

1. Basic Variable Declaration

javascript

```
let name = 'Alice';  
console.log(name); // Alice
```

2. String Concatenation

javascript

```
let greeting = 'Hello, ';  
let name = 'Bob';  
let message = greeting + name;  
console.log(message); // Hello, Bob
```

3. String Interpolation

javascript

```
let name = 'Charlie';  
let message = `Hello, ${name}`;  
console.log(message); // Hello, Charlie
```

4. Arithmetic Operations

javascript

```
let a = 10;  
let b = 5;  
let sum = a + b;  
let difference = a - b;  
let product = a * b;  
let quotient = a / b;  
  
console.log(sum); // 15  
console.log(difference); // 5  
console.log(product); // 50  
console.log(quotient); // 2
```

5. Conditional Statement (If-Else)

javascript

```
let age = 18;  
  
if (age >= 18) {  
    console.log('Adult');
```

```
} else {  
  console.log('Minor');  
}
```

6. Switch Statement

javascript

```
let day = 'Monday';
```

```
switch (day) {  
  case 'Monday':  
    console.log('Start of the week');  
    break;  
  case 'Friday':  
    console.log('End of the week');  
    break;  
  default:  
    console.log('Midweek');  
}
```

7. For Loop

javascript

```
for (let i = 0; i < 5; i++) {  
  console.log(i);  
}
```

8. While Loop

javascript

```
let i = 0;  
while (i < 5) {  
  console.log(i);  
  i++;  
}
```

9. Do-While Loop

javascript

```
let i = 0;  
do {  
  console.log(i);  
  i++;  
} while (i < 5);
```

10. Function Declaration

javascript

```
function greet(name) {  
  return 'Hello, ' + name;  
}
```

```
console.log(greet('Alice')); // Hello, Alice
```

11. Function Expression

javascript

```
const greet = function(name) {  
  return 'Hello, ' + name;  
};
```

```
console.log(greet('Bob')); // Hello, Bob
```

12. Arrow Function

javascript

```
const greet = (name) => 'Hello, ' + name;
```

```
console.log(greet('Charlie')); // Hello, Charlie
```

13. Default Parameters

javascript

```
function greet(name = 'Guest') {  
  return 'Hello, ' + name;  
}
```

```
console.log(greet()); // Hello, Guest
```

```
console.log(greet('Dave')); // Hello, Dave
```

14. Rest Parameters

javascript

```
function sum(...numbers) {  
  return numbers.reduce((total, num) => total + num, 0);  
}
```

```
console.log(sum(1, 2, 3, 4, 5)); // 15
```

15. Destructuring Assignment

javascript

```
let person = { name: 'Eve', age: 25 };  
let { name, age } = person;  
console.log(name); // Eve  
console.log(age); // 25
```

16. Array Destructuring

javascript

```
let numbers = [1, 2, 3];  
let [first, second, third] = numbers;  
console.log(first); // 1  
console.log(second); // 2  
console.log(third); // 3
```

17. Object Creation

javascript

```
let person = {  
  name: 'Alice',  
  age: 30  
};
```

```
console.log(person.name); // Alice  
console.log(person.age); // 30
```

18. Adding Properties to Objects

javascript

```
let person = {};  
person.name = 'Bob';  
person.age = 25;
```

```
console.log(person.name); // Bob  
console.log(person.age); // 25
```

19. Array Methods (Push, Pop)

javascript

```
let colors = ['red', 'green'];  
colors.push('blue');
```

```
console.log(colors); // ['red', 'green', 'blue']
```

```
colors.pop();  
console.log(colors); // ['red', 'green']
```

20. Array Methods (Shift, Unshift)

javascript

```
let numbers = [1, 2, 3];  
numbers.unshift(0);  
console.log(numbers); // [0, 1, 2, 3]
```

```
numbers.shift();  
console.log(numbers); // [1, 2, 3]
```

21. Array Methods (Slice)

javascript

```
let fruits = ['apple', 'banana', 'cherry'];  
let citrus = fruits.slice(1, 3);  
console.log(citrus); // ['banana', 'cherry']
```

22. Array Methods (Splice)

javascript

```
let animals = ['cat', 'dog', 'rabbit'];  
animals.splice(1, 1, 'hamster');  
console.log(animals); // ['cat', 'hamster', 'rabbit']
```

23. Array Methods (Map)

javascript

```
let numbers = [1, 2, 3];  
let doubled = numbers.map(num => num * 2);  
console.log(doubled); // [2, 4, 6]
```

24. Array Methods (Filter)

javascript

```
let numbers = [1, 2, 3, 4];  
let evenNumbers = numbers.filter(num => num % 2 === 0);  
console.log(evenNumbers); // [2, 4]
```

25. Array Methods (Reduce)

javascript

```
let numbers = [1, 2, 3, 4];  
let sum = numbers.reduce((total, num) => total + num, 0);  
console.log(sum); // 10
```

26. String Methods (toUpperCase)

javascript

```
let text = 'hello';  
console.log(text.toUpperCase()); // HELLO
```

27. String Methods (toLowerCase)

javascript

```
let text = 'HELLO';  
console.log(text.toLowerCase()); // hello
```

28. String Methods (includes)

javascript

```
let text = 'Hello, world!';  
console.log(text.includes('world')); // true
```

29. String Methods (substring)

javascript

```
let text = 'Hello, world!';  
console.log(text.substring(0, 5)); // Hello
```

30. String Methods (split)

javascript

```
let text = 'apple,banana,cherry';  
let fruits = text.split(',');  
console.log(fruits); // ['apple', 'banana', 'cherry']
```

31. String Methods (replace)

javascript

```
let text = 'Hello, world!';  
let newText = text.replace('world', 'JavaScript');  
console.log(newText); // Hello, JavaScript!
```

32. String Methods (trim)

javascript

```
let text = ' Hello ';  
console.log(text.trim()); // Hello
```

33. Date Object

javascript

```
let now = new Date();  
console.log(now); // Current date and time
```

34. Date Methods (getFullYear)

javascript

```
let now = new Date();  
console.log(now.getFullYear()); // Current year
```

35. Date Methods (getMonth)

javascript

```
let now = new Date();  
console.log(now.getMonth()); // Current month (0-11)
```

36. Date Methods (getDate)

javascript

```
let now = new Date();  
console.log(now.getDate()); // Current day of the month
```

37. Date Methods (getDay)

javascript

```
let now = new Date();  
console.log(now.getDay()); // Current day of the week (0-6)
```

38. Date Methods (getHours)

javascript

```
let now = new Date();  
console.log(now.getHours()); // Current hour
```

39. Date Methods (getMinutes)

javascript

```
let now = new Date();  
console.log(now.getMinutes()); // Current minutes
```

40. Date Methods (getSeconds)

javascript

```
let now = new Date();  
console.log(now.getSeconds()); // Current seconds
```

41. Math Methods (abs)

javascript

```
console.log(Math.abs(-5)); // 5
```

42. Math Methods (round)

javascript

```
console.log(Math.round(4.7)); // 5
```

43. Math Methods (floor)

javascript

```
console.log(Math.floor(4.7)); // 4
```

44. Math Methods (ceil)

javascript

```
console.log(Math.ceil(4.3)); // 5
```

45. Math Methods (random)

javascript

```
console.log(Math.random()); // Random number between 0 and 1
```

46. Math Methods (max)

javascript

```
console.log(Math.max(1, 5, 3)); // 5
```

47. Math Methods (min)

javascript


```
console.log(Math.min(1, 5, 3)); // 1
```

48. Event Listener (Click Event)

javascript

```
document.getElementById('myButton').addEventListener('click', () => {  
    alert('Button clicked!');  
});
```

