**Grade 8**

Lesson 1: Introduction to JavaScript

**Lesson Objectives (What you will learn):**

* Understand what JavaScript is
* Learn why we use it
* See how it’s used in real websites

**What is JavaScript?**

* JavaScript is a language used to make web pages *interactive* (fun and active).
* It helps websites do things when you click or type.
* It works together with:
  + **HTML** – gives structure to the web page (like headings, paragraphs)
  + **CSS** – makes the web page look nice (colors, fonts)
  + **JavaScript** – adds actions (like pop-ups, buttons that do something)

**Real Examples of JavaScript**

* Games like the **Chrome Dino game**
* Login forms or password strength checkers
* Maps that move and zoom, like **Google Maps**
* Websites like **YouTube** and **Google** – buttons, videos, and search use JavaScript

**Try This Example**

<!DOCTYPE html>

<html>

<body>

<h2>JavaScript in Action</h2>

<button onclick="alert('Hello, JavaScript!')">Click Me</button>

</body>

</html>

Click the button and see what happens!  
Try changing the message inside the quotes.

**Exercises**

**1. Change the Message**

Change this part of the code to say something new:

alert("Your message here!");

**2. Say Your Name**

Write a script that shows your name in an alert box:

<script>

alert("Hello, my name is [Your Name]!");

</script>

**3. Ask for a Name**

This script asks the user for their name and says hello:

<!DOCTYPE html>

<html>

<head>

<title>JavaScript Prompt</title>

</head>

<body>

<button onclick="askName()">Enter Your Name</button>

<script>

function askName() {

let name = prompt("What is your name?");

alert("Hello, " + name + "! Welcome to JavaScript.");

}

</script>

</body>

</html>

**Gamification:**

**Challenge: Make your own welcome message page!**

**Stars earned: 3 stars for creative messages!**

**Summary**

* JavaScript makes websites **interactive**.
* You can use it to build games, check forms, or make buttons do things.
* You learned how to show messages and ask users for input using alerts and prompts.

**Quiz questions of this lesson**

1. **What is JavaScript used for?**

a) Designing web pages

b) Making web pages interactive

c) Storing data

d) Writing server-side code only

1. **Which of the following is a real-world example of JavaScript?**

a) Google Docs formatting text

b) Google Maps displaying dynamic locations

c) Microsoft Word saving a document

d) An Excel spreadsheet formula

1. **Which tag is used to add JavaScript inside an HTML file?**

a) <java>

b) <script>

c) <js>

d) <code>

1. **What will this code do?**

<script>  
 alert("Welcome to JavaScript!");  
</script>

a) Print text on the web page

b) Show an alert box with a message

c) Change the background color

d) Do nothing

1. **How do you display a pop-up asking for the user's name?**

a) window.ask("Enter your name:")

b) alert("Enter your name:")

c) prompt("Enter your name:")

d) input("Enter your name:")

**Answer Key:**

1 - b, 2 - b, 3 - b, 4 - b, 5 – c

**Lesson 2: Comments and Console Log**

**Learning Objectives:**

* Learn how to write comments in code.
* Use console.log() to print messages for debugging**.**

**Learning Outcomes:**

* Students will explain the purpose of comments.
* Students will use console.log() in their code.

**Quick Recap (Previous Lesson)**

* JavaScript makes websites **interactive**.
* It works with:
  + **HTML** (structure),
  + **CSS** (style),
  + **JavaScript** (actions).
* You saw how JavaScript is used in real sites like YouTube and Google.

**You Practiced:**

* Showing a message using alert()
* Asking for a name with prompt()

**What are Comments?**

Comments help explain what your code does. They are not run by the browser.

// This is a single-line comment

/\* This is a

multi-line comment \*/

**What is console.log()?**

Used to print messages in the browser's console (Developer Tools).

console.log("This message shows in the console");

**Try This:**

1. Write a comment above an alert()

// This shows a welcome message

alert("Hello there!");

1. Use console.log() instead of alert() to show a message

console.log("This is a console message");

**Real-World Examples:**

* Programmers use comments to document their code.
* Developers use console.log() to test and debug code in the browser.

