      border-radius: 8px;
      outline: none;
      transition: border-color 0.3s ease-in-out;
    }
  </style>
</head>
<body>

  <input type="password" id="passwordInput" placeholder="أدخل كلمة المرور">

  <script>
    let passwordInput = document.getElementById("passwordInput");

    passwordInput.addEventListener("input", function () {
      let length = passwordInput.value.length;
      if (length < 6) {
        passwordInput.style.borderColor = "red";
      } else if (length <= 10) {
        passwordInput.style.borderColor = "yellow";
      } else {
        passwordInput.style.borderColor = "green";
      }
    });
  </script>

```

```
});
</script>

</body>
</html>
```

## 🔥 ما التحسينات التي أضفناها؟

- ✓ تغيير لون الحدود (`border-color`) بدلاً من النص، حتى يعمل مع `password`
- ✓ استخدام `input` من النوع `password` بدلاً من `togglePassword`
- ✓ تحسين تجربة المستخدم عبر `transition` لتغيير اللون بسلاسة
- ♦ جرب الكود الآن، وإذا كنت تريد أي تحسينات إضافية، أخبرني! 🚀



التحديات الثانية

## 📌 تحديات جديدة في DOM و Events لجافا سكريبت! 🚀

إليك مجموعة من التحديات لتطوير مهاراتك في **DOM & Events** باستخدام JavaScript!

## ✓ التحدي الأول: تغيير الخلفية عند الضغط على زر

📌 المطلوب:

- أنشئ زر عند النقر عليه يتغير لون خلفية الصفحة عشوائياً.

📝 الكود المطلوب:

```
html

<!DOCTYPE html>
<html lang="ar">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>تغيير الخلفية</title>
</head>
<body>

  <button id="changeBg">تغيير الخلفية</button>

  <script>
    let btn = document.getElementById("changeBg");

    btn.addEventListener("click", function () {
      let randomColor = `#${Math.floor(Math.random() * 16777215).toString(16)}`;
      document.body.style.backgroundColor = randomColor;
    });
  </script>
```

```
</body>
</html>
```

## التحدي الثاني: عداد تنازلي

### المطلوب:

- أنشئ عداد يبدأ من 10 إلى 0 ويعرض في `<p>`، وعند وصوله إلى 0 يظهر `alert()` يقول "انتهى الوقت!".

### الكود المطلوب:

```
html

<!DOCTYPE html>
<html lang="ar">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>عداد تنازلي</title>
</head>
<body>

  <p id="countdown">10</p>
  <button id="startCountdown">ابدأ العداد</button>

  <script>
    let count = 10;
    let p = document.getElementById("countdown");
    let btn = document.getElementById("startCountdown");

    btn.addEventListener("click", function () {
      let interval = setInterval(() => {
        count--;
        p.textContent = count;

        if (count <= 0) {
          clearInterval(interval);
          alert("!!انتهى الوقت");
        }
      }, 1000);
    });
  </script>

</body>
</html>
```

## التحدي الثالث: تبديل النص بين لغتين عند الضغط على زر

### المطلوب:

- عند الضغط على الزر، يتحول النص بين اللغة العربية والإنجليزية.

### الكود المطلوب:



```

<!DOCTYPE html>
<html lang="ar">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>تبدیل اللغة</title>
</head>
<body>

  <h1 id="text">مرحبا بالعالم</h1>
  <button id="toggleLang">تبدیل اللغة</button>

  <script>
    let text = document.getElementById("text");
    let btn = document.getElementById("toggleLang");
    let isArabic = true;

    btn.addEventListener("click", function () {
      if (isArabic) {
        text.textContent = "Hello, World!";
        isArabic = false;
      } else {
        text.textContent = "مرحبا بالعالم!";
        isArabic = true;
      }
    });
  </script>

</body>
</html>

```

## التحدي الرابع: عرض وإخفاء كلمة المرور

المطلوب: 

- أنشئ حقل إدخال لكلمة المرور مع زر لإظهار/إخفاء كلمة المرور.

الكود المطلوب: 

```

<!DOCTYPE html>
<html lang="ar">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>إظهار وإخفاء كلمة المرور</title>
</head>
<body>

  <input type="password" id="password" placeholder="أدخل كلمة المرور">
  <button id="togglePassword">👁️</button>