49. Event Listener (Mouse Over Event)

javascript

```
document.getElementById('myElement').addEventListener('mouseover',  
() => {  
    console.log('Mouse over the element');  
});
```

50. Event Listener (Mouse Out Event)

javascript

```
document.getElementById('myElement').addEventListener('mouseout', ()  
=> {  
    console.log('Mouse out of the element');  
});
```

51. Event Listener (Keydown Event)

javascript

```
document.addEventListener('keydown', (event) => {  
    console.log('Key pressed:', event.key);  
});
```

52. Event Listener (Keyup Event)

javascript

```
document.addEventListener('keyup', (event) => {  
    console.log('Key released:', event.key);  
});
```

53. Event Listener (Submit Event)

javascript

```
document.getElementById('myForm').addEventListener('submit', (event)  
=> {
```

```
    event.preventDefault(); // Prevent form submission
    console.log('Form submitted');
  });
```

54. Event Listener (Change Event)

javascript

```
document.getElementById('mySelect').addEventListener('change',
(event) => {
  console.log('Selected value:', event.target.value);
});
```

55. Event Listener (Input Event)

javascript

```
document.getElementById('myInput').addEventListener('input', (event)
=> {
  console.log('Input value:', event.target.value);
});
```

56. Event Listener (Focus Event)

javascript

```
document.getElementById('myInput').addEventListener('focus', () => {
  console.log('Input focused');
});
```

57. Event Listener (Blur Event)

javascript

```
document.getElementById('myInput').addEventListener('blur', () => {
  console.log('Input lost focus');
});
```

58. Event Listener (Double Click Event)

javascript

```
document.getElementById('myElement').addEventListener('dblclick', ()
=> {
  console.log('Element double clicked');
});
```

59. Event Listener (Resize Event)

javascript

```
window.addEventListener('resize', () => {  
  console.log('Window resized');  
});
```

60. Event Listener (Scroll Event)

javascript

```
window.addEventListener('scroll', () => {  
  console.log('Page scrolled');  
});
```

61. Event Listener (Drag Start Event)

javascript

```
document.getElementById('myDraggable').addEventListener('dragstart',  
(() => {  
  console.log('Drag started');  
}));
```

62. Event Listener (Drop Event)

javascript

```
document.getElementById('dropZone').addEventListener('drop', (event)  
=> {  
  event.preventDefault();  
  console.log('Element dropped');  
}));
```

63. Event Listener (Drag Over Event)

javascript

```
document.getElementById('dropZone').addEventListener('dragover',  
(event) => {  
  event.preventDefault();  
  console.log('Drag over drop zone');  
});
```

64. Event Listener (Touch Start Event)

javascript

```
document.getElementById('myElement').addEventListener('touchstart', ()  
=> {  
  console.log('Touch started');  
});
```

```
});
```

65. Event Listener (Touch End Event)

javascript

```
document.getElementById('myElement').addEventListener('touchend', ()  
=> {  
  console.log('Touch ended');  
});
```

66. Event Listener (Touch Move Event)

javascript

```
document.getElementById('myElement').addEventListener('touchmove',  
() => {  
  console.log('Touch moved');  
});
```

67. Event Listener (Context Menu Event)

javascript

```
document.getElementById('myElement').addEventListener('contextmenu'  
, (event) => {  
  event.preventDefault();  
  console.log('Right-click menu prevented');  
});
```

68. Event Listener (Pointer Down Event)

javascript

```
document.getElementById('myElement').addEventListener('pointerdown'  
, () => {  
  console.log('Pointer down');  
});
```

69. Event Listener (Pointer Up Event)

javascript

```
document.getElementById('myElement').addEventListener('pointerup', ()  
=> {  
  console.log('Pointer up');  
});
```

70. Event Listener (Pointer Move Event)

javascript

```
document.getElementById('myElement').addEventListener('pointermove', () => {  
  console.log('Pointer moving');  
});
```

71. Event Listener (Animation Start Event)

javascript

```
document.getElementById('myElement').addEventListener('animationstart', () => {  
  console.log('Animation started');  
});
```

72. Event Listener (Animation End Event)

javascript

```
document.getElementById('myElement').addEventListener('animationend', () => {  
  console.log('Animation ended');  
});
```

73. Event Listener (Transition End Event)

javascript

```
document.getElementById('myElement').addEventListener('transitionend', () => {  
  console.log('Transition ended');  
});
```

74. Event Listener (Focus In Event)

javascript

```
document.getElementById('myInput').addEventListener('focusin', () => {  
  console.log('Input focused in');  
});
```

75. Event Listener (Focus Out Event)

javascript

```
document.getElementById('myInput').addEventListener('focusout', () => {  
  console.log('Input focused out');  
});
```

76. Event Listener (Load Event)

javascript

```
window.addEventListener('load', () => {  
  console.log('Page loaded');  
});
```

77. Event Listener (Unload Event)

javascript

```
window.addEventListener('unload', () => {  
  console.log('Page unloaded');  
});
```

78. Event Listener (Before Unload Event)

javascript

```
window.addEventListener('beforeunload', (event) => {  
  event.preventDefault();  
  event.returnValue = ""; // Required for most browsers  
  console.log('Page unload warning');  
});
```

79. Event Listener (Error Event)

javascript

```
window.addEventListener('error', (event) => {  
  console.error('Error occurred:', event.message);  
});
```

80. Event Listener (Message Event)

javascript

```
window.addEventListener('message', (event) => {  
  console.log('Message received:', event.data);  
});
```

81. Event Listener (Hash Change Event)

javascript

```
window.addEventListener('hashchange', () => {  
  console.log('Hash changed');  
});
```

82. Event Listener (Popstate Event)

javascript

```
window.addEventListener('popstate', () => {  
  console.log('History state changed');  
});
```

83. Event Listener (Pointer Cancel Event)

javascript

```
document.getElementById('myElement').addEventListener('pointercancel', () => {  
  console.log('Pointer canceled');  
});
```

84. Event Listener (Touch Cancel Event)

javascript

```
document.getElementById('myElement').addEventListener('touchcancel', () => {  
  console.log('Touch canceled');  
});
```

85. Event Listener (Transition Cancel Event)

javascript

```
document.getElementById('myElement').addEventListener('transitioncancel', () => {  
  console.log('Transition canceled');  
});
```

86. Event Listener (Change Event for Select)

javascript

```
document.getElementById('mySelect').addEventListener('change', (event) => {  
  console.log('Selected option:', event.target.value);  
});
```

87. Event Listener (Input Event for Textarea)

javascript

```
document.getElementById('myTextarea').addEventListener('input',
```

```
(event) => {  
  console.log('Textarea input:', event.target.value);  
});
```

88. Event Listener (Click Event on Button)

javascript

```
document.getElementById('myButton').addEventListener('click', () => {  
  console.log('Button clicked');  
});
```

89. Event Listener (Mouse Enter Event)

javascript

```
document.getElementById('myElement').addEventListener('mouseenter',  
( ) => {  
  console.log('Mouse entered element');  
});
```

90. Event Listener (Mouse Leave Event)

javascript

```
document.getElementById('myElement').addEventListener('mouseleave'  
, ( ) => {  
  console.log('Mouse left element');  
});
```

91. Event Listener (Context Menu Event)

javascript

```
document.getElementById('myElement').addEventListener('contextmenu'  
, (event) => {  
  event.preventDefault();  
  console.log('Right-click context menu disabled');  
});
```

92. Event Listener (Focus Event on Form Element)

javascript

```
document.getElementById('myFormInput').addEventListener('focus', ( )  
=> {  
  console.log('Form input focused');  
});
```


93. Event Listener (Blur Event on Form Element)

javascript

```
document.getElementById('myFormInput').addEventListener('blur', () => {
  console.log('Form input lost focus');
});
```