**Exercises:**

* Add comments to your previous scripts
* Log your name using console.log()

**Gamification:**

* **Challenge: Add comments to a code and explain each step.**
* **Stars: 3 stars for clear explanations.**

**Summary:**

* Comments explain code
* console.log() is used for debugging

**Quiz**

**Question 1:**  
What is the purpose of a comment in JavaScript code?  
A. To style the webpage  
B. To show a message to users  
C. To explain what the code does (but not run it)  
D. To run faster code

**Question 2:**  
Which of the following is a correct single-line comment in JavaScript?  
A. <!-- This is a comment -->  
B. // This is a comment  
C. # This is a comment  
D. /\* This is a comment \*/

**Question 3:**  
What does the following code do?

console.log("Testing");

A. Shows a popup message  
B. Displays "Testing" in an alert box  
C. Prints "Testing" in the browser's console  
D. Hides the message from the user

**Question 4:**  
Where can you see messages from console.log()?  
A. On the web page  
B. In the browser console (Developer Tools)  
C. In an email  
D. In the address bar

**Question 5:**  
Which of the following is a correct multi-line comment?  
A. // This is a comment  
B. /\* This is a comment \*/  
C. # This is a comment  
D. <!-- This is a comment -->

**Answers**

1. C
2. B
3. C
4. B
5. B

**Lesson 3: Variables**

**Lesson Goals (What you will learn)**

* Learn how to create and use variables in JavaScript.
* Understand how to use let and const to store data.

**Quick Recap (Previous Lesson)**

**What You Learned:**

* **Comments** help explain what your code does, but don’t run.
* console.log() prints messages in the **console** (for testing and debugging).

**Comment Types:**

javascript

Copy code

// This is a single-line comment

/\* This is a

multi-line comment \*/

**You Practiced:**

* Adding comments to your scripts
* Logging your name using console.log()

**1. What is a Variable?**

* A **variable** is like a container that holds data.
* You can store different types of data in variables, like numbers, text, or even more complex data.
* The value stored in a variable can be changed if you use let, or it can stay constant with const.

Example:

javascript

Copy code

let name = "John"; // The variable 'name' stores the value "John"

const age = 13; // The variable 'age' stores the value 13, and it cannot be changed

**2. Real-Life Examples of Variables**

* Storing a user’s name after they enter it in a form.
* Keeping track of points in a game.
* Tracking the weather temperature throughout the day.

**3. Examples**

Example 1 – Store Your Favorite Color

javascript

Copy code

let color = "blue"; // Store the color blue in the variable 'color'

console.log("My favorite color is " + color); // Print the message with the color

Example 2 – Storing Name and Age

javascript

Copy code

let name = "Alice"; // Store the name 'Alice' in the variable 'name'

const age = 25; // Store the constant value 25 in the variable 'age'

console.log(name + " is " + age + " years old."); // Print the message

**4. Activities**

Activity 1 – Try It Yourself  
Create two variables: one for your name and one for your age. Then use console.log() to display them in a message.

Activity 2 – Changing a Variable  
Try changing the value of a let variable. What happens if you try to change a const variable?

**5. Exercises**

Exercise 1 – Create Variables for Your Name and Age  
Write a program where you create variables to store your name and age. Then use console.log() to print a message that includes both values.

Example:

javascript

Copy code

let myName = "Your Name";

const myAge = 18;

console.log(myName + " is " + myAge + " years old.");

Exercise 2 – Display Your Favorite Hobby  
Create a variable that stores your favorite hobby. Then print a message that says: "My favorite hobby is [hobby]."

Example:

javascript

Copy code

let hobby = "reading";

console.log("My favorite hobby is " + hobby);

**Gamification for this Lesson**

**Challenge of the Day:**  
Create 3 variables that describe yourself (e.g., name, age, hobby).

* Earn 1 star for each variable!

**Lesson Summary**

* **Variables** are used to store data in your code.
* Use let for variables whose values can change, and use const for values that should remain constant.
* You can use console.log() to print the values of your variables and see them in action.

**Quiz:**

**1. What is a variable in JavaScript?**  
A. A command to print text  
B. A container used to store data  
C. A special comment in your code  
D. A type of loop

**2. Which keyword allows you to store a value that can change later?**  
A. const  
B. variable  
C. let  
D. change

**3. What happens if you try to change the value of a variable declared with const?**  
A. It updates normally  
B. It deletes the variable  
C. It gives an error  
D. It turns into a let variable

**4. What does this code print?**

let hobby = "painting";

console.log("My favorite hobby is " + hobby);

A. My favorite hobby is painting  
B. hobby  
C. My favorite hobby is + hobby  
D. painting

**5. Which of the following is the correct way to declare a variable for a name using let?**  
A. name = let "Sarah";  
B. let "Sarah" = name;  
C. let name = "Sarah";  
D. var name = Sarah;

**Answers**

**1. B**

**2. C**

**3. C**

**4. A**

**5. C**

**Grade 9**

Lesson 1: JavaScript Functions  
 **Lesson 1: JavaScript Functions**

**Lesson Goals (What you will learn)**

* Understand what a function is in JavaScript.
* Learn why we use functions.
* See how functions are used in real life.
* Write simple JavaScript functions.

