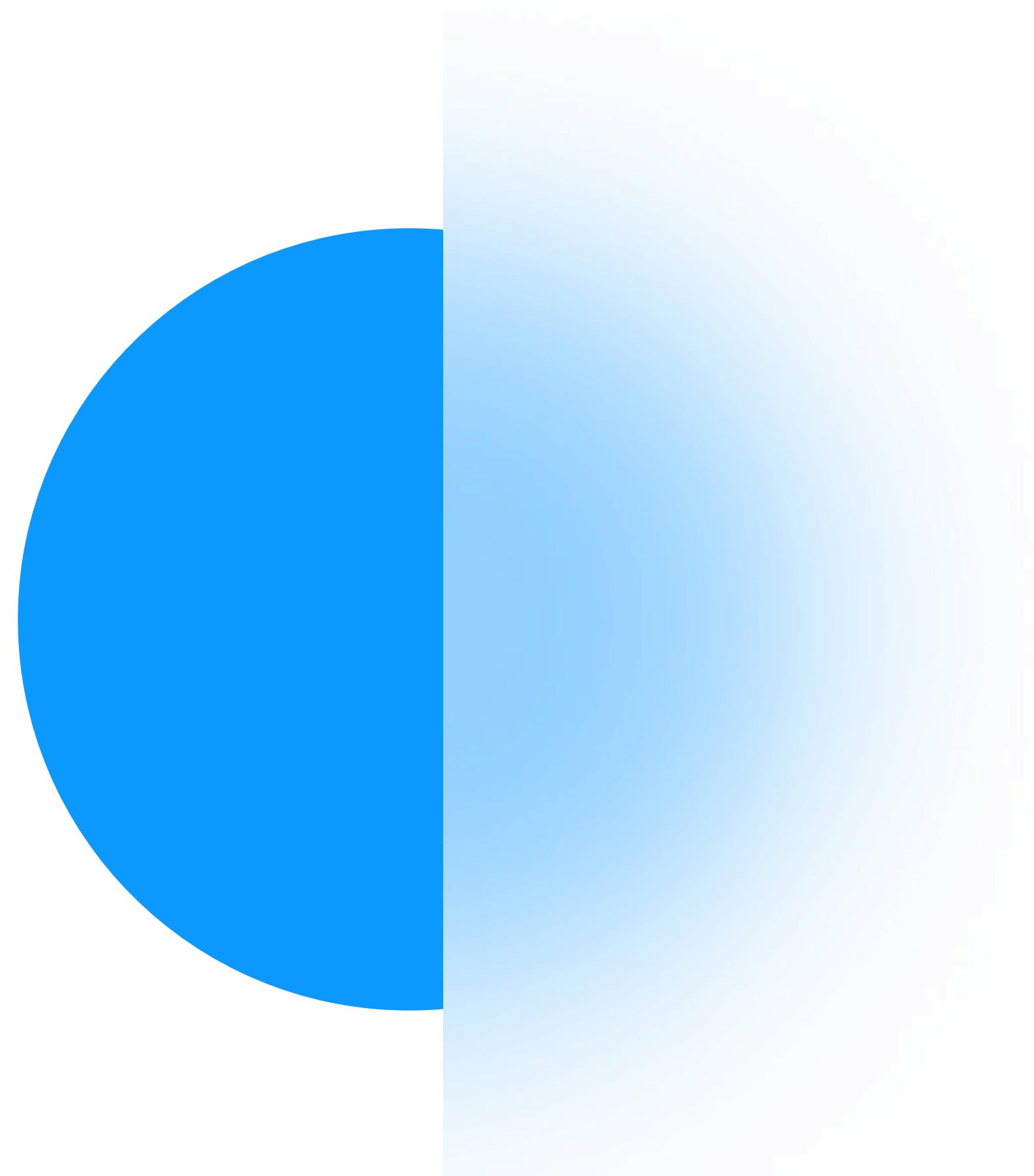


JS

(JavaScript)

Les 1 Basics & Events & First function (html+css+js)