94. Event Listener (Submit Event on Form)

javascript

```
document.getElementById('myForm').addEventListener('submit', (event)
=> {
  event.preventDefault();
  console.log('Form submitted');
});
```

95. Event Listener (Keypress Event)

javascript

```
document.addEventListener('keypress', (event) => {
  console.log('Key pressed:', event.key);
});
```

96. Event Listener (Mouse Wheel Event)

javascript

```
document.addEventListener('wheel', (event) => {
  console.log('Mouse wheel used');
});
```

97. Event Listener (Animation Iteration Event)

javascript

```
document.getElementById('myElement').addEventListener('animationiter
ation', () => {
  console.log('Animation iteration');
});
```

98. Event Listener (Touch Cancel Event)

javascript

```
document.getElementById('myElement').addEventListener('touchcancel',
```

```
() => {  
  console.log('Touch canceled');  
});
```

99. Event Listener (Pointer Enter Event)

javascript

```
document.getElementById('myElement').addEventListener('pointerenter',  
( ) => {  
  console.log('Pointer entered element');  
});
```

100. Event Listener (Pointer Leave Event)

javascript

```
document.getElementById('myElement').addEventListener('pointerleave',  
( ) => {  
  console.log('Pointer left element');  
});
```

101. Basic Object Method

javascript

```
let person = {  
  name: 'Alice',  
  greet() {  
    return `Hello, ${this.name}`;  
  }  
};
```

```
console.log(person.greet()); // Hello, Alice
```

102. Object Constructor Function

javascript

```
function Person(name) {  
  this.name = name;  
  this.greet = function() {  
    return `Hello, ${this.name}`;  
  };  
}
```

```
let alice = new Person('Alice');
console.log(alice.greet()); // Hello, Alice
```

103. Class Definition

javascript

```
class Person {
  constructor(name) {
    this.name = name;
  }

  greet() {
    return `Hello, ${this.name}`;
  }
}
```

```
let alice = new Person('Alice');
console.log(alice.greet()); // Hello, Alice
```

104. Static Method in Class

javascript

```
class MathHelper {
  static add(x, y) {
    return x + y;
  }
}
```

```
console.log(MathHelper.add(5, 3)); // 8
```

105. Inheritance in Classes

javascript

```
class Animal {
  constructor(name) {
    this.name = name;
  }

  speak() {
    return `${this.name} makes a sound.`;
  }
}
```

```
class Dog extends Animal {
```

```
    speak() {  
      return `${this.name} barks.`;  
    }  
  }  
}
```

```
let myDog = new Dog('Rex');  
console.log(myDog.speak()); // Rex barks.
```

106. Promise Creation

javascript

```
let promise = new Promise((resolve, reject) => {  
  let success = true;  
  if (success) {  
    resolve('Operation successful');  
  } else {  
    reject('Operation failed');  
  }  
});
```

```
promise.then(result => {  
  console.log(result); // Operation successful  
}).catch(error => {  
  console.log(error); // Operation failed  
});
```

107. Async/Await Syntax

javascript

```
async function fetchData() {  
  return 'Data fetched';  
}
```

```
fetchData().then(result => {  
  console.log(result); // Data fetched  
});
```

108. Error Handling with Try-Catch

javascript

```
try {  
  let result = riskyOperation();  
  console.log(result);  
}
```

```
} catch (error) {  
  console.error('Error occurred:', error);  
}
```

```
function riskyOperation() {  
  throw new Error('Something went wrong');  
}
```

109. Set Creation

javascript

```
let mySet = new Set();  
mySet.add(1);  
mySet.add(2);  
mySet.add(2); // Duplicate value, will not be added
```

```
console.log(mySet); // Set { 1, 2 }
```

110. Map Creation

javascript

```
let myMap = new Map();  
myMap.set('key1', 'value1');  
myMap.set('key2', 'value2');
```

```
console.log(myMap.get('key1')); // value1  
console.log(myMap.has('key2')); // true
```

111. WeakSet Creation

javascript

```
let weakSet = new WeakSet();  
let obj = {};  
weakSet.add(obj);
```

```
console.log(weakSet.has(obj)); // true
```

112. WeakMap Creation

javascript

```
let weakMap = new WeakMap();  
let obj = {};  
weakMap.set(obj, 'value');
```

```
console.log(weakMap.get(obj)); // value
```

113. Array of Objects

javascript

```
let people = [  
  { name: 'Alice', age: 30 },  
  { name: 'Bob', age: 25 }  
];  
  
people.forEach(person => {  
  console.log(`${person.name} is ${person.age} years old.`);  
});
```

114. Array Methods (Find)

javascript

```
let numbers = [1, 2, 3, 4, 5];  
let found = numbers.find(num => num > 3);  
console.log(found); // 4
```

115. Array Methods (Find Index)

javascript

```
let numbers = [1, 2, 3, 4, 5];  
let index = numbers.findIndex(num => num > 3);  
console.log(index); // 3
```

116. Array Methods (Some)

javascript

```
let numbers = [1, 2, 3, 4, 5];  
let hasEven = numbers.some(num => num % 2 === 0);  
console.log(hasEven); // true
```

117. Array Methods (Every)

javascript

```
let numbers = [2, 4, 6, 8];  
let allEven = numbers.every(num => num % 2 === 0);  
console.log(allEven); // true
```

118. Array Methods (Sort)

javascript

```
let numbers = [4, 2, 5, 1, 3];
numbers.sort((a, b) => a - b);
console.log(numbers); // [1, 2, 3, 4, 5]
```

119. Array Methods (Reverse)

javascript

```
let numbers = [1, 2, 3];
numbers.reverse();
console.log(numbers); // [3, 2, 1]
```

120. Array Methods (Reduce Right)

javascript

```
let numbers = [1, 2, 3];
let sum = numbers.reduceRight((total, num) => total + num, 0);
console.log(sum); // 6
```

121. Array Methods (Fill)

javascript

```
let numbers = [1, 2, 3, 4, 5];
numbers.fill(0, 2, 4);
console.log(numbers); // [1, 2, 0, 0, 5]
```

122. Array Methods (Copy Within)

javascript

```
let numbers = [1, 2, 3, 4, 5];
numbers.copyWithin(0, 3);
console.log(numbers); // [4, 5, 3, 4, 5]
```

123. Date Object Methods (setFullYear)

javascript

```
let date = new Date();
date.setFullYear(2025);
console.log(date.getFullYear()); // 2025
```

124. Date Object Methods (setMonth)

javascript

```
let date = new Date();
```

```
date.setMonth(11); // December
console.log(date.getMonth()); // 11
```

125. Date Object Methods (setDate)

javascript

```
let date = new Date();
date.setDate(15);
console.log(date.getDate()); // 15
```

126. Date Object Methods (setHours)

javascript

```
let date = new Date();
date.setHours(10);
console.log(date.getHours()); // 10
```

127. Date Object Methods (setMinutes)

javascript

```
let date = new Date();
date.setMinutes(30);
console.log(date.getMinutes()); // 30
```

128. Date Object Methods (setSeconds)

javascript

```
let date = new Date();
date.setSeconds(45);
console.log(date.getSeconds()); // 45
```

129. Template Literals

javascript

```
let name = 'Alice';
let age = 30;
let message = `Name: ${name}, Age: ${age}`;
console.log(message); // Name: Alice, Age: 30
```

130. Tagged Templates

javascript

```
function highlight(strings, ...values) {
  return strings.reduce((acc, str, i) => acc + (i ? `<strong>${values[i - 1]}`
```



```
</strong>${str}` : str), "");  
}
```

```
let name = 'Alice';  
let age = 30;  
let message = highlight`Name: ${name}, Age: ${age}`;  
console.log(message); // Name: <strong>Alice</strong>, Age:  
<strong>30</strong>
```

131. Map Iteration

javascript

```
let myMap = new Map();  
myMap.set('key1', 'value1');  
myMap.set('key2', 'value2');  
  
for (let [key, value] of myMap) {  
  console.log(`${key}: ${value}`);  
}  
// key1: value1  
// key2: value2
```

132. Set Iteration

javascript

```
let mySet = new Set([1, 2, 3]);  
mySet.add(4);  
  
for (let value of mySet) {  
  console.log(value);  
}  
// 1  
// 2  
// 3  
// 4
```

133. WeakMap Iteration (Not Directly Possible)

javascript

```
// WeakMaps do not support direct iteration.  
// They are meant for holding weak references.
```

134. WeakSet Iteration (Not Directly Possible)

javascript

// WeakSets do not support direct iteration.
// They are meant for holding weak references.