**1️. What is a JavaScript Function?**

* A **function** is a group of instructions (code) that do something specific.
* Think of a function like a **recipe** – you write it once, and you can use it again and again.
* Example: A function that says "Hello", or one that adds two numbers.

**2️. Real Life Examples of Functions**

* **Checking Forms**: Make sure an email is typed in before submitting a form.
* **Welcome Message**: Show a message when someone visits your site.
* **Calculator**: Add or multiply numbers.
* **Buttons**: Do something when a button is clicked.

**3️. Examples**

**Example 1 – Say Hello**

<!DOCTYPE html>

<html>

<body>

<h2>Click the button to see a message:</h2>

<button onclick="sayHello()">Click Me</button>

<script>

function sayHello() {

alert("Hello, welcome to JavaScript!");

}

</script>

</body>

</html>

**Activity:**  
Change the message to say: **"Good morning, Grade 9 learners"**

**Example 2 – Show a Custom Message**

<!DOCTYPE html>

<html>

<body>

<h2>Show a Custom Message</h2>

<p>Click the button to show a message:</p>

<button onclick="showMessage('Welcome to JavaScript!')">Click Me</button>

<script>

function showMessage(message) {

alert(message);

}

</script>

</body>

</html>

**Example 3 – Multiply Two Numbers**

<!DOCTYPE html>

<html>

<body>

<h2>Multiply Two Numbers</h2>

<p>Click the button to multiply:</p>

<button onclick="multiplyNumbers(5, 6)">Click Me</button>

<script>

function multiplyNumbers(a, b) {

let result = a \* b;

alert("The result is: " + result);

}

</script>

</body>

</html>

**4. Exercises**

**Exercise 1 – Add Two Numbers**

**Create a function that adds 5 + 10 and shows the result.**

**Answer:**

<!DOCTYPE html>

<html>

<body>

<h2>Adding Two Numbers</h2>

<p>Click the button to add:</p>

<button onclick="addTwoNumbers()">Click Me</button>

<script>

function addTwoNumbers() {

let num1 = 5;

let num2 = 10;

let sum = num1 + num2;

alert(sum);

}

</script>

</body>

</html>

**Exercise 2 – Greet Someone by Name**

**Make a function that says "Hello, [name]!"**

**Answer:**

<!DOCTYPE html>

<html>

<body>

<h2>Personalized Greeting</h2>

<p>Click the button to greet:</p>

<button onclick="greet('John')">Click Me</button>

<script>

function greet(name) {

alert("Hello, " + name + "!");

}

</script>

</body>

</html>

**Gamification for this lesson:**

* **Challenge of the day: Create a function that tells a joke!**
* **Stars earned: 4 starts if it makes someone laugh**

**Lesson Summary**

**What is a Function?**

* A function is a reusable group of code that does something.
* Like a recipe – you write it once, and use it many times.

**Where Are Functions Used?**

* To check forms.
* To show welcome messages.
* To do math like adding or multiplying.
* To run actions when a button is clicked.

**Quiz: JavaScript Functions**

**1. What is the primary purpose of a function in JavaScript?**

a) To make the website look good

b) To store large amounts of data

c) To perform a specific task and be reused

d) To style the webpage

**2. Which of the following is a real-world example of using a JavaScript function?**

a) Adding a border to an image using CSS

b) Checking if a user entered a valid email in a form

c) Saving data on the server

d) Changing the font color of text on the webpage

**3. How do you define a function in JavaScript?**

a) function myFunction;

b) function myFunction() {}

c) function: myFunction();

d) myFunction = function() {}

**4. What will this JavaScript code do?**

<button onclick="showMessage('Welcome to JavaScript!')">Click Me</button>  
<script>  
 function showMessage(message) {  
 alert(message);  
 }  
</script>