  <script>
    let passwordInput = document.getElementById("password");
    let toggleBtn = document.getElementById("togglePassword");

    toggleBtn.addEventListener("click", function () {
      if (passwordInput.type === "password") {
        passwordInput.type = "text";
      }
    });
  </script>

</body>
</html>

```

```
toggleBtn.textContent = "🔒";
} else {
  passwordInput.type = "password";
  toggleBtn.textContent = "👁️";
}
});
</script>

</body>
</html>
```

## ✓ التحدي الخامس: عداد ضغطات على زر مع إعادة التعيين

المطلوب: 📌

- عند الضغط على الزر، يزداد الرقم، وعند الضغط على إعادة تعيين، يعود إلى الصفر.

الكود المطلوب: 📝

```
html

<!DOCTYPE html>
<html lang="ar">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>عداد الضغوط</title>
</head>
<body>

  <p id="counter">0</p>
  <button id="increase">زيادة</button>
  <button id="reset">إعادة تعيين</button>

  <script>
    let count = 0;
    let counter = document.getElementById("counter");
    let increaseBtn = document.getElementById("increase");
    let resetBtn = document.getElementById("reset");

    increaseBtn.addEventListener("click", function () {
      count++;
      counter.textContent = count;
    });

    resetBtn.addEventListener("click", function () {
      count = 0;
      counter.textContent = count;
    });
  </script>

</body>
</html>
```

ملاحظات: 💡

- ♦ يمكنك تجربة كل التحديات على متصفحك مباشرةً.
- ♦ حاول تعديل التحديات وإضافة ميزات جديدة بنفسك لتحسين مهاراتك! 🚀
- ♦ أخبرني أي تحدي أعجبك أكثر، وإذا كنت تريد تحديات أصعب! 😊



بدون الحل

## 🚀 تحديات جديدة في DOM و Events لجافا سكريبت! 📌

إليك مجموعة من التحديات لتطوير مهاراتك في DOM & Events باستخدام JavaScript!

### ✅ التحدي الأول: تغيير الخلفية عند الضغط على زر

📌 المطلوب:

- أنشئ زر عند النقر عليه يتغير لون خلفية الصفحة عشوائيًا.

### ✅ التحدي الثاني: عداد تنازلي

📌 المطلوب:

- أنشئ عداد يبدأ من 10 إلى 0 ويعرض في `<p>`، وعند وصوله إلى 0 يظهر `alert()` يقول "انتهى الوقت!".

### ✅ التحدي الثالث: تبديل النص بين لغتين عند الضغط على زر

📌 المطلوب:

- عند الضغط على الزر، يتحول النص بين اللغة العربية والإنجليزية.

### ✅ التحدي الرابع: عرض وإخفاء كلمة المرور

📌 المطلوب:

- أنشئ حقل إدخال لكلمة المرور مع زر لإظهار/إخفاء كلمة المرور.

## التحدي الخامس: عداد ضغطات على زر مع إعادة التعيين

### المطلوب:

- عند الضغط على الزر، يزداد الرقم، وعند الضغط على إعادة تعيين، يعود إلى الصفر.

### ملاحظات:

- حاول حل التحديات بنفسك دون النظر إلى الحل مباشرة!
- أخبرني إذا احتجت إلى تلميحات أو مساعدة في أي تحدي. 🚀😊



```
1-let prent = document.getElementById(`prent`);
prent.addEventListener(`click`, function () {
  let randomColor = `#${Math.floor(Math.random() * 16777215).toString(16)}`;
  prent.style.backgroundColor = randomColor;
});
```

### الكود صحيح ويعمل بدون مشاكل!

عند النقر على العنصر `prent`، سيتم تغيير لون الخلفية إلى لون عشوائي.

### تحسينات مقترحة للكود: 🔥

#### 1. ضمان أن الكود يعمل حتى لو كان اللون العشوائي قصيرًا

- أحيانًا يكون اللون أقل من 6 أرقام، لذا يمكننا استخدام `padStart(6, '0')` لضمان أن اللون يكون دائمًا مكونًا من 6 أرقام.

#### 2. إضافة تأثير انقالي (`transition`) لجعل التغيير سلسًا.

### الكود بعد التحسين

```
html
<!DOCTYPE html>
<html lang="ar">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>تغيير لون الخلفية</title>
  <style>
    #prent {
      width: 200px;
      height: 200px;
      background-color: lightgray;
      display: flex;
```