135. Handling Errors in Promises

javascript

```
let promise = new Promise((resolve, reject) => {  
  reject('Error occurred');  
});
```

```
promise.catch(error => {  
  console.error('Caught error:', error);  
});
```

136. Chaining Promises

javascript

```
let promise = new Promise((resolve, reject) => {  
  resolve('Success');  
});
```

```
promise  
  .then(result => result + '!')  
  .then(result => console.log(result)); // Success!
```

137. Combining Promises

javascript

```
let promise1 = Promise.resolve('First');  
let promise2 = Promise.resolve('Second');
```

```
Promise.all([promise1, promise2]).then(results => {  
  console.log(results); // [ 'First', 'Second' ]  
});
```

138. Promise Race

javascript

```
let promise1 = new Promise((resolve) => setTimeout(resolve, 500,  
'First'));  
let promise2 = new Promise((resolve) => setTimeout(resolve, 100,  
'Second'));
```

```
Promise.race([promise1, promise2]).then(result => {
  console.log(result); // Second
});
```

139. Async Function with Error Handling

javascript

```
async function fetchData() {
  try {
    let response = await fetch('https://api.example.com/data');
    let data = await response.json();
    console.log(data);
  } catch (error) {
    console.error('Error fetching data:', error);
  }
}
```

```
fetchData();
```

140. Debouncing Function

javascript

```
function debounce(func, delay) {
  let timer;
  return function(...args) {
    clearTimeout(timer);
    timer = setTimeout(() => func.apply(this, args), delay);
  };
}
```

```
let handleResize = debounce(() => {
  console.log('Resized');
}, 500);
```

```
window.addEventListener('resize', handleResize);
```

141. Throttling Function

javascript

```
function throttle(func, limit) {
  let lastFunc;
  let lastRan;
  return function() {
```

```

const context = this;
const args = arguments;
if (!lastRan) {
  func.apply(context, args);
  lastRan = Date.now();
} else {
  clearTimeout(lastFunc);
  lastFunc = setTimeout(() => {
    if ((Date.now() - lastRan) >= limit) {
      func.apply(context, args);
      lastRan = Date.now();
    }
  }, limit - (Date.now() - lastRan));
}
};
}

let handleScroll = throttle(() => {
  console.log('Scrolled');
}, 200);

```

```

window.addEventListener('scroll', handleScroll);

```

142. Event Delegation

javascript

```

document.getElementById('parent').addEventListener('click', (event) => {
  if (event.target && event.target.matches('button.child')) {
    console.log('Child button clicked');
  }
});

```

143. Custom Event Creation

javascript

```

let event = new CustomEvent('myEvent', { detail: { message: 'Hello World' } });
document.dispatchEvent(event);

document.addEventListener('myEvent', (e) => {
  console.log(e.detail.message); // Hello World
});

```

144. Event Bubbling

javascript

```
document.getElementById('child').addEventListener('click', () => {  
  console.log('Child clicked');  
});
```

```
document.getElementById('parent').addEventListener('click', () => {  
  console.log('Parent clicked');  
});
```

145. Event Capturing

javascript

```
document.getElementById('parent').addEventListener('click', () => {  
  console.log('Parent clicked');  
}, true);
```

```
document.getElementById('child').addEventListener('click', () => {  
  console.log('Child clicked');  
});
```

146. Event Removal

javascript

```
function handleClick() {  
  console.log('Button clicked');  
}
```

```
let button = document.getElementById('myButton');  
button.addEventListener('click', handleClick);  
button.removeEventListener('click', handleClick);
```

147. Form Data Handling

javascript

```
document.getElementById('myForm').addEventListener('submit', (event)  
=> {  
  event.preventDefault();  
  let formData = new FormData(event.target);  
  console.log(formData.get('username')); // Username input value  
});
```

148. Fetching Data with Fetch API

javascript

```
fetch('https://api.example.com/data')
  .then(response => response.json())
  .then(data => console.log(data));
```

149. Abort Fetch Request

javascript

```
let controller = new AbortController();
let signal = controller.signal;

fetch('https://api.example.com/data', { signal })
  .then(response => response.json())
  .then(data => console.log(data))
  .catch(error => {
    if (error.name === 'AbortError') {
      console.log('Fetch aborted');
    } else {
      console.error('Fetch error:', error);
    }
  });
```

```
controller.abort(); // Abort the fetch request
```

150. Local Storage

javascript

```
localStorage.setItem('name', 'Alice');
console.log(localStorage.getItem('name')); // Alice
localStorage.removeItem('name');
```

151. Session Storage

javascript

```
sessionStorage.setItem('sessionName', 'Bob');
console.log(sessionStorage.getItem('sessionName')); // Bob
sessionStorage.removeItem('sessionName');
```

152. Cookies

javascript

```
document.cookie = 'username=Alice; expires=Fri, 31 Dec 9999 23:59:59 GMT';
console.log(document.cookie); // username=Alice
```

153. Service Workers (Basic Registration)

javascript

```
if ('serviceWorker' in navigator) {
  navigator.serviceWorker.register('/service-worker.js')
    .then(registration => {
      console.log('Service Worker registered with scope:',
registration.scope);
    })
    .catch(error => {
      console.error('Service Worker registration failed:', error);
    });
}
```

154. WebSocket Communication

javascript

```
let socket = new WebSocket('ws://example.com/socketserver');
```

```
socket.onopen = () => {
  console.log('WebSocket connection established');
  socket.send('Hello Server');
};
```

```
socket.onmessage = (event) => {
  console.log('Message from server:', event.data);
};
```

```
socket.onerror = (error) => {
  console.error('WebSocket error:', error);
};
```

```
socket.onclose = () => {
  console.log('WebSocket connection closed');
};
```

155. Intersection Observer

javascript

```

let observer = new IntersectionObserver((entries) => {
  entries.forEach(entry => {
    if (entry.isIntersecting) {
      console.log('Element is visible');
    }
  });
});

```

```

let target = document.querySelector('#targetElement');
observer.observe(target);

```

156. Mutation Observer

javascript

```

let targetNode = document.getElementById('targetNode');
let observer = new MutationObserver((mutations) => {
  mutations.forEach(mutation => {
    console.log('Mutation observed:', mutation);
  });
});

```

```

observer.observe(targetNode, { attributes: true, childList: true, subtree:
true });

```

157. Custom Elements

javascript

```

class MyElement extends HTMLElement {
  constructor() {
    super();
    this.innerHTML = '<p>Custom Element</p>';
  }
}

```

```

customElements.define('my-element', MyElement);

```

```

document.body.innerHTML = '<my-element></my-element>';

```

158. Shadow DOM

javascript

```

class ShadowElement extends HTMLElement {
  constructor() {

```



```

    super();
    let shadow = this.attachShadow({ mode: 'open' });
    shadow.innerHTML = '<p>Shadow DOM content</p>';
  }
}