a) Display a welcome message when the button is clicked

b) Change the color of the button when clicked

c) Multiply two numbers when the button is clicked

d) Nothing will happen

**5. Which of the following is an example of using a function with parameters?**

a) function greet() { alert("Hello!"); }

b) function greet(name) { alert("Hello, " + name + "!"); }

c) function greet(name, age) { alert("Hello, " + name + " who is " + age + " years old!"); }

d) Both b and c

**6. What does the following JavaScript function do?**

function multiplyNumbers(a, b) {  
 let result = a \* b;  
 alert(result);  
}

a) Adds two numbers

b) Divides two numbers

c) Multiplies two numbers

d) Subtracts two numbers

**7. How can you modify the function sayHello() to say "Good morning, Grade 9 learners"?**

a) alert("Good morning, Grade 9 learners!");

b) function sayHello() { alert("Good morning, Grade 9 learners!"); }

c) sayHello("Good morning, Grade 9 learners!");

d) Both a and b

**8. What type of value does a function with parameters return?**

a) Always a number

b) Always a string

c) It depends on the function’s task

d) It always returns nothing

**9. Which statement about functions is TRUE?**

a) Functions can only be used once

b) Functions are used to repeat tasks multiple times

c) Functions are only used in backend programming

d) Functions cannot have parameters

**10. What is the output of the following code?**

<button onclick="addTwoNumbers()">Click Me</button>  
<script>  
 function addTwoNumbers() {  
 let num1 = 5;  
 let num2 = 10;  
 let sum = num1 + num2;  
 alert(sum);  
 }  
</script>

a) 15

b) 5

c) 10

d) Nothing

**Answer Key:**

1 – c, 2 – b, 3 – b, 4 – a, 5 – d, 6 – c, 7 – d, 8 – c, 9 – b, 10 – a

**Lesson 2: Return Values in Functions**

**Lesson Goals (What you will learn)**

* Understand how to return values from a function.
* Know the difference between alert() and return.
* Learn how to use returned values in your code.

**Quick Recap (Previous Lesson)**

In the last lesson, you learned:

* What a function is (a reusable block of code).
* How to write and call a function.
* How to use alert() to show messages.
* How to pass values (like names or numbers) into a function.

**1️. What is a Return Value?**

* Sometimes a function doesn’t just *do* something—it *gives something back*.
* return is used to **send a result back** to the place where the function was called.
* It’s like ordering food from a kitchen—the kitchen (function) returns your meal (value) when it’s ready!

**Important:** alert() shows the result on screen, but return gives back the result so you can use it again later.

**2️. Real-Life Examples of Return**

* Math: Return the result of a calculation.
* Name Tags: Return someone’s full name.
* Apps: Return user input for processing.

**3️. Examples**

**Example 1 – Simple Return**

javascript

Copy code

function sayHi() {

return "Hi there!";

}

console.log(sayHi()); // Shows: Hi there!

**Example 2 – Add Two Numbers**

javascript

Copy code

function add(a, b) {

return a + b;

}

console.log(add(5, 3)); // Shows: 8

**4. Exercises**

**Exercise 1 – Try It Yourself**

Change the sayHi() function to return:  
"Good afternoon, learners!"

**Activity 2 – Multiply and Return**

Write a function that returns the result of multiplying 4 and 7.  
Use console.log() to show the result.

**5. Exercises**

**Exercise 1 – Square a Number**

Write a function that returns the square of a number (number × number).  
Try with console.log(square(5)); → It should show **25**.

**Exercise 2 – Return Full Name**

Write a function that returns your full name as a string.  
Example:

javascript

Copy code

function getFullName() {

return "Sam Junior Kayombo";

}

console.log(getFullName());

**Gamification for this Lesson**

**Challenge of the Day:**

Create a function called getGreeting(name) that returns a greeting like  
"Hello, Angela!" when you give it a name.

* Earn 3 stars if your function works for any name.

**Lesson Summary**

* Functions can return values using the return keyword.
* Returned values can be stored, printed, or used in other functions.
* Use console.log() to show what a function returns.