```

        align-items: center;
        justify-content: center;
        text-align: center;
        font-size: 20px;
        font-weight: bold;
        border-radius: 10px;
        transition: background-color 0.5s ease-in-out;
    }
</style>
</head>
<body>

    <div id="prent">اضغط هنا</div>

    <script>
        let prent = document.getElementById("prent");

        prent.addEventListener("click", function () {
            let randomColor = `#${Math.floor(Math.random() * 16777215).toString(16).padStart(6, '0')}`;
            prent.style.backgroundColor = randomColor;
        });
    </script>

</body>
</html>

```

## 🔥 ما الفوائد بعد التحسين؟

- ✅ إصلاح مشكلة الألوان القصيرة باستخدام `padStart(6, '0')`.
  - ✅ إضافة تأثير انقالي (`transition`) يجعل التغيير أكثر سلاسة.
  - ✅ تحسين التصميم بجعل العنصر يظهر بشكل جميل.
- 🚀 جرب الكود الآن وأخبرني إذا كنت تحتاج أي تعديلات إضافية! 😊



```

2-let prenants = document.getElementById(`prenents`);
let p = document.getElementById(`p`);
let counter = 10;
prenants.addEventListener(`click`, function () {
    counter--;
    if (counter <= 0) {
        alert(`انتهت الوقت`);
    }
    p.textContent = counter;
});

```

✅ الكود يعمل لكنه يحتاج لبعض التحسينات!

## 🔴 المشاكل الموجودة:

### 1. العداد يصل إلى القيم السالبة بعد انتهاء الوقت

- عندما يصل `counter` إلى 0، لا يتوقف، بل يستمر بالنقصان إلى القيم السالبة.
- الحل: يجب إيقاف التغيير عند الصفر باستخدام `return`.

## 2. تحسين الرسالة عند انتهاء العداد

- حاليًا، يتم عرض `alert` ("انتهت الوقت")، لكنه لا يوقف العداد.
- يمكننا إخفاء الزر أو إلغاء حدث `click` بعد انتهاء الوقت.

### الكود بعد التحسين ✓

```
html

<!DOCTYPE html>
<html lang="ar">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>عداد تنازلي بالنقر</title>
</head>
<body>

  <button id="prents">اضغط لإنقاص العداد</button>
  <p id="p">10</p>

  <script>
    let prents = document.getElementById("prents");
    let p = document.getElementById("p");
    let counter = 10;

    prents.addEventListener("click", function () {
      if (counter <= 0) return; // إيقاف الحدث إذا انتهى العداد

      counter--;
      p.textContent = counter;

      if (counter <= 0) {
        alert("انتهى الوقت");
        prents.disabled = true; // تعطيل الزر بعد انتهاء الوقت
      }
    });
  </script>

</body>
</html>
```

### التحسينات التي تمت: ✨

- ✓ إيقاف العداد عند الصفر لتجنب الأرقام السالبة.
- ✓ تعطيل الزر (`disabled = true`) بعد انتهاء الوقت حتى لا يتم النقر عليه مجددًا.
- ✓ تحسين تجربة المستخدم بجعل النص واضحًا.

🚀 جرب الكود الآن وأخبرني إذا كنت بحاجة إلى تعديلات أخرى! 😊



```
2-let p = document.getElementById(`p`); //p
let textInput = document.getElementById("textInput"); //input
textInput.addEventListener(`click`, function () {
  if (p.textContent === "") {
    p.textContent = "";
  }
});
```

```

} else {
  p.textContent = "";
}
});

```

## المشكلة في الكود:

الكود الحالي لا يقوم بأي تغيير فعلي في النص الموجود داخل `

` عند النقر على ``.

- الشرط ``if (p.textContent === "")` لا يحقق أي فائدة لأنه في الحالتين يتم تعيين `p.textContent = ""`.
- لا يوجد تفاعل مفيد يحدث عند النقر.

## ✓ الحل: تحديث محتوى ` ` بناءً على إدخال المستخدم

📌 الفكرة: عند إدخال نص في ``، يتم عرضه تلقائيًا في `

` أثناء الكتابة.

## 📄 الكود الصحيح:

```

html

<!DOCTYPE html>
<html lang="ar">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>تحديث النص عند الكتابة</title>
</head>
<body>

  <input type="text" id="textInput" placeholder="اكتب شيئًا هنا">
  <p id="p"></p>

  <script>
    let p = document.getElementById("p");
    let textInput = document.getElementById("textInput");

    textInput.addEventListener("input", function () {
      p.textContent = textInput.value;
    });
  </script>