```

```

customElements.define('shadow-element', ShadowElement);

```

```

document.body.innerHTML = '<shadow-element></shadow-element>';

```

159. Form Validation

javascript

```

document.getElementById('myForm').addEventListener('submit', (event)
=> {
  let isValid = document.getElementById('myInput').checkValidity();
  if (!isValid) {
    event.preventDefault();
    console.log('Form is invalid');
  }
});

```

160. Drag and Drop

javascript

```

document.getElementById('draggable').addEventListener('dragstart',
(event) => {
  event.dataTransfer.setData('text/plain', event.target.id);
});

```

```

document.getElementById('dropzone').addEventListener('drop', (event)
=> {
  event.preventDefault();
  let data = event.dataTransfer.getData('text/plain');
  let draggable = document.getElementById(data);
  event.target.appendChild(draggable);
});

```

```

document.getElementById('dropzone').addEventListener('dragover',
(event) => {
  event.preventDefault();
});

```

Here are additional examples focusing on form validation in JavaScript:

1. Basic Form Validation

html

```
<form id="myForm">
  <input type="text" id="username" required placeholder="Username">
  <input type="submit" value="Submit">
</form>
```

```
<script>
  document.getElementById('myForm').addEventListener('submit',
(event) => {
    let username = document.getElementById('username').value;
    if (!username) {
      alert('Username is required');
      event.preventDefault();
    }
  });
</script>
```

2. Validation for Email Format

html

```
<form id="emailForm">
  <input type="email" id="email" required placeholder="Email">
  <input type="submit" value="Submit">
</form>
```

```
<script>
  document.getElementById('emailForm').addEventListener('submit',
(event) => {
    let email = document.getElementById('email').value;
    if (!email || !email.includes('@')) {
      alert('Please enter a valid email address');
      event.preventDefault();
    }
  });
</script>
```

3. Validation for Minimum Length

html

```

<form id="passwordForm">
  <input type="password" id="password" minlength="8" required
placeholder="Password">
  <input type="submit" value="Submit">
</form>

<script>
  document.getElementById('passwordForm').addEventListener('submit',
(event) => {
    let password = document.getElementById('password').value;
    if (password.length < 8) {
      alert('Password must be at least 8 characters long');
      event.preventDefault();
    }
  });
</script>

```

4. Validation for Confirm Password

html

```

<form id="confirmPasswordForm">
  <input type="password" id="password" required
placeholder="Password">
  <input type="password" id="confirmPassword" required
placeholder="Confirm Password">
  <input type="submit" value="Submit">
</form>

<script>

document.getElementById('confirmPasswordForm').addEventListener('s
ubmit', (event) => {
  let password = document.getElementById('password').value;
  let confirmPassword =
document.getElementById('confirmPassword').value;
  if (password !== confirmPassword) {
    alert('Passwords do not match');
    event.preventDefault();
  }
});
</script>

```

5. Validation for Radio Buttons

html

```
<form id="radioForm">
  <label><input type="radio" name="gender" value="male"> Male</label>
  <label><input type="radio" name="gender" value="female"> Female</
label>
  <input type="submit" value="Submit">
</form>
```

```
<script>
  document.getElementById('radioForm').addEventListener('submit',
(event) => {
    let gender =
document.querySelector('input[name="gender"]:checked');
    if (!gender) {
      alert('Please select a gender');
      event.preventDefault();
    }
  });
</script>
```

6. Validation for Checkbox

html

```
<form id="checkboxForm">
  <label><input type="checkbox" id="terms" required> I agree to the
terms and conditions</label>
  <input type="submit" value="Submit">
</form>
```

```
<script>
  document.getElementById('checkboxForm').addEventListener('submit',
(event) => {
    let terms = document.getElementById('terms').checked;
    if (!terms) {
      alert('You must agree to the terms and conditions');
      event.preventDefault();
    }
  });
</script>
```

7. Validation for Number Range

html

```
<form id="ageForm">
  <input type="number" id="age" min="18" max="100" required
placeholder="Age">
  <input type="submit" value="Submit">
</form>

<script>
  document.getElementById('ageForm').addEventListener('submit',
(event) => {
    let age = document.getElementById('age').value;
    if (age < 18 || age > 100) {
      alert('Age must be between 18 and 100');
      event.preventDefault();
    }
  });
</script>
```

8. Validation for Date

html

```
<form id="dateForm">
  <input type="date" id="date" required>
  <input type="submit" value="Submit">
</form>

<script>
  document.getElementById('dateForm').addEventListener('submit',
(event) => {
    let date = document.getElementById('date').value;
    if (!date) {
      alert('Please select a date');
      event.preventDefault();
    }
  });
</script>
```

9. Validation for URL

html

```

<form id="urlForm">
  <input type="url" id="website" required placeholder="Website URL">
  <input type="submit" value="Submit">
</form>

<script>
  document.getElementById('urlForm').addEventListener('submit', (event)
=> {
    let website = document.getElementById('website').value;
    if (!website) {
      alert('Please enter a valid URL');
      event.preventDefault();
    }
  });
</script>

```

10. Validation with Constraint Validation API

html

```

<form id="constraintForm">
  <input type="text" id="name" required pattern="[A-Za-z]+"
placeholder="Name">
  <input type="submit" value="Submit">
</form>

<script>
  document.getElementById('constraintForm').addEventListener('submit',
(event) => {
    let name = document.getElementById('name');
    if (!name.validity.valid) {
      alert('Name must contain only letters');
      event.preventDefault();
    }
  });
</script>

```

11. Validation for Custom Error Messages

html

```

<form id="customErrorForm">
  <input type="text" id="username" required placeholder="Username"
aria-describedby="usernameError">
  <div id="usernameError" style="color: red;"></div>

```

```
<input type="submit" value="Submit">
</form>
```

```
<script>
```

```
document.getElementById('customErrorForm').addEventListener('submit', (event) => {
  let username = document.getElementById('username');
  let error = document.getElementById('usernameError');
  if (!username.value) {
    error.textContent = 'Username is required';
    event.preventDefault();
  } else {
    error.textContent = "";
  }
});
```

```
</script>
```

12. Validation for File Upload

html

```
<form id="fileForm">
  <input type="file" id="file" accept="image/*" required>
  <input type="submit" value="Submit">
</form>
```

```
<script>
  document.getElementById('fileForm').addEventListener('submit', (event) => {
    let file = document.getElementById('file').files[0];
    if (!file) {
      alert('Please select a file');
      event.preventDefault();
    }
  });
</script>
```

13. Validation with Custom Patterns

html

```
<form id="customPatternForm">
  <input type="text" id="zipcode" pattern="\d{5}" placeholder="ZIP Code"
  required>
```

```
<input type="submit" value="Submit">
</form>
```

```
<script>
```

```
document.getElementById('customPatternForm').addEventListener('submit', (event) => {
    let zipcode = document.getElementById('zipcode');
    if (!zipcode.checkValidity()) {
        alert('ZIP Code must be 5 digits');
        event.preventDefault();
    }
});
```

```
</script>
```

14. Validation for Multiple Fields

html

```
<form id="multiFieldForm">
    <input type="text" id="name" required placeholder="Name">
    <input type="email" id="email" required placeholder="Email">
    <input type="password" id="password" minlength="6" required
placeholder="Password">
    <input type="submit" value="Submit">
</form>
```

```
<script>
```

```
document.getElementById('multiFieldForm').addEventListener('submit',
(event) => {
    let name = document.getElementById('name').value;
    let email = document.getElementById('email').value;
    let password = document.getElementById('password').value;

    if (!name || !email || !password) {
        alert('All fields are required');
        event.preventDefault();
    }
});
```

```
</script>
```