Sure! Here's a **quiz** based on *Lesson 2: Return Values in Functions* with:

* 10 multiple-choice questions
* 2 fill-in-the-blank questions
* 1 write-a-code task

**Quiz**

**1. What does the return keyword do in a function?**  
A. Shows an alert box  
B. Repeats the function  
C. Sends a value back to where the function was called  
D. Stops the program

**2. Which of these will return the value 10?**  
A. function getTen() { alert(10); }  
B. function getTen() { return 10; }  
C. function getTen() { console.log(10); }  
D. function getTen() { return "10"; }

**3. What is the difference between return and alert()?**  
A. return is slower than alert()  
B. return is only used in games  
C. alert() returns values while return only displays them  
D. return gives back a value, alert() just shows it on screen

**4. What will this code output?**

function greet() {

return "Hello!";

}

console.log(greet());

A. Nothing  
B. Hello!  
C. greet  
D. undefined

**5. What does this function return?**

function add(a, b) {

return a + b;

}

A. A string  
B. An alert  
C. The sum of a and b  
D. Nothing

**6. Why is it useful to return a value from a function?**  
A. So we can reuse the result  
B. To stop the code  
C. To show a popup  
D. To store the function name

**7. Which of the following calls will output 12?**

function multiply(a, b) {

return a \* b;

}

A. console.log(multiply(4, 3));  
B. console.log(multiply(2, 2));  
C. console.log(multiply(3, 2));  
D. console.log(multiply(5, 1));

**Fill in the Blanks**

**8. The \_\_\_\_\_\_\_ keyword is used in a function to send a value back.**  
**9. alert() shows something to the user, but \_\_\_\_\_\_\_ gives a value that can be reused.**

**Write a Code**

**10. Write a function called getGreeting(name) that takes a name and returns a greeting like "Hello, Angela!"**

**Answers**

**Multiple Choice**

1. **C**
2. **B**
3. **D**
4. **B**
5. **C**
6. **A**
7. **A**

**Fill in the Blanks**

1. return
2. return

**Code Answer**

function getGreeting(name) {

return "Hello, " + name + "!";

}

console.log(getGreeting("Angela")); // Hello, Angela!

**Lesson 3: Conditional Statements (if/else)**

**Lesson Goals (What you will learn)**

* Understand what conditional statements are.
* Learn how to use if, else if, and else to make decisions.
* Write simple programs that respond differently based on conditions.

**Quick Recap (Previous Lesson)**

In the last lesson, you learned:

* How to return values from a function using return.
* That functions can give results back, like math answers or names.
* How to store and display returned values using console.log().

**1. What are Conditional Statements?**

* Conditional statements help us make decisions in code.
* We use them to run certain parts of code only when something is true.
* The basic structure uses if, and we can also use else if and else to add more options.

Example:

let age = 15;

if (age >= 18) {

console.log("You are an adult.");

} else {

console.log("You are a minor.");

}

This code checks if someone is 18 or older. If they are, it says they are an adult. If not, it says they are a minor.

**2. Real-Life Examples of Conditional Statements**

* Login checks: If the password is correct, log in. Otherwise, show an error.
* Age restrictions: If someone is old enough, allow access. If not, block access.
* Games: If your score is high enough, move to the next level.

**3. Examples**

**Example 1** – Check Age

let age = 20;

if (age >= 18) {

console.log("You are an adult.");

} else {

console.log("You are a minor.");

}

**Example 2** – Grade Feedback

let score = 85;

if (score >= 90) {

console.log("Excellent!");

} else if (score >= 70) {

console.log("Good job!");

} else {

console.log("Keep trying.");

}

**Example 3** – Check Light Color

let light = "green";

if (light === "green") {

console.log("Go");

} else if (light === "yellow") {

console.log("Slow down");

} else {

console.log("Stop");

}

**4. Activities**

Activity 1 – Try It Yourself  
Change the age in the first example to 17.  
What message appears? Try changing it to 25.

Activity 2 – Create a Traffic Signal  
Write a function that takes a color and shows:

* “Go” if green,
* “Slow down” if yellow,
* “Stop” if red.

**5. Exercises**

**Exercise 1** – Check if a Number is Positive or Negative  
Write a function that checks if a number is positive or negative.

Example:

function checkNumber(num) {

if (num >= 0) {

console.log("Positive");

} else {

console.log("Negative");

}

}

checkNumber(-5);

**Exercise 2** – Password Checker  
Write a function that checks if a password is correct.

Example:

function checkPassword(input) {

let correctPassword = "js123";

if (input === correctPassword) {

console.log("Access granted");

} else {

console.log("Access denied");

}

}

checkPassword("js123");

**Gamification for this Lesson**

Challenge of the Day:  
Create a function that checks your grade and returns a message:

* 90–100: "Excellent"
* 70–89: "Well done"
* Below 70: "Try again"
* Earn 3 stars if it works correctly with all 3 conditions.