الموضوع	الوقت (التوقيت السوري)	التاريخ
HTML: Basics & Elements & page anatomy	12:00	2025\3\1
HTML: Basics check	12:00	2025\3\8
CSS: Basics & Selectors	12:00	2025\3\15
Typography & Colors	12:00	2025\3\22
Positions & :has() :not()	12:00	2025\3\29
Flexbox	12:00	2025\4\5
Grid	19:30	2025\4\7
States & Media Querie	19:30	2025\4\9
Animations	19:30	2025\4\11
JS: Basics & Events & First function (html+css+js)	19:30	2025\4\14
JS: Conditional Statements & Arrays & Loops	19:30	2025\4\16
JS: API & Fetch data (pokeapi)	19:30	2025\4\18
Last check	19:30	2025\4\20



HTML



HTML + CSS



**HTML + CSS
+ JAVASCRIPT**

JavaScript

- ما هي JavaScript؟
لغة برمجة تُستخدم لإضافة التفاعل والдинاميكية على صفحات الويب.
- أين تُستخدم؟
تُضمن داخل صفحات HTML ويمكن ربطها بملفات CSS لتحسين التصميم والتفاعل.
- دمج HTML وCSS وJS معاً:
 - HTML: يحدد الهيكل (العناصر مثل الزر، الرسالة، العناوين).
 - CSS: يُنسق هذه العناصر (ألوان، أحجام، تنسيقات).
 - JS: يُضيف التفاعل (مثلاً، تغيير محتوى العنصر عند النقر).

1



```
1 <body>
2   <h1>Test website</h1>
3
4   <script>
5     console.log("Hello world");
6   </script>
7 </body>
```

طرق تضمين الـ JS في

الـ HTML

طرق تضمين :JavaScript

1. داخل الصفحة (Internal): كتابة الكود داخل وسم <script> في ملف .HTML

2



```
1 <body>
2   <h1>Test website</h1>
3
4   <script src="./script/script.js"></script>
5 </body>
```

2. خارجي (External): ربط ملف JS منفصل باستخدام وسم <script src="app.js"></script>

```
<head>
  <script src="script.js"></script>
</head>
```

```
<head>
  <script src="script.js" defer></script>
</head>
```

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

طرق تضمين الـ **JS** في

HTML

طرق تضمين **JavaScript**

1. وضع كود الرابط داخل الـ **head**

2. وضع كود الرابط داخل الـ **body**

العيوب	الميزات	وقت التنفيذ	الموقع
<ul style="list-style-type: none"> ✖ قد لا تكون عناصر DOM جاهزة ✖ احتمال حدوث أخطاء ✖ يبطئ تحميل الصفحة 	<ul style="list-style-type: none"> - يبدأ التنفيذ مبكراً 	<ul style="list-style-type: none"> قبل تحميل الصفحة 	<ul style="list-style-type: none"> <head>
<ul style="list-style-type: none"> - السكريبت ينفذ بعد كل العناصر (ليس مبكراً) 	<ul style="list-style-type: none"> - السكريبت جاهز DOM ✓ ✓ يقلل من وقت الانتظار ✓ لا يؤثر على تحميل الصفحة 	<ul style="list-style-type: none"> بعد تحميل الصفحة 	<ul style="list-style-type: none"> <body>
<ul style="list-style-type: none"> - يجب أن يكون السكريبت لا يعتمد على تنفيذ فوري قبل تحميل الصفحة 	<ul style="list-style-type: none"> ✓ يتم تحميل السكريبت بشكل موازي ✓ DOM جاهز وقت التنفيذ ✓ ترتيب أنظف 	<ul style="list-style-type: none"> بعد تحميل الصفحة 	<ul style="list-style-type: none"> <head> مع تحميل الصفحة

أنواع المتغيرات حسب القيمة

Data Types	Description	Example
String	Represents textual data	'hello', "hello world!" etc
Number	An integer or a floating-point number	3, 3.234, 3e-2 etc
Boolean	Any of two values: true or false	true and false
Undefined	A data type whose variable is not initialised	let a;
Null	Represents textual data	denotes a null value
Object	Key-value pairs of collection of data	let student = {};

