21. Use for loop to print each item in an array.

```
for(let i=0; i<students.length; i++) {
  console.log(students[i]);
}
```

22. What does the array.length property return?

It gives total number of items in array.

23. Use forEach to iterate over an array and print all items.

```
students.forEach(function(s) {
  console.log(s);
});
```

24. How do you check if a variable is an array?

```
Array.isArray(students);
```

25. Write an arrow function to multiply two numbers.

```
let multiply = (a, b) => a * b;
console.log(multiply(4, 5));
```

26. Convert a traditional function to an arrow function.

```
// Normal
function greet() {
  return "Hello";
}
// Arrow
let greet = () => "Hello";
```

27. Write an arrow function that returns the square of a number.

```
let square = n => n * n;
```

28. Create an arrow function that returns a greeting message.

```
let greet = name => `Hello, ${name}!`;

console.log(greet("John"));
```

29. Use an arrow function inside map() to double each number in an array.

```
let nums = [1, 2, 3];

let doubled = nums.map(n => n * 2);
```

30. What is the difference in this context between arrow and regular functions?

Arrow functions do not have their own this; instead, they inherit it from the surrounding context, which makes them useful in situations like callbacks where maintaining the outer this is important. In contrast, regular functions have their own this value that depends on how they are called. Another key difference is that arrow functions do not have their own arguments object, so we cannot access passed arguments using arguments inside them. Regular functions, however, do have access to the arguments object, which allows us to work with function parameters dynamically.

31. How do you add a click event to a button in JavaScript?

```
document.getElementById("myBtn").addEventListener("click", () => {

  alert("Clicked!");

});
```

32. Write JavaScript that changes the text of a paragraph when a button is clicked.

```
document.getElementById("btn").onclick = () => {

  document.getElementById("para").textContent = "Text Changed!";

};
```

```
});
```

33. How do you add a mouseover event to an element?

```
document.getElementById("box").addEventListener("mouseover", function() {
  console.log("Hovered");
});
```

34. Write an event handler that logs the value of an input field when typing.

```
document.getElementById("myInput").addEventListener("input", (e) => {
  console.log(e.target.value);
});
```

35. What’s the purpose of preventDefault() in event handling?

The purpose of preventDefault() in event handling is to stop the browser’s default action for a specific event from happening.

36. How do you get the value of an input field using JavaScript?

```
let value = document.getElementById("name").value;
```

37. Write JavaScript to validate if a form field is not empty.

```
let value = document.getElementById("username").value;
if (value === "") {
  alert("Field can't be empty");
}
```

38. How do you handle form submission using JavaScript?

```
document.getElementById("form").addEventListener("submit", (e) => {
  e.preventDefault();
  console.log("Form submitted");
});
```

39. How do you stop a form from refreshing the page when submitted?

Use event.preventDefault() inside submit event.

40. How can you reset a form using Javascript?

document.getElementById("form").reset();