**Lesson Summary**

* Conditional statements let your code make decisions.
* Use if to check if something is true.
* Use else if to check another condition if the first one is false.
* Use else when none of the other conditions are true.

**Quiz**

**1. What does an if statement do in JavaScript?**  
A. Loops the code  
B. Makes a decision  
C. Declares a function  
D. Stops the program

**2. What will this code output?**

let age = 20;

if (age >= 18) {

console.log("You are an adult.");

} else {

console.log("You are a minor.");

}

A. You are a minor.  
B. age is 20  
C. You are an adult.  
D. Nothing

**3. What will this code display if score = 60?**

if (score >= 90) {

console.log("Excellent!");

} else if (score >= 70) {

console.log("Good job!");

} else {

console.log("Keep trying.");

}

A. Excellent!  
B. Good job!  
C. Keep trying.  
D. Nothing

**4. What keyword checks another condition after if?**  
A. elseif  
B. then  
C. else if  
D. or

**5. What will happen if no condition in an if/else if block is true and there is no else block?**  
A. Code will crash  
B. Code inside if still runs  
C. Nothing happens  
D. It shows an alert

**6. What is the correct way to compare two values in an if statement?**  
A. if (a = b)  
B. if (a == b)  
C. if a equals b  
D. if (a => b)

**7. Which statement will check if a number is negative?**  
A. if (num > 0)  
B. if (num < 0)  
C. if (num == 0)  
D. if (num >= 0)

**Fill in the Blanks (2 Questions)**

**8**. The else block runs when all the other \_\_\_\_\_\_\_ are false.  
**9**. You use === to compare both value and \_\_\_\_\_\_\_ in JavaScript.

**Code Challenge (1 Question)**

**10. Write a function called checkGrade(score) that returns the following messages based on score:**

* 90–100: "Excellent"
* 70–89: "Well done"
* Below 70: "Try again"

**Answers**

**Multiple Choice**

1. **B**
2. **C**
3. **C**
4. **C**
5. **C**
6. **B**
7. **B**

**Fill in the Blanks**

1. if statements
2. type

**Code Answer**

function checkGrade(score) {

if (score >= 90) {

return "Excellent";

} else if (score >= 70) {

return "Well done";

} else {

return "Try again";

}

}

console.log(checkGrade(85)); // Output: Well done

**Grade 10**

Lesson 1: Working with Arrays and Objectives

**Lesson Goals (What you will learn):**

* Understand what arrays and objects are.
* See how arrays and objects help us organize information.
* Learn how to use arrays and objects in JavaScript.
* Use real-life examples to practice.

**What are Arrays and Objects?**

**Arrays**

* An array is a list of items stored in one variable.
* Think of it like a row of boxes with numbers (starting from 0).

**Example:**

let fruits = ["Apple", "Banana", "Cherry"];

console.log(fruits[1]); // Shows: Banana

**Objects**

* An object stores data in key-value pairs (like a name tag with info).
* Each key (or property) has a value.

**Example:**

let person = {

name: "John",

age: 30,

job: "Developer"

};

console.log(person.name); // Shows: John

**Examples**

**Example 1: Array of Fruits**

<script>

let fruits = ["apple", "banana", "cherry"];

fruits.push("orange"); // Add a new fruit

document.getElementById("demo").innerHTML = fruits;

</script>

**Example 2: Car Object**

<script>

let car = {

make: "Tesla",

model: "Model S",

year: 2022

};

car.color = "red"; // Add color

document.getElementById("demo").innerHTML = car.make + " " + car.model + ", Color: " + car.color;

</script>

**Example 3: Array of Student Objects**

<script>

let students = [

{ name: "John", age: 16 },

{ name: "Jane", age: 15 },

{ name: "Bob", age: 17 }

];

students.push({ name: "Alice", age: 18 });

let text = "";

for (let i = 0; i < students.length; i++) {

text += students[i].name + " is " + students[i].age + " years old.<br>";

}

document.getElementById("demo").innerHTML = text;

</script>

**Exercises**

**Exercise 1: Array of Colors**

**Task:**  
Create an array of 5 colors. Add one more color. Show all colors using a loop.

<script>

let colors = ["red", "blue", "green", "yellow", "orange"];

colors.push("purple");

let colorList = "";

for (let i = 0; i < colors.length; i++) {

colorList += colors[i] + "<br>";