تعريفات اساسيات في ال js

المتغيرات (variables)

1. يجب ان يحصل كل عنصر في الجافاسكريبت على متغير خاص به حتى ولو تم استدعاء العنصر من ملف ال .html

2. لتعريف variable ثابت القيمة (value) نستخدم const

3. لتعريف variable متغير القيمة (value) نستخدم let

4. عند إنشاء متغير خارج الدوال (Functions) عندما يكون صالح للاستخدام في جميع أنحاء الملف

5. عند إنشاء متغير داخل دوالة (Function) عندما يكون صالح للاستخدام فقط داخل هذه الدالة ولا يمكن استخدامه في أي مكان خارج هذه الدالة إلا

في حالة استخدام مايسمنى Callback function

ملاحظة مهمة: يمكن تعريف variable متغير القيمة (value) باستخدام var لكنها لم تعد مستخدمة حاليا ويتم الاعتماد على let

تعريفات اساسيات في الـ js

:Assignment Operators

ملاحظة مهمة:

- في حالة الجمع:

عند جمع رقمين فإننا نحصل على نتيجة جمعهم

لكن عند جمع نصين كمثال: "A" + "B" = "AB" | "5" + "2" = "52"

كذلك في حال جمع نص مع رقم: "A" + 5 = "A5" | "5" + 2 = "52"

- علامات الطرح -، الضرب *، القسمة /، باقي القسمة %، الأُس **

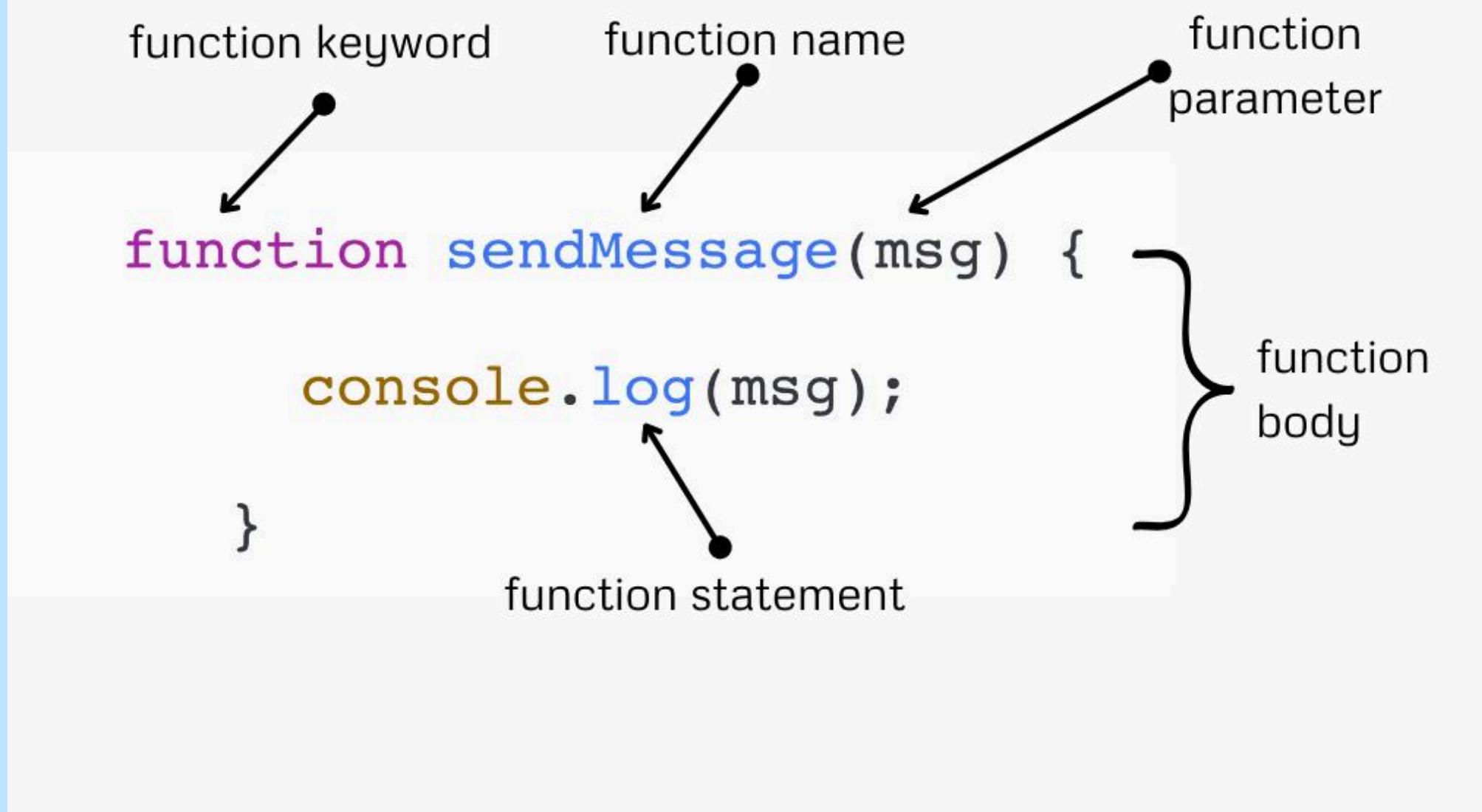
هذه العمليات تحاول تحويل النصوص إلى أرقام أولاً، ثم تنفذ العملية.

