

# JavaScript:

مقدمة إلى البرمجة بلغة جافا سكريبت،

تعريف المتغيرات والدوال

د. محمد الحربي

## ما هو برنامج الحاسب الآلي؟

- هو مجموعة منظمة من الأوامر والتعليمات المكتوبة باستخدام لغة برمجة معينة، تهدف إلى توجيه الحاسوب للقيام بسلسلة من العمليات أو حل مشكلة محددة بشكل آلي.

# ما هي لغة JavaScript؟

- لغة برمجة نصية (Scripting Language) تستخدم لإضافة التفاعلية لصفحات الويب.
- تعمل في المتصفح (Client-side).
- من خلال JS نستطيع:
  - التحقق من صحة النماذج.
  - التحكم في عناصر الصفحة (إخفاء/إظهار/تغيير).
  - إنشاء تطبيقات ويب تفاعلية وكاملة.

# لماذا JavaScript؟

- هي اللغة المسؤولة بشكل كبير عن التفاعلية في مواقع الإنترنت.
- تعمل مباشرة داخل المتصفح دون الحاجة لأي إضافات.
- أغلب (ربما جميع) مواقع الإنترنت تستخدم لغة JS بشكل أو بآخر.
- سهولة التعلم و التدريب.
- أشهر استخداماتها:
  - الخرائط التفاعلية
  - الوسائط المتعددة
  - التحقق من صحة النماذج
  - إلخ

# التعليقات JavaScript

- كتابة التعليقات تساعد على فهم التعليمات البرمجية بشكل أفضل.
- تمكن المبرمج من التعبير عن أفكاره بوضوح داخل التعليمات البرمجية.
- تمكن المبرمجين من التواصل و التعاون في كتابة التعليمات البرمجية بشكل أفضل.
- يمكنك من تعطيل أجزاء من التعليمات البرمجية مؤقتًا دون حذفها (لغرض التجربة أو التصحيح).

# التعليقات JavaScript

- أنواع التعليقات في JavaScript:

تعليق السطر الواحد	تعليق متعدد الأسطر
<pre>Code ..... // هذا تعليق سطر واحد Code .....</pre>	<pre>Code ..... /* هذا تعليق على عدة أسطر */ Code ..... /* هذا تعليق متعدد الأسطر في سطر واحد */ Code .....</pre>

# أين أكتب الـ JavaScript

## ● مباشرة داخل صفحة HTML

- يمكنك كتابة تعليمات JavaScript باستخدام وسم `<script>` إما:
  - في رأس الصفحة `<head>`.
  - أو نهاية `<body>` (ويفضل هنا عمليًا).

```
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>JS مقدمة</title>

  <script>
    |   // هنا JavaScript كود
  </script>
</head>
```

# أين أكتب الـ JavaScript

- الطريقة الأولى: مباشرة داخل صفحة HTML

- يمكنك كتابة تعليمات JavaScript باستخدام وسم `<script>` إما:
  - في رأس الصفحة `<head>`.
  - أو نهاية `<body>` (ويفضل هنا عملياً لأنه يضمن تحميل الصفحة أولاً قبل تنفيذ الأكواد)

`<head>`

```
<meta charset="UTF-8">
<meta name="viewport" content="width=
<title>JS مقدمة</title>
```

```
<script>
```

```
    // هنا JavaScript كود
```

```
</script>
```

`</head>`

`<body>`

```
<!-- محتوى الصفحة -->
```

```
<script>
```

```
    // هنا JavaScript كود
```

```
</script>
```

`</body>`

`</html>`



# أين أكتب الـ JavaScript

- الطريقة الثانية: استخدام ملف JavaScript خارجي
  - خيار أفضل للحفاظ على تنظيم الكود
  - نكتب JavaScript في ملف مستقل بامتداد .js، ثم نربطه بالصفحة. (مشابه لربط ملف css بالصفحة)
  - يوضع الوسم عادة قبل نهاية <body/>.

```
<body>  
    <!-- محتوى الصفحة -->  
  
    <script src="script.js"></script>  
</body>  
</html>
```

