$ node zero_to_builder.js

النسخة العربية

JS + TS

JAVASCRIPT + TYPESCRIPT

ZERO TO BUILDER

كتاب جافاسكريبت وتايب سكريبت الذي يعلّمك أن **تبني**، لا أن تحفظ.

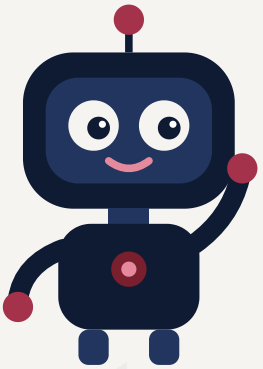
ماذا يوجد بالداخل؟

- شرح خطوة بخطوة
- 8 مشاريع حقيقية
- كويز في كل فصل
- تمارين بـ 3 مستويات
- امتحان نهائي
- مفتاح الإجابات + أكواد الحلول
- ركن الأخطاء الشائعة
- نصائح للتعلم بالذكاء الاصطناعي
- الأكواد كاملة

مناسب للمبتدئين · بدون أي خبرة سابقة

Elmasryx

أهلاً! هيا نحوّلك إلى صانع حقيقي.



بِت · مرشدك

مرحباً، أنا **بِت**. سأكون معك في كل صفحة. وعندما يبدو شيء صعباً، سأقفز وأجعله بسيطاً. اتفقنا؟

جافاسكريبت تعمل داخل **كل متصفح على هذا الكوكب**. هي التي تجعل المواقع حيّة: أزرار تتفاعل، وقوائم تنفتح، ومحادثات مباشرة، وألعاب. و**تايب سكريبت** هي نفس جافاسكريبت لكن مع حزام أمان يكتشف الأخطاء قبل أن يراها المستخدم. أتقن الاثنتين، وستردّان لك الجميل:

ابنِ مواقع
كل موقع تفاعلي على الإنترنت يستخدمها.

وظيفة
وظائف الواجهة الأمامية والخلفية والـ Full-stack تطلبها.

عمل حر
العملاء يدفعون مقابل نماذج ولوحات تحكم وتطبيقات ويب.

لغة واحدة لكل شيء
مواقع وسيرفرات وتطبيقات موبايل وكمبيوتر.

66%

من المطورين يستخدمون جافاسكريبت، اللغة رقم 1 في أكبر استبيان للمطورين في العالم. وأصبحت تايب سكريبت اللغة رقم 1 على GitHub في أغسطس 2025.

المصادر: Stack Overflow Developer Survey 2025 · GitHub Octoverse 2025

Elmasryx

جاهز خلال 10 دقائق.

خبر جميل: أنت تملك بالفعل أهم أداة. **متصفح الإنترنت** عندك يشغّل جافاسكريبت. كل ما تحتاجه دفتر مجاني تكتب فيه.

1 نزّل VS Code

نزّله مجاناً من **code.visualstudio.com** وثبّته. هناك ستكتب الكود.

2 مجلد فيه ملفان

أنشئ مجلداً اسمه `my-js`، وبداخله `index.html` و `app.js`. ضع هذا في `index.html`:

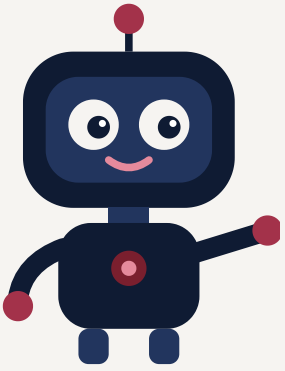
INDEX.HTML

```html
<!DOCTYPE html>
<html>
  <head><meta charset="UTF-8"></head>
  <body>
    <h1>My first JS page</h1>
    <script src="app.js"></script> 1
  </body>
</html>
```

3 افتح الـ Console

افتح `index.html` في Chrome أو Edge. اضغط **F12** (على الماك: ⌥ ⌘ **I**) ثم تبويب **Console**. هذه شاشة جافاسكريبت الخاصة بك.

1 هذا السطر يقول للصفحة: "شغّلي كود جافاسكريبت الموجود في `app.js`". وهناك ستكتب كل الكود.



نصيحة بِت

غيّرت `app.js`؟ احفظ واضغط **F5** لتحديث الصفحة. لا يوجد لابتوب؟ ابحث عن "online JavaScript playground" وشغّل الأمثلة من المتصفح.

سطر واحد.
برنامجك الأول.

افتح `app.js` واكتب هذا السطر. احفظ، ثم حدّث الصفحة.

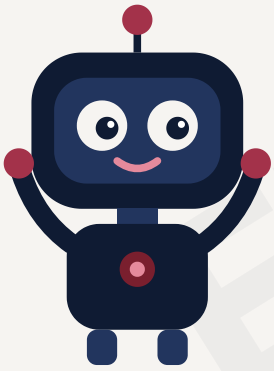
APP.JS

```
console.log(1)("Hello, world!"2);3
```

1. `console.log` معناها **"اعرض هذا في الـ Console."**
2. النص يُكتب بين **علامتي تنصيص**. هكذا تعرف جافاسكريبت أنه نص وليس أمراً.
3. **الفاصلة المنقوطة** تنهي السطر، كما تنهي النقطة الجملة.