إذا لم يكن بالإمكان تحويل النص إلى رقم، النتيجة تكون

.NaN (Not a Number)

الرمز	الاسم	ما يعادلها	مثال	الناتج بعد التنفيذ
=	إسناد	x = y	;let x = 5	x = 5
+=	إسناد مع جمع	x = x + y	;let x = 5; x += 3	x = 8
-=	إسناد مع طرح	x = x - y	;let x = 5; x -= 2	x = 3
*=	إسناد مع ضرب	x = x * y	;let x = 4; x *= 3	x = 12
/=	إسناد مع قسمة	x = x / y	;let x = 10; x /= 2	x = 5
%=	إسناد مع باقي القسمة	x = x % y	;let x = 10; x %= 4	x = 2
**=	إسناد مع الأُس	x = x ** y	;let x = 2; x **= 3	x = 8

Function Declaration



```
● ● ●  
1 function printData(name) {  
2   console.log("Hello", name);  
3 }  
4  
5 printData("Jon");  
6  
7 // the result in the console will be: Hello Jon
```

تعريفات اساسيات في الـ js

الدوال (Functions)

الدالة هي قطعة من الكود تُنفذ عند استدعائها.

```
● ● ●  
1 function printData() {  
2   console.log("Hello world");  
3 }  
4  
5 printData();  
6  
7 // the result in the console will be: Hello world  
8
```

تعريفات اساسيات في الـ js

الدوال (Callback Functions)

وهي دوال يتم إستدعائها داخل دوال اخرى للقيام بوظيفة ما لكمال نوع من الاجراء.

```

1 function PrintSumResult() {
2   let resultText = "The sum is:";
3   let a = 5;
4   let b = 10;
5   let sumResult = sum(a, b); // Call the sum function with a and b as arguments
6   // The sum function takes two parameters and returns their sum
7   console.log(resultText + " " + sumResult); // Output: The sum is: 15
8 }
9
10 function sum(a, b) {
11   return a + b;
12 }
13
14 PrintSumResult(); // Call the PrintSumResult function to execute the code
15

```

```

1 function PrintSumResult() {
2   let resultText = "The sum is:";
3   let a = "A"; ←
4   let b = "M"; ←
5   let sumResult = sum(a, b); // Call the sum function with a and b as arguments
6   // The sum function takes two parameters and returns their sum
7   console.log(resultText + " " + sumResult); // the output is "The sum is: AM"
8 }
9
10 function sum(a, b) {
11   return a + b;
12 }
13
14 PrintSumResult(); // Call the PrintSumResult function to execute the code
15

```

إن () هو sum() يقوم بإرجاع قيمة وهي callback function وهي ناتج جمع ايا قيمتين يتم تمريرهم إليه دائما.

```

1 function printData() {
2   let data = getdata();
3   // getdata() is called and the returned value is stored in the variable data
4   console.log("Hello", data);
5 }
6
7 function getdata() {
8   return "Bryan";
9   //The return statement stops the execution of a function and returns a value.
10 }
11
12 printData();
13 // the result in the console will be: Hello Bryan

```

إن () هو getdata() يقوم بإرجاع قيمة وهي "Bryan" دائمًا.