# كتابة أول كود JavaScript

- `console.log` هي دالة (Function) تستخدم لعرض رسائل نصية أو قيم المتغيرات داخل وحدة التحكم (Console) الخاصة بالمتصفح أثناء تنفيذ الكود.
- تستخدم لفحص القيم أثناء كتابة الكود (Debugging).
- مفيدة لإجراء اختبارات

```
<script>  
|   console.log("Hello, World!");  
</script>
```

# كتابة أول كود JavaScript

- alert هي دالة (Function) في JavaScript تستخدم لعرض رسالة منبثقة (نافذة صغيرة) على شاشة المستخدم.
- تستخدم لإظهار نتائج بسيطة أو لاختبار الكود أثناء البرمجة.

<> 03.html  
JS 03.js

<body>

<!-- محتوى الصفحة -->

<script src="03.js"></script>

</body>

</html>

alert("Welcome to JavaScript!");

# المتغيرات في JavaScript

userName
@price
first_name
\$amount
2name
_isValid
user-name
first name
let

- المتغير (variable) هو مكان لتخزين البيانات.
- قواعد تسمية المتغيرات:
  - تتكون من حروف، أرقام، - (underscores)، و علامة الدولار \$.
  - لا تبدأ برقم.
  - لا تحتوي على مسافات.
  - لا تستخدم أي من الكلمات المحجوزة ( كلمات النظام ) مثل var, let, if, for.
  - حساسة لحالة الأحرف (Case Sensitive).
- جميع المتغيرات لابد أن يكون لها أسماء مميزة و غير مكررة.

# المتغيرات في JavaScript

- المتغير (variable) هو مكان لتخزين البيانات.
- قواعد تسمية المتغيرات:
  - تتكون الأسماء من حروف، أرقام، - (underscores)، و علامة الدولارات \$.
  - لا تبدأ برقم.
  - لا تحتوي على مسافات.
  - لا تستخدم أي من الكلمات المحجوزة ( كلمات النظام).
  - حساسة لحالة الأحرف (Case Sensitive).
- جميع المتغيرات لابد أن يكون لها أسماء مميزة و غير مكررة.

أمثلة خاطئة	أمثلة صحيحة
2name	userName
user-name	age
first name	first_name
let	_isValid
@price	\$amount

# المتغيرات في JavaScript

● لتعريف المتغيرات فإننا نستخدم `let` أو `const` (بدلاً من `var` القديمة).

- `let` متغير يمكن تغيير قيمته في أي وقت
- `const` متغير قيمته ثابتة لا يمكن تغييرها لاحقاً

```
<script>
  let name = "Ahmed";
  const age = 25;

  console.log(name); // Ahmed
  console.log(age);  // 25
</script>
```

# المتغيرات في JavaScript

● لتعريف المتغيرات فإننا نستخدم `let` أو `const` (بدلاً من القديمة `var`).

○ `let` متغير يمكن تغيير قيمته في أي وقت

○ `const` متغير قيمته ثابتة لا يمكن تغييرها لاحقاً

```
<script>
```

```
let name = "Ahmed";
```

```
const age = 25;
```


```
console.log(name);
```

```
console.log(age);
```

```
name = "Sara";
```

```
age = 40;
```

```
</script>
```

Sara	Ahmed	name
	25	age

# المتغيرات في JavaScript

- لتعريف المتغيرات فإننا نستخدم `let` أو `const` (بدلاً من القديمة `var`).

- `let` متغير يمكن تغيير قيمته في أي وقت
- `const` متغير قيمته ثابتة لا يمكن تغييرها لاحقاً

```
<script>
```

```
let name = "Ahmed";
```

```
const age = 25;
```


```
console.log(name);
```

```
console.log(age);
```

```
name = "Sara";
```

```
age = 40;
```

```
</script>
```

Sara	Ahmed	name
	25	age





# الدوال (Functions) في JavaScript

- الدالة هي مجموعة من التعليمات البرمجية تنفذ مهمة معينة.
- نكتب الدالة مرة واحدة ونستدعيها عدة مرات عند حاجتنا إليها مما يوفر علينا التكرار ويساعد على كتابة التعليمات البرمجية بشكل منظم.