15. Validation for Time

html


```
<form id="timeForm">
  <input type="time" id="time" required>
  <input type="submit" value="Submit">
</form>
```

```
<script>
  document.getElementById('timeForm').addEventListener('submit',
(event) => {
    let time = document.getElementById('time').value;
    if (!time) {
      alert('Please select a time');
      event.preventDefault();
    }
  });
</script>
```

16. Validation with HTML5 Attributes

html

```
<form id="html5AttrForm">
  <input type="text" id="username" required maxlength="20"
placeholder="Username">
  <input type="number" id="age" min="1" max="120" required
placeholder="Age">
  <input type="submit" value="Submit">
</form>
```

```
<script>
  document.getElementById('html5AttrForm').addEventListener('submit',
(event) => {
    let username = document.getElementById('username').value;
    let age = document.getElementById('age').value;

    if (username.length > 20) {
      alert('Username must be 20 characters or less');
      event.preventDefault();
    }
    if (age < 1 || age > 120) {
      alert('Age must be between 1 and 120');
      event.preventDefault();
    }
  });
</script>
```

17. Custom Validation Using setCustomValidity

html

```
<form id="customValidityForm">
  <input type="text" id="customInput" required placeholder="Enter text">
  <input type="submit" value="Submit">
</form>
```

<script>

```
document.getElementById('customValidityForm').addEventListener('submit', (event) => {
  let customInput = document.getElementById('customInput');
  if (customInput.value === 'invalid') {
    customInput.setCustomValidity('This value is not allowed');
    event.preventDefault();
  } else {
    customInput.setCustomValidity('');
  }
});
```

</script>

18. Validation for Multiple Checkbox Selections

html

```
<form id="checkboxGroupForm">
  <label><input type="checkbox" name="interest" value="tech"> Tech</label>
  <label><input type="checkbox" name="interest" value="sports"> Sports</label>
  <label><input type="checkbox" name="interest" value="music"> Music</label>
  <input type="submit" value="Submit">
</form>
```

<script>

```
document.getElementById('checkboxGroupForm').addEventListener('submit', (event) => {
  let interests =
document.querySelectorAll('input[name="interest"]:checked');
  if (interests.length < 1) {
```

```
    alert('Please select at least one interest');
    event.preventDefault();
  }
});
</script>
```

Here are some general JavaScript examples that are commonly used in HTML and CSS pages:

1. Toggle Visibility

html

```
<button id="toggleButton">Toggle Text</button>
<p id="toggleText">This is some text.</p>
```

```
<script>
  document.getElementById('toggleButton').addEventListener('click', ()
=> {
    let text = document.getElementById('toggleText');
    text.style.display = (text.style.display === 'none') ? 'block' : 'none';
  });
</script>
```

2. Change Text Content

html

```
<button id="changeTextButton">Change Text</button>
<p id="text">Original Text</p>
```

```
<script>

document.getElementById('changeTextButton').addEventListener('click',
() => {
  document.getElementById('text').textContent = 'Text Changed!';
});
</script>
```

3. Change Background Color

html

```
<button id="changeColorButton">Change Background Color</button>
```

```
<script>
```

```
document.getElementById('changeColorButton').addEventListener('click', () => {  
  document.body.style.backgroundColor = 'lightblue';  
});  
</script>
```

4. Form Submission Alert

html

```
<form id="alertForm">  
  <input type="text" id="name" placeholder="Your Name">  
  <input type="submit" value="Submit">  
</form>
```

```
<script>  
  document.getElementById('alertForm').addEventListener('submit', (event) => {  
    alert('Form Submitted');  
    event.preventDefault(); // Prevent form from submitting for demonstration  
  });  
</script>
```

5. Image Slider

html

```
<button id="prev">Previous</button>  
  
<button id="next">Next</button>
```

```
<script>  
  let images = ['image1.jpg', 'image2.jpg', 'image3.jpg'];  
  let index = 0;  
  
  document.getElementById('prev').addEventListener('click', () => {  
    index = (index > 0) ? index - 1 : images.length - 1;  
    document.getElementById('slider').src = images[index];  
  });  
  
  document.getElementById('next').addEventListener('click', () => {  
    index = (index < images.length - 1) ? index + 1 : 0;
```

```
    document.getElementById('slider').src = images[index];
  });
</script>
```

6. Dynamic Content Loading

html

```
<button id="loadContent">Load Content</button>
<div id="content"></div>
```

```
<script>
  document.getElementById('loadContent').addEventListener('click', () =>
  {
    document.getElementById('content').innerHTML = '<p>New content
loaded!</p>';
  });
</script>
```

7. Scroll to Top Button

html

```
<button id="scrollTopButton" style="display: none;">Scroll to Top</
button>
```

```
<script>
  window.addEventListener('scroll', () => {
    let button = document.getElementById('scrollTopButton');
    button.style.display = (window.scrollY > 200) ? 'block' : 'none';
  });

  document.getElementById('scrollTopButton').addEventListener('click',
  () => {
    window.scrollTo({ top: 0, behavior: 'smooth' });
  });
</script>
```

8. Modal Window

html

```
<button id="openModal">Open Modal</button>
<div id="modal" style="display: none; position: fixed; top: 50%; left: 50%;
transform: translate(-50%, -50%); padding: 20px; background: white;
border: 1px solid black;">
```

```
<p>This is a modal window</p>
<button id="closeModal">Close</button>
</div>
```

```
<script>
  document.getElementById('openModal').addEventListener('click', () =>
  {
    document.getElementById('modal').style.display = 'block';
  });

  document.getElementById('closeModal').addEventListener('click', () =>
  {
    document.getElementById('modal').style.display = 'none';
  });
</script>
```

9. Accordion Effect

html

```
<button class="accordion">Section 1</button>
<div class="panel" style="display: none;">
  <p>Content for section 1.</p>
</div>
```

```
<button class="accordion">Section 2</button>
<div class="panel" style="display: none;">
  <p>Content for section 2.</p>
</div>
```

```
<script>
  document.querySelectorAll('.accordion').forEach(button => {
    button.addEventListener('click', () => {
      let panel = button.nextElementSibling;
      panel.style.display = (panel.style.display === 'block') ? 'none' :
'block';
    });
  });
</script>
```

10. Countdown Timer

html

```
<p id="timer">10</p>
```

```

<script>
  let count = 10;
  let timer = document.getElementById('timer');

  setInterval(() => {
    if (count > 0) {
      count--;
      timer.textContent = count;
    }
  }, 1000);
</script>

```

11. Drag and Drop Example

html

```

<div id="dragItem" draggable="true" style="width: 100px; height: 100px;
background: red;"></div>
<div id="dropZone" style="width: 200px; height: 200px; background:
blue; margin-top: 20px;"></div>

```

```

<script>
  document.getElementById('dragItem').addEventListener('dragstart',
(event) => {
    event.dataTransfer.setData('text/plain', 'dragItem');
  });

  document.getElementById('dropZone').addEventListener('dragover',
(event) => {
    event.preventDefault();
  });

  document.getElementById('dropZone').addEventListener('drop', (event)
=> {
    event.preventDefault();
    let data = event.dataTransfer.getData('text/plain');
    if (data === 'dragItem') {
      event.target.appendChild(document.getElementById(data));
    }
  });
</script>

```

12. Form Autofill

html

```
<form id="autofillForm">
```

```
  <input type="text" id="name" placeholder="Name">
```

```
  <input type="text" id="email" placeholder="Email">
```

```
  <input type="submit" value="Submit">
```

```
</form>
```

```
<script>
```

```
  document.getElementById('autofillForm').addEventListener('submit',  
(event) => {
```

```
    event.preventDefault(); // Prevent actual submission for demo
```

```
    document.getElementById('name').value = 'John Doe';
```

```
    document.getElementById('email').value = 'john.doe@example.com';
```

```
  });
```

```
</script>
```