}

document.getElementById("demo").innerHTML = colorList;

</script>

**Exercise 2: Book Object**

**Task:**  
Create an object for a book with title, author, and year. Add genre. Show the details.

<script>

let book = {

title: "JavaScript for Beginners",

author: "John Doe",

year: 2020

};

book.genre = "Programming";

let bookInfo = "Title: " + book.title + "<br>" +

"Author: " + book.author + "<br>" +

"Year: " + book.year + "<br>" +

"Genre: " + book.genre;

document.getElementById("demo").innerHTML = bookInfo;

</script>

**Gamification:  
 Challenge: Create an Array of your 3 favourite artists and loop through it**

**Stars earned: 1 star per correct item shown!**

**Summary**

* **Array** = A list of items. You can add more items using push() and go through them using loops.
  + Example: let fruits = ["Apple", "Banana"];
* **Object** = A group of info with labels (keys and values).
  + Example: let person = { name: "John", age: 30 };
* Use dot “.” to access object info: person.name

**Quiz: Working with Arrays and Objects in JavaScript**

**What is an array in JavaScript?**a) A collection of key-value pairs  
b) A collection of unordered values  
c) A collection of ordered values  
d) A function used to store data

**How do you access the first element in an array?**a) array[0]  
b) array[1]  
c) array[1st]  
d) array.first()

**What is an object in JavaScript?**a) A function that performs a specific task  
b) A collection of key-value pairs  
c) A list of numbers  
d) A type of array

**Which method is used to add an item to the end of an array?**a) push()  
b) pop()  
c) shift()  
d) unshift()

**What will the following code display?**let car = {  
 make: "Tesla",  
 model: "Model 3",  
 year: 2021  
};  
car.color = "blue";  
console.log(car.color);  
  
a) undefined  
b) blue  
c) Model 3  
d) Tesla

**How can you loop through an array of objects to display each object's property in JavaScript?**a) Using a for loop  
b) Using forEach()  
c) Using a while loop  
d) All of the above

**Which of the following is true about arrays in JavaScript?**a) Arrays are unordered collections  
b) Arrays are indexed by strings  
c) Arrays can store values of different data types  
d) Arrays can only store numbers

**What does the push() method do when applied to an array?**a) Adds an element to the beginning of the array  
b) Adds an element to the end of the array  
c) Removes an element from the beginning of the array  
d) Removes an element from the end of the array

**How do you access a property of an object in JavaScript?**a) object.propertyName  
b) object[propertyName]  
c) Both a and b  
d) None of the above

**Given the following code, what will be displayed?**let students = [  
 { name: "Alice", age: 17 },  
 { name: "Bob", age: 18 }  
];  
students.push({ name: "Charlie", age: 16 });  
console.log(students[2].name);  
  
a) Charlie  
b) Bob  
c) Alice  
d) undefined

**Answer Key:**

1 - c, 2 - a, 3 - b, 4 - a, 5 - b, 6 - d, 7 - c, 8 - b, 9 - c, 10 - a

**Lesson 2: Array Methods (push, pop, shift, unshift)**

**Lesson Goals (What you will learn):**

• Understand what common array methods do.  
• Learn how to add and remove items from arrays.  
• Practice using push, pop, shift, and unshift.  
• Use real-life examples to apply these methods.

**What are Array Methods?**

Arrays store a list of items. JavaScript gives us built-in methods to **add** or **remove** items easily:

**Common Methods:**

| **Method** | **What it does** |
| --- | --- |
| push() | Adds item to the **end** of the array |
| pop() | Removes item from the **end** of the array |
| shift() | Removes item from the **start** of the array |
| unshift() | Adds item to the **start** of the array |

**Examples**

**Example 1: Using push() and pop()**

<script>

let fruits = ["apple", "banana"];

fruits.push("cherry"); // Add to end

fruits.pop(); // Remove from end

document.getElementById("demo").innerHTML = fruits;

</script>

**Example 2: Using shift()**

<script>

let numbers = [10, 20, 30];

numbers.shift(); // Removes 10

document.getElementById("demo").innerHTML = numbers;

</script>

**Example 3: Using unshift()**

<script>

let colors = ["blue", "green"];

colors.unshift("red"); // Adds red at start

document.getElementById("demo").innerHTML = colors;

</script>

**Exercises**

**Exercise 1: Remove First Number**

**Task:**  
Create an array of 4 numbers. Remove the first number using shift().

<script>

let numbers = [5, 10, 15, 20];

numbers.shift();

document.getElementById("demo").innerHTML = numbers;

</script>

**Exercise 2: Add Colors**

**Task:**  
Create an array of 3 colors. Add 2 more colors using push().

<script>

let colors = ["yellow", "pink", "green"];

colors.push("blue");

colors.push("purple");

document.getElementById("demo").innerHTML = colors;

</script>

**Gamification:**

**Challenge:**  
Start with an array of 2 favorite foods. Use unshift() to add a new favorite food at the start, and push() to add another at the end.  
**Stars earned:** 1 star per correct method used!