النتيجة

```
Hello, world!
```



بِت

لقد أمرت الكمبيوتر بشيء، **ونفّذه!** تريده على الصفحة نفسها؟ جرّب `alert("Hi!");` وحدّث الصفحة. ستظهر لك نافذة تقول مرحباً.



ركن الأخطاء

نسيت علامتي التنصيص؟ ستظن جافاسكريبت أن `Hello` اسم يجب أن تعرفه:

```
Uncaught ReferenceError: Hello is not defined
```

الحل: أعد النص بين علامتي التنصيص. والرسالة الحمراء في الـ Console تخبرك برقم السطر أيضاً.

الفصل 01

01 أساسيات جافاسكريبت

كل تطبيق تستخدمه يفعل ثلاثة أشياء: **يتذكر** المعلومات، و**يحسب**، و**يتكلم** معك. وبعد هذا الفصل، كودك سيفعل الثلاثة.

let const صناديق عليها ملصقات

ماذا يفعل؟

يخزّن قيمة تحت اسم، لتستخدمها مرة أخرى لاحقاً.

لماذا تحتاجه؟

سلة المشتريات تتذكر المنتجات، واللعبة تتذكر نقاطك. هذه الذاكرة هي المتغيرات.

APP.JS

```
const① name = "Sara";
let② age = 24;
age = age + 1;③    // happy birthday!

console.log(name, "is now", age);
```

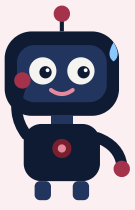
① `const` = صندوق **لا يتغير أبداً**. استخدمه دائماً كبداية.

② `let` = صندوق **يمكنك تغييره** لاحقاً.

③ خذ القيمة القديمة، وأضف 1، وضعها في نفس الصندوق.

النتيجة

```
Sara is now 25
```



ركن الأخطاء

حاولت تغيير `const`؟ جافاسكريبت تحميه:

```
TypeError: Assignment to constant variable.
```

الحل: إذا كانت القيمة ستتغير، استخدم `let`.

نصوص، أرقام، صح أو خطأ.

"Sara"	24	true	null
string	number	boolean	فارغ
نص، دائماً بين علامتي تنصيص.	صحيح أو عشري: 9.99	نعم أو لا: true / false.	"لا شيء هنا، عن قصد".

`${ }` ضع القيم داخل النص

ماذا يفعل؟
يبني جملة تخلط النص بالقيم بشكل نظيف.

لماذا تحتاجه؟
كل "أهلاً يا سارة!" و"الإجمالي: $42" على أي موقع مبنية بهذه الطريقة.

MATH.JS

```
const item = "Coffee";
const price = 3.5;
const qty = 4;

const total = price * qty;
console.log(`${item} x ${qty} = $${total}`);
console.log(10 % 3);
```

1. الحساب مثل الآلة الحاسبة: `+ - * /`.
2. علامات `` ` `` المائلة (وليس علامات التنصيص!) تسمح لك بوضع القيم داخل `${ }`.
3. `%` تعطي **باقي القسمة**: 10 ÷ 3 = 3 والباقي **1**.

النتيجة

```
Coffee x 4 = $14
1
```

اجعل صفحتك تدير محادثة.

prompt() اسأل المستخدم

ماذا يفعل؟
يُظهر مربعاً يكتب فيه المستخدم إجابته، ويعيدها لكودك.

لماذا تحتاجه؟
البرامج التي تستمع مفيدة: آلات حاسبة، اختبارات، نماذج.

APP.JS

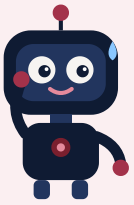
```
const name = prompt(1)("What's your name?");
const age = Number(2)(prompt("How old are you?"));

console.log(`Hi ${name}! In 10 years: ${age + 10}`);
```

1. `prompt` تعرض سؤالاً وتعيد ما كتبه المستخدم، و**دائماً كنص**.
2. `Number()` تحوّل نصاً مثل `"21"` إلى الرقم `21` ليعمل الحساب.

النتيجة

```
Hi Omar! In 10 years: 31
```

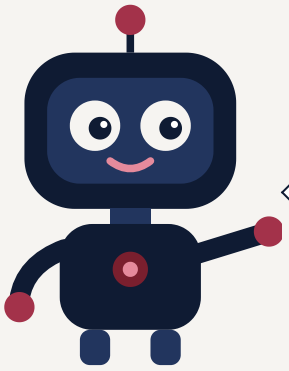


ركن الأخطاء

لو نسيت `Number()` ستلصق جافاسكريبت النصوص بدل أن تجمعها:

`"21" + 10 → "2110"`

الحل: ضع داخل `Number()` أي شيء ستحسب به.



بِت

المواقع الحقيقية تستخدم مربعات إدخال وليس `prompt`.
ستبنيها في الفصل 6. الآن `prompt` تجعل الأمور بسيطة.