13. Dynamic Style Changes

html

```
<button id="styleButton">Change Styles</button>
```

```
<p id="styledText">This text will change style.</p>
```

```
<script>
```

```
  document.getElementById('styleButton').addEventListener('click', () =>  
{
```

```
    let text = document.getElementById('styledText');
```

```
    text.style.color = 'green';
```

```
    text.style.fontSize = '24px';
```

```
    text.style.fontWeight = 'bold';
```

```
  });
```

```
</script>
```

14. Form Field Focus

html

```
<form id="focusForm">
```

```
  <input type="text" id="input1" placeholder="Focus me first">
```

```
  <input type="text" id="input2" placeholder="Focus me second">
```

```
  <input type="submit" value="Submit">
```

```
</form>
```

```
<script>
```



```
document.getElementById('focusForm').addEventListener('submit',  
(event) => {  
  event.preventDefault(); // Prevent actual submission for demo  
  document.getElementById('input2').focus();  
});  
</script>
```

15. Image Hover Effect

html

```

```

```
<script>  
  let image = document.getElementById('hoverImage');  
  
  image.addEventListener('mouseover', () => {  
    image.style.opacity = 0.5;  
  });  
  
  image.addEventListener('mouseout', () => {  
    image.style.opacity = 1;  
  });  
</script>
```

16. Character Count

html

```
<textarea id="textarea" rows="4" cols="50" placeholder="Type  
here..."></textarea>  
<p id="charCount">Characters remaining: 100</p>
```

```
<script>  
  const textarea = document.getElementById('textarea');  
  const charCount = document.getElementById('charCount');  
  const maxLength = 100;  
  
  textarea.addEventListener('input', () => {  
    let remaining = maxLength - textarea.value.length;  
    charCount.textContent = `Characters remaining: ${remaining}`;  
  });  
</script>
```

17. Responsive Navigation Menu

html

```
<button id="menuButton">Menu</button>
<nav id="navMenu" style="display: none;">
  <a href="#">Home</a>
  <a href="#">About</a>
  <a href="#">Services</a>
  <a href="#">Contact</a>
</nav>
```

```
<script>
  document.getElementById('menuButton').addEventListener('click', () =>
  {
    let navMenu = document.getElementById('navMenu');
    navMenu.style.display = (navMenu.style.display === 'none') ? 'block' :
'none';
  });
</script>
```

18. Content Editable Div

html

```
<div id="editableDiv" contenteditable="true" style="border: 1px solid
black; padding: 10px;">
  Edit this text...
</div>
```

```
<script>
  document.getElementById('editableDiv').addEventListener('input', () =>
  {
    console.log('Content changed:',
document.getElementById('editableDiv').innerHTML);
  });
</script>
```

19. Form Data Display

html

```
<form id="displayForm">
  <input type="text" id="firstName" placeholder="First Name">
  <input type="text" id="lastName" placeholder="Last Name">
```

```

    <input type="submit" value="Display Data">
</form>
<p id="formData"></p>

<script>
  document.getElementById('displayForm').addEventListener('submit',
(event) => {
    event.preventDefault();
    let firstName = document.getElementById('firstName').value;
    let lastName = document.getElementById('lastName').value;
    document.getElementById('formData').textContent = `Full Name: $
{firstName} ${lastName}`;
  });
</script>

```

20. Checkbox Toggle

html

```

<input type="checkbox" id="toggleCheckbox">
<label for="toggleCheckbox">Check me to toggle visibility</label>
<div id="toggleDiv" style="display: none;">Now you see me!</div>

```

```

<script>

document.getElementById('toggleCheckbox').addEventListener('change'
, (event) => {
  let toggleDiv = document.getElementById('toggleDiv');
  toggleDiv.style.display = event.target.checked ? 'block' : 'none';
});
</script>

```

21. Date Picker

html

```

<input type="date" id="datePicker">
<p id="selectedDate"></p>

```

```

<script>
  document.getElementById('datePicker').addEventListener('change', ()
=> {
    let selectedDate = document.getElementById('datePicker').value;
    document.getElementById('selectedDate').textContent = `Selected
Date: ${selectedDate}`;
  });

```

```
});  
</script>
```

22. Email Validation

html

```
<form id="emailForm">  
  <input type="email" id="email" required placeholder="Enter your  
email">  
  <input type="submit" value="Submit">  
</form>  
<p id="emailError" style="color: red;"></p>  
  
<script>  
  document.getElementById('emailForm').addEventListener('submit',  
(event) => {  
    let email = document.getElementById('email').value;  
    let emailError = document.getElementById('emailError');  
    if (!email.includes('@')) {  
      emailError.textContent = 'Please enter a valid email address';  
      event.preventDefault();  
    } else {  
      emailError.textContent = "";  
    }  
  });  
</script>
```

23. Loading Spinner

html

```
<button id="showSpinner">Show Spinner</button>  
<div id="spinner" style="display: none;">Loading...</div>  
  
<script>  
  document.getElementById('showSpinner').addEventListener('click', ()  
=> {  
    document.getElementById('spinner').style.display = 'block';  
    setTimeout(() => {  
      document.getElementById('spinner').style.display = 'none';  
    }, 2000); // Hide spinner after 2 seconds  
  });  
</script>
```

24. Real-Time Search Filtering

html

```
<input type="text" id="search" placeholder="Search...">
<ul id="itemList">
  <li>Apple</li>
  <li>Banana</li>
  <li>Cherry</li>
  <li>Date</li>
</ul>
```

```
<script>
  document.getElementById('search').addEventListener('input', () => {
    let searchValue =
document.getElementById('search').value.toLowerCase();
    let items = document.querySelectorAll('#itemList li');

    items.forEach(item => {
      item.style.display =
item.textContent.toLowerCase().includes(searchValue) ? 'list-item' :
'none';
    });
  });
</script>
```

25. Change Font Size

html

```
<button id="increaseFont">Increase Font Size</button>
<button id="decreaseFont">Decrease Font Size</button>
<p id="text">This is some text.</p>
```

```
<script>
  let fontSize = 16;
  const text = document.getElementById('text');

  document.getElementById('increaseFont').addEventListener('click', ()
=> {
    fontSize += 2;
    text.style.fontSize = `${fontSize}px`;
  });
```

```

    document.getElementById('decreaseFont').addEventListener('click', ()
=> {
    fontSize -= 2;
    text.style.fontSize = `${fontSize}px`;
    });
</script>

```

26. Dynamic Table Rows

html

```

<button id="addRow">Add Row</button>
<table id="dynamicTable" border="1">
  <tr>
    <th>Name</th>
    <th>Age</th>
  </tr>
</table>

<script>
  document.getElementById('addRow').addEventListener('click', () => {
    let table = document.getElementById('dynamicTable');
    let row = table.insertRow();
    let cell1 = row.insertCell(0);
    let cell2 = row.insertCell(1);
    cell1.textContent = 'John Doe';
    cell2.textContent = '30';
  });
</script>

```

27. Timer Countdown with Start and Stop

html

```

<button id="startTimer">Start Timer</button>
<button id="stopTimer">Stop Timer</button>
<p id="timerDisplay">Time: 0</p>

<script>
  let timer;
  let seconds = 0;

  document.getElementById('startTimer').addEventListener('click', () => {
    timer = setInterval(() => {
      seconds++;
    }, 1000);
  });

```

```

        document.getElementById('timerDisplay').textContent = `Time: $
{seconds}`;
    }, 1000);
});

document.getElementById('stopTimer').addEventListener('click', () => {
    clearInterval(timer);
});
</script>