**Summary**

• Use push() to add to the end  
• Use pop() to remove from the end  
• Use shift() to remove from the start  
• Use unshift() to add to the start  
• These methods help manage and update array content easily

**Quiz**

1. What does push() do?  
   a) Removes the first item  
   b) Adds to the beginning  
   c) Adds to the end  
   d) Removes from the end
2. What does shift() do?  
   a) Adds to the end  
   b) Removes from the end  
   c) Adds to the start  
   d) Removes from the start
3. Which method removes the last item?  
   a) pop()  
   b) push()  
   c) shift()  
   d) unshift()
4. Which method adds an item at the beginning of an array?  
   a) pop()  
   b) unshift()  
   c) push()  
   d) shift()
5. Given the code:

let items = ["pen", "pencil"];

items.unshift("eraser");

console.log(items[0]);

What will be shown?  
a) pencil  
b) pen  
c) eraser  
d) undefined

**Answer Key:**

1 - c  
2 - d  
3 - a  
4 - b  
5 - c

**Lesson 3: Looping Through Arrays (forEach)**

**Lesson Goals (What you will learn):**

• Understand what the forEach method does.  
• Learn how to use forEach to go through each item in an array.  
• Practice printing and working with array items.

**Recap:**

* **push() – add to the end**
* **pop() – remove from the end**
* **shift() – remove from the start**
* **unshift() – add to the start**

**What is forEach()?**

The forEach() method runs a function once for each item in an array. It's a simple and clean way to loop through arrays.

**Examples**

**Example 1: Loop through Colors**

<script>

let colors = ["red", "green", "blue"];

colors.forEach(function(color) {

document.getElementById("demo").innerHTML += color + "<br>";

});

</script>

**Example 2: Print Student Names**

<script>

let students = ["John", "Jane", "Sam"];

students.forEach(function(name) {

document.getElementById("demo").innerHTML += name + "<br>";

});

</script>

**Exercises**

**Exercise 1: List Days of the Week**  
Create an array of the 7 days of the week. Use forEach to print each one.

<script>

let days = ["Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday"];

days.forEach(function(day) {

document.getElementById("demo").innerHTML += day + "<br>";

});

</script>

**Exercise 2: Print Student Names**  
Create an array of 4 student names. Use forEach to display them.

<script>

let students = ["Alice", "Brian", "Chloe", "David"];

students.forEach(function(student) {

document.getElementById("demo").innerHTML += student + "<br>";

});

</script>

**Gamification:**

**Challenge:**  
Create an array of your 4 favourite movies. Use forEach to list them on the screen.  
**Stars earned:** 1 star per movie shown!

**Summary**

• forEach() is used to loop through all items in an array.  
• You provide a function to forEach() that runs for every item.  
• Clean and readable way to go through lists.

**Quiz:**

1. What does forEach() do?  
   a) Adds an item to the end of an array  
   b) Removes the first item  
   c) Loops through each item in the array  
   d) Finds the largest number in an array
2. How many times does forEach() run?  
   a) Just once  
   b) Until the last index  
   c) Once for each item  
   d) Only if a condition is true
3. Given this code:

let fruits = ["apple", "banana", "cherry"];

fruits.forEach(function(fruit) {

console.log(fruit);

});

What will be logged first?  
a) banana  
b) apple  
c) cherry  
d) undefined

1. In array.forEach(function(item) { ... }), what does item represent?  
   a) The name of the array  
   b) The total number of items  
   c) Each item in the array  
   d) The function itself
2. What will this display?

let pets = ["dog", "cat"];

pets.forEach(function(pet) {

console.log(pet + " is cute");

});

a) "dog"  
b) "cat"  
c) "dog is cute" and "cat is cute"  
d) Nothing

**Answer Key:**

1 - c  
2 - c  
3 - b  
4 - c  
5 - c