</body>
</html>

```

## 🔥 ما الذي تغير؟

- ✓ عند الكتابة داخل ``، يتم تحديث محتوى `

` تلقائيًا.
- ✓ تم استبدال ``click`` بـ ``input`` لأن الحدث ``click`` لا يناسب إدخال النصوص.
- ✓ تم تحسين تجربة المستخدم بجعل التفاعل ديناميكيًا.

🚀 جرب الكود الآن، وإذا كنت بحاجة إلى تحسينات أخرى، أخبرني بذلك! 😊



```
3-let p = document.getElementById("p");
let textInput = document.getElementById("textInput");

textInput.addEventListener(`click`, function () {
  if (p.textContent === "") {
    p.textContent = "";
  } else {
    p.textContent = "";
  }
});
```

## ● المشكلة في الكود:

الكود الحالي لا يؤدي إلى أي تغيير مرئي، لأن:

1. الشرط `if (p.textContent === "")` غير مفيد لأنه يتم تعيين نفس القيمة في الحالتين `p.textContent` و `p.textContent`.
2. الحدث `click` غير مناسب هنا، لأنه لا يغير النص بناءً على الإدخال، بل يعمل فقط عند الضغط على `<input>`.

## ✓ الحل: جعل `<p>` يظهر النص المكتوب داخل `<input>` أثناء الكتابة

### 📌 الفكرة:

- عند إدخال نص داخل `<input>`، يتم عرضه مباشرة داخل `<p>`.
- إذا كان الحقل فارغاً، يمكن جعل `<p>` يعرض رسالة افتراضية.

## 📄 الكود الصحيح:

html

```
<!DOCTYPE html>
<html lang="ar">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>تحديث النص عند الكتابة</title>
</head>
<body>

  <input type="text" id="textInput" placeholder="اكتب شيئاً هنا">
  <p id="p">النص سيظهر هنا</p>

  <script>
    let p = document.getElementById("p");
    let textInput = document.getElementById("textInput");

    textInput.addEventListener("input", function () {
      if (textInput.value.trim() === "") {
        p.textContent = "...النص سيظهر هنا";
      } else {

```



```

    }
    p.textContent = textInput.value;
  });
</script>

</body>
</html>

```

## ما الذي تغير؟ 🔥

- ✓ تم استبدال `click` بـ `input`، لأن `input` يستجيب عند الكتابة مباشرة.
  - ✓ تم تحسين التفاعل بحيث لا يختفي النص عند إدخال قيم فارغة، بل يظهر نص افتراضي.
  - ✓ تم استخدام `trim` () لمنع المسافات الفارغة فقط من التسبب في إخفاء النص.
- 🚀 جرب الكود الآن، وإذا كنت بحاجة إلى تحسينات أخرى، أخبرني بذلك! 😊



## حل التحدي الثالث

### ✓ التحدي الثالث: تبديل النص بين لغتين عند الضغط على زر

#### 📌 المطلوب:

- عند الضغط على الزر، يتم تغيير النص بين العربية والإنجليزية.
- يمكن تطبيق ذلك على `

` أو `

# ` أو أي عنصر نصي آخر.

### 📖 الحل باستخدام JavaScript فقط:

```

html

<!DOCTYPE html>
<html lang="ar">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>تبديل اللغة</title>
</head>
<body>

  <h1 id="title">مرحبا بالعالم</h1>
  <button id="toggleBtn">تغيير اللغة</button>

  <script>
    let title = document.getElementById("title");
    let toggleBtn = document.getElementById("toggleBtn");

    toggleBtn.addEventListener("click", function () {
      if (title.textContent === "مرحبا بالعالم") {
        title.textContent = "Hello World";
        toggleBtn.textContent = "Change to Arabic";
      } else {
        title.textContent = "مرحبا بالعالم";
        toggleBtn.textContent = "تغيير اللغة";
      }
    });
  </script>

```

```
});
</script>

</body>
</html>
```

## 🔥 شرح الكود:

✓ عند الضغط على الزر:

- إذا كان النص بالعربية `مرحبا بالعالم`، يتم تغييره إلى `Hello World`.
- إذا كان النص بالإنجليزية `Hello World`، يعود إلى `مرحبا بالعالم`.
- ✓ يتم أيضًا تحديث نص الزر ليشير إلى اللغة التي سيتم التبديل إليها.
- ✓ الكود بسيط وسريع ويعمل بكفاءة بدون الحاجة إلى مكتبات إضافية.

🚀 جرب الكود الآن وأخبرني إذا كنت تحتاج إلى أي تحسينات! 😊