## ● لماذا نستخدم الدوال؟

- لتنظيم الكود وتقسيمه إلى أجزاء صغيرة سهلة الفهم.
- لتجنب تكرار نفس الكود عدة مرات.
- لجعل الكود أسهل للصيانة والتطوير.
- لإعادة استخدام نفس الوظيفة في أماكن مختلفة من البرنامج.

# الصيغة الأساسية لإنشاء دالة في JavaScript

- يبدأ تعريف الدالة في JS باستخدام الكلمة `function`
- يتبع ذلك، اسم الدالة و يفضل أن يكون الاسم يدل على وظيفة الدالة مثلا `printName`، `.addNumbers`
- بعد ذلك، يتم إضافة المدخلات (اختيارية) و هي القيم التي ترسل إلى الدالة.

```
function name(parameter1, parameter2, parameter3) {  
    // كود يتم تنفيذه  
}
```

# الصيغة الأساسية لإنشاء دالة في JavaScript

- يبدأ تعريف الدالة في JS باستخدام الكلمة `function`
- يتبع ذلك، اسم الدالة و يفضل أن يكون الاسم يدل على وظيفة الدالة مثلا `printName`، `.addNumbers`
- بعد ذلك، يتم إضافة المدخلات (اختيارية) و هي القيم التي ترسل إلى الدالة.

```
function greet() {  
    alert("Welcome to my website!");  
}
```

# استدعاء الدالة في JavaScript

- يتم استدعاء الدالة لتنفيذها باستخدام اسم الدالة.

```
function greet() {  
    alert("Welcome to my website!");  
}
```

```
greet(); // استدعاء الدالة
```

# المدخلات في دالة JavaScript

- في كثير من الحالات فإننا نحتاج إلى أن نرسل بيانات إلى الدالة، يتم ذلك من خلال المدخلات (أو المعاملات) التي ترسل إلى الدالة.

```
function greet(name) {  
    alert("Welcome to my website " + name + "!");  
}
```

```
greet("Waleed"); // استدعاء الدالة
```

# الإرجاع (return) في الدوال في JavaScript

- تستخدم return لإرسال نتيجة أو معلومة من الدالة إلى الكود الذي استدعاها.
- بدون return، تنفذ الدالة أوامر فقط، ولكن لا تعيد قيمة.
- عندما تصل JavaScript إلى عبارة return داخل الدالة:
  - تتوقف الدالة فوراً عن التنفيذ.
  - يتم إرجاع قيمة إلى المكان الذي استدعيت منه الدالة.

```
function addNumbers(x, y) {  
    return x + y;  
}
```

```
let sum = addNumbers(5, 7);  
console.log(sum);
```

# الإرجاع (return) في الدوال في JavaScript

- مثال تطبيقي: اكتب دالة لحساب العمر بناءً على سنة الميلاد

# تطبيق عملي للربط بين HTML و JavaScript

- كما تعلمنا سابقًا، HTML تهتم ببناء محتوى الصفحة، وJavaScript يضيف الذكاء والتفاعل لهذا المحتوى.
- لذلك ربط HTML مع JavaScript يجعل صفحات الويب تفاعلية مع المستخدم.

```
<h1>أهلاً بك في موقعي</h1>
```

```
<button onclick="sayWelcome()">اضغط للترحيب</button>
```

HTML

```
<script>
```

```
  function sayWelcome() {
```

```
    let myName = "أحمد";
```

```
    alert("!في موقعنا " + myName + " مرحبًا بك يا");
```

```
  }
```

```
</script>
```

JS



# تطبيق عملي للربط بين HTML و JavaScript

أهلاً بك في موقعي

اضغط للترحيب

HTML

127.0.0.1:5500 says

مرحباً بك يا أحمد في موقعنا!

OK

JS

شكراً لكم