تعريفات اساسيات في الـ js

الدوال السهمية (Arrow Functions)

الوظائف السهمية هي اختصار لكتابة الوظائف في JavaScript.

```

1 let printData = function (name) {
2   console.log("Hallo", name + "!");
3 }
4
5 printData("Jack");
6
7 // resulte: Hallo Jack!

```

```

1 let printData = (name) => {
2   console.log("Hallo", name + "!");
3 }
4
5 printData("Jack");
6
7 // resulte: Hallo Jack!

```

اختصارات

```

1 function printData(name) {
2   console.log("Hello", name);
3 }
4
5 printData("Jon");
6
7 // the result in the console will be: Hello Jon

```

ملاحظة مهمة: خلال مرحلة التعلم من الأفضل استخدام الطريقة العاديّة لبناء الدوال وليس الطرق المختصرة لبياناً حفظ وفهم و التعود على كيفية التعامل مع بيئه الجافاسكريبت

مجموعة من الأحداث الشائعة:

Event Attribute	Description
<code>onclick</code>	Triggered when an element is clicked.
<code>onmouseover</code>	Fired when the mouse pointer moves over an element.
<code>onmouseout</code>	Occurs when the mouse pointer leaves an element.
<code>onkeydown</code>	Fired when a key is pressed down.
<code>onkeyup</code>	Fired when a key is released.
<code>onchange</code>	Triggered when the value of an input element changes.
<code>onload</code>	Occurs when a page has finished loading.
<code>onsubmit</code>	Fired when a form is submitted.
<code>onfocus</code>	Occurs when an element gets focus.
<code>onblur</code>	Fired when an element loses focus.

تعريفات أساسيات في الـ js

الأحداث (Events)

- حدث هو تفاعل المستخدم مع الصفحة (مثل النقر، تمرير الفأرة، إلخ).
- باستخدام `click` زر نربط حدث `addEventListener` (عنصر محدد) (الزر) ونقوم بتحديد الدالة التي تُنفذ عند حدوث الحدث.

يدعم JavaScript مجموعة متنوعة من أنواع الأحداث. تتضمن الفئات الشائعة ما يلي:

- Mouse Events:** click, dblclick, mousemove, mouseover, mouseout
- Keyboard Events:** keydown, keypress, keyup
- Form Events:** submit, change, focus, blur
- Window Events:** load, resize, scroll

html

```

1 <body>
2   <h1>Hallo user!</h1>
3   <button>Click me</button>
4
5   <script src="./script/script.js"></script>
6 </body>

```

تعريفات اساسيات في ال js

:First function

js

```

1 const Title = document.querySelector("h1");
2 const button = document.querySelector("button");
3
4 button.addEventListener("click", printInfo);
5
6 function printInfo() {
7   Title.textContent = "Hello World!";
8 }

```

Result

Hallo user!

Click me

1. يتم إنشاء ملف الجافاسكريبت وربطه بملف html

2. نكتب داخل ملف الجافاسكريبت console.log("Hello world") وتحقق من رؤية Hello world في كونسول المتصفح

[f12 > console | mouse right click > inspect > console]

وبهذا تكون الملفات ارتبطت بشكل صحيح.

3. يتم تعريف متغير لكل عنصر مرتبط للحدث:

a. العناصر التي تنفذ التعديل

b. العناصر التي يتم التعديل عليها

4. يتم ربط العنصر الذي ينفذ التعديل بالحدث (event) المناسب و الدالة

function المناسب لها.

5. يتم بناء الدالة function

The end of JS les 1