```

28. Character Limit

html

```

<textarea id="textArea" rows="4" cols="50" maxlength="150"
placeholder="Type here..."></textarea>
<p id="charLimit">Character limit: 150</p>

```

```

<script>
    document.getElementById('textArea').addEventListener('input', () => {
        let textArea = document.getElementById('textArea');
        let charLimit = document.getElementById('charLimit');
        charLimit.textContent = `Character limit: ${150 - textArea.value.length}
`;
    });
</script>

```

29. Change Image Source

html

```


<button id="changeImage">Change Image</button>

```

```

<script>
    document.getElementById('changeImage').addEventListener('click', ()
=> {
        document.getElementById('image').src = 'image2.jpg';
    });
</script>

```

30. Drag and Drop File Upload

html

```
<div id="dropArea" style="width: 200px; height: 200px; border: 2px
dashed #ccc; text-align: center; line-height: 200px;">Drop files here</
div>
<p id="fileName"></p>
```

```
<script>
  let dropArea = document.getElementById('dropArea');

  dropArea.addEventListener('dragover', (event) => {
    event.preventDefault();
    dropArea.style.backgroundColor = '#e0e0e0';
  });

  dropArea.addEventListener('dragleave', () => {
    dropArea.style.backgroundColor = '#fff';
  });

  dropArea.addEventListener('drop', (event) => {
    event.preventDefault();
    dropArea.style.backgroundColor = '#fff';

    let files = event.dataTransfer.files;
    if (files.length > 0) {
      document.getElementById('fileName').textContent = `File name: $
{files[0].name}`;
    }
  });
</script>
```

31. Form Field Highlight on Focus

html

```
<input type="text" id="highlightField" placeholder="Focus on me">
```

```
<style>
  .highlight {
    border: 2px solid blue;
  }
</style>
```

```
<script>
  document.getElementById('highlightField').addEventListener('focus', ()
=> {
```



```

    document.getElementById('highlightField').classList.add('highlight');
  });

  document.getElementById('highlightField').addEventListener('blur', ()
=> {

document.getElementById('highlightField').classList.remove('highlight');
  });
</script>

```

32. Real-Time Currency Conversion

html

```

<input type="number" id="amount" placeholder="Amount in USD">
<p id="convertedAmount">Amount in EUR: </p>

```

```

<script>
  document.getElementById('amount').addEventListener('input', () => {
    let amount = parseFloat(document.getElementById('amount').value);
    let conversionRate = 0.85; // Example conversion rate
    let convertedAmount = amount * conversionRate;
    document.getElementById('convertedAmount').textContent = `Amount
in EUR: ${convertedAmount.toFixed(2)}`;
  });
</script>

```

33. Text Color Change on Hover

html

```

<p id="hoverText" style="cursor: pointer;">Hover over me!</p>

```

```

<script>
  document.getElementById('hoverText').addEventListener('mouseover',
  () => {
    document.getElementById('hoverText').style.color = 'red';
  });

  document.getElementById('hoverText').addEventListener('mouseout', ()
=> {
    document.getElementById('hoverText').style.color = 'black';
  });
</script>

```

34. Toggle Class on Click

html

```
<button id="toggleClassButton">Toggle Class</button>
<div id="box" style="width: 100px; height: 100px; background: blue;"></div>
```

```
<style>
  .red {
    background: red;
  }
</style>
```

```
<script>
```

```
document.getElementById('toggleClassButton').addEventListener('click',
() => {
  document.getElementById('box').classList.toggle('red');
});
```

```
</script>
```

35. Element Resizing

html

```
<div id="resizeElement" style="width: 100px; height: 100px; background:
green; resize: both; overflow: auto;"></div>
```

```
<script>
  document.getElementById('resizeElement').addEventListener('resize',
() => {
    console.log('Element resized');
  });
```

```
</script>
```

36. Form Data Serialization

html

```
<form id="serializeForm">
  <input type="text" name="username" placeholder="Username">
  <input type="password" name="password" placeholder="Password">
  <input type="submit" value="Serialize">
</form>
```

```
<p id="serializedData"></p>
```

```
<script>
  document.getElementById('serializeForm').addEventListener('submit',
(event) => {
    event.preventDefault();
    let formData = new FormData(event.target);
    let serialized = Array.from(formData.entries()).map(([key, value]) => `${
{key}: ${value}}`).join(', ');
    document.getElementById('serializedData').textContent = `Serialized
Data: ${serialized}`;
  });
</script>
```

37. Auto-Complete Input

html

```
<input type="text" id="autoComplete" placeholder="Start typing...">
<ul id="suggestions" style="display: none; border: 1px solid #ccc;"></ul>
```

```
<script>
const suggestions = ['Apple', 'Banana', 'Cherry', 'Date', 'Fig', 'Grape'];
const input = document.getElementById('autoComplete');
const suggestionList = document.getElementById('suggestions');

input.addEventListener('input', () => {
  let value = input.value.toLowerCase();
  suggestionList.innerHTML = "";
  if (value) {
    suggestions.forEach(item => {
      if (item.toLowerCase().startsWith(value)) {
        let li = document.createElement('li');
        li.textContent = item;
        li.addEventListener('click', () => {
          input.value = item;
          suggestionList.style.display = 'none';
        });
        suggestionList.appendChild(li);
      }
    });
    suggestionList.style.display = suggestionList.children.length > 0 ?
'block' : 'none';
  } else {

```

```

        suggestionList.style.display = 'none';
    }
});
</script>

```

38. Image Gallery Lightbox

html

```


<div id="lightbox" style="display: none; position: fixed; top: 50%; left:
50%; transform: translate(-50%, -50%); padding: 20px; background:
white; border: 1px solid black;">
    
    <button id="closeLightbox">Close</button>
</div>

<script>
    document.getElementById('galleryImage').addEventListener('click', ()
=> {
        let lightbox = document.getElementById('lightbox');
        let lightboxImage = document.getElementById('lightboxImage');
        lightboxImage.src = document.getElementById('galleryImage').src;
        lightbox.style.display = 'block';
    });

    document.getElementById('closeLightbox').addEventListener('click', ()
=> {
        document.getElementById('lightbox').style.display = 'none';
    });
</script>

```

39. Custom Tooltip

html

```

<button id="tooltipButton">Hover for Tooltip</button>
<div id="tooltip" style="display: none; position: absolute; background:
black; color: white; padding: 5px;">This is a tooltip</div>

<script>
    let button = document.getElementById('tooltipButton');
    let tooltip = document.getElementById('tooltip');

```

```

button.addEventListener('mouseover', (event) => {
  tooltip.style.display = 'block';
  tooltip.style.left = `${event.pageX + 10}px`;
  tooltip.style.top = `${event.pageY + 10}px`;
});

button.addEventListener('mouseout', () => {
  tooltip.style.display = 'none';
});
</script>

```

40. Dynamic Table Sorting

html

```

<table id="sortableTable" border="1">
  <thead>
    <tr>
      <th>Name</th>
      <th>Age</th>
    </tr>
  </thead>
  <tbody>
    <tr><td>John</td><td>30</td></tr>
    <tr><td>Jane</td><td>25</td></tr>
    <tr><td>Doe</td><td>40</td></tr>
  </tbody>
</table>
<button id="sortTable">Sort by Age</button>

<script>
  document.getElementById('sortTable').addEventListener('click', () => {
    let rows = Array.from(document.querySelectorAll('#sortableTable
tbody tr'));
    rows.sort((a, b) => {
      return parseInt(a.cells[1].textContent) -
      parseInt(b.cells[1].textContent);
    });
    document.querySelector('#sortableTable tbody').append(...rows);
  });
</script>

```