```
4-let pass = document.getElementById(`pass`);
let btn = document.getElementById(`btn`);
btn.addEventListener(`click`, function () {
  if (pass.type === "password") {
    pass.type = "text";
  } else {
    pass.type = "password";
  }
});
```

✓ الكود صحيح ويعمل بدون مشاكل!

📌 عند الضغط على الزر، يتم إظهار أو إخفاء كلمة المرور عن طريق تغيير `type` بين `password` و `text`.

## 🔥 تحسينات مقترحة

1. تغيير نص الزر ليوضح للمستخدم ما الذي سيحدث عند النقر عليه.
2. إضافة أيقونة للوضوح مثل 🔒 / 🔓.
3. تحسين تجربة المستخدم بجعل الزر أكثر جاذبية.

✓ الكود بعد التحسين:

```
<!DOCTYPE html>
<html lang="ar">
<head>
```

html

```

<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>إظهار وإخفاء كلمة المرور</title>
<style>
  input {
    padding: 10px;
    font-size: 16px;
    width: 200px;
  }
  button {
    padding: 10px;
    font-size: 16px;
    cursor: pointer;
  }
</style>
</head>
<body>

<input type="password" id="pass" placeholder="أدخل كلمة المرور">
<button id="btn">👁 عرض</button>

<script>
  let pass = document.getElementById("pass");
  let btn = document.getElementById("btn");

  btn.addEventListener("click", function () {
    if (pass.type === "password") {
      pass.type = "text";
      btn.textContent = "🔒 إخفاء";
    } else {
      pass.type = "password";
      btn.textContent = "👁 عرض";
    }
  });
</script>

</body>
</html>

```

## التحسينات الجديدة: ✨

✓ تم تغيير نص الزر بين 👁 عرض و 🔒 إخفاء ليكون أكثر وضوحًا.

✓ إضافة أيقونات لجعل الزر أكثر جاذبية.

✓ تحسين التصميم بجعل الإدخال والزر أكثر أناقة.

🚀 جرب الكود وأخبرني إن كنت بحاجة إلى أي تعديلات! 😊



التحدي الخامس:

## ✓ التحدي الخامس: إنشاء عداد تنازلي وعند الانتهاء يظهر تنبيه

📌 المطلوب:

- عند النقر على عنصر معين، يتم إخفاؤه لمدة ثانية واحدة ثم يظهر مرة أخرى.
- العداد يبدأ من 10 مرات، وعند الوصول إلى الصفر يتم إيقاف التفاعل وإظهار رسالة تنبيه.

html

```

<!DOCTYPE html>
<html lang="ar">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>تحدي الاختفاء والظهور</title>
  <style>
    #prent {
      width: 200px;
      height: 200px;
      background-color: lightblue;
      display: flex;
      align-items: center;
      justify-content: center;
      text-align: center;
      font-size: 20px;
      font-weight: bold;
      border-radius: 10px;
      cursor: pointer;
      transition: opacity 0.5s ease-in-out;
    }
  </style>
</head>
<body>

  <div id="prent">اضغط علي</div>

  <script>
    let prent = document.getElementById("prent");
    let hideCount = 10;

    function hideAndShow() {
      if (hideCount <= 0) {
        alert("!! انتهى الوقت");
        prent.removeEventListener("click", hideAndShow);
        return;
      }

      hideCount--;
      prent.style.opacity = "0"; // إخفائه بالكامل بدلاً من إظهاره بشفافية
      setTimeout(() => {
        prent.style.opacity = "1"; // إعادة إظهاره بعد ثانية
      }, 1000);
    }

    prent.addEventListener("click", hideAndShow);
  </script>

</body>
</html>

```

## 🔥 تحسينات مضافة:

- ✓ استخدام `opacity` بدلاً من `visibility` لجعل التأثير أكثر سلاسة.
- ✓ إيقاف الحدث بعد 10 نقرات لتجنب استمرار التأثير بعد انتهاء العداد.
- ✓ تحسين التصميم والتفاعل بجعل العنصر يبدو جذاباً للمستخدم.

🚀 جرب الكود وأخبرني إذا كنت بحاجة إلى أي تعديلات! 😊

Is this conversation helpful so far?