<<<[Prepared by : "Ismail Shah"]>>>

Mcqs of chap 21 - 50 from the book "the smarter way to learn javascript" :

1. What is the correct syntax for a for loop in JavaScript?

A) for (i = 0; i <= 10; i++)
B) for i = 0; i <= 10; i++
C) for i (0; 10; i++)
D) loop (i = 0; i < 10; i++)

Answer: A
Explanation: A is the correct format. JavaScript for loops follow the syntax: for (initialization; condition; increment).

---

2. What will the following code output?

```
for (let i = 0; i < 3; i++) {
  console.log(i);
}
```

A) 1 2 3
B) 0 1 2
C) 0 1 2 3
D) Nothing

Answer: B
Explanation: The loop starts at i = 0 and runs while i < 3. So it prints 0, 1, and 2.

---

3. Which part of the for loop updates the loop variable?

A) Initialization
B) Condition
C) Increment
D) Body

Answer: C
Explanation: The increment part updates the loop variable after each iteration.

---

4. What does this loop do?

```
for (let i = 5; i > 0; i--) {
  console.log(i);
}
```

A) Counts up from 1 to 5
B) Logs 5 to 1
C) Logs 0 to 5
D) Logs 1 to 4

Answer: B
Explanation: It starts at 5 and decreases i until it is greater than 0.

---

5. How many times will this loop run?

```
for (let i = 0; i < 10; i += 2) {
  console.log(i);
}
```

A) 10
B) 5
C) 2
D) Infinite

Answer: B
Explanation: The loop increases by 2 each time, so it runs 5 times: 0, 2, 4, 6, 8.

---

6. What is a good use of a flag variable in a for loop?

A) To store the last loop value
B) To track a condition being met

C) To add two numbers
D) To break out of the loop immediately

Answer: B
Explanation: A flag is often a Boolean used to mark if a certain condition was met during the loop.

---

7. What will be the value of found after this loop runs?

```
let nums = [1, 2, 3, 4];
let found = false;
for (let i = 0; i < nums.length; i++) {
  if (nums[i] === 3) {
    found = true;
  }
}
```

A) true
B) false
C) undefined
D) 3

Answer: A
Explanation: Since 3 is in the array, found becomes true when nums[i] === 3.

---

8. What happens in a nested loop?

A) Loops run at the same time
B) One loop is placed inside another
C) Loops cancel each other
D) Only the outer loop runs

Answer: B
Explanation: In a nested loop, the inner loop runs completely every time the outer loop runs once.

---

9. What is the output of this code?

```
for (let i = 1; i <= 2; i++) {
  for (let j = 1; j <= 2; j++) {
    console.log(i * j);
  }
}
```

A) 1 2 2 4
B) 1 1 2 2
C) 1 2 3 4
D) 1 2 3 4 5

Answer: A
Explanation: Output is the product of i and j: 1×1, 1×2, 2×1, 2×2.

---

10. What's a potential risk of nested loops?

A) Too few iterations
B) Slow performance for large data
C) Not supported in modern JavaScript
D) They never end

Answer: B
Explanation: Nested loops multiply the number of iterations, which can slow down performance for large datasets.

11. What is the output of this code?

```
let str = "JavaScript";
console.log(str.toLowerCase());
```

A) JavaScript
B) javascript
C) JAVASCRIPT
D) Error

Answer: B
Explanation: .toLowerCase() converts all characters in the string to lowercase.

---

12. Which method returns the number of characters in a string?

A) count()
B) size()
C) length
D) getLength()

Answer: C
Explanation: The .length property gives the total number of characters in a string.

---

13. What is the result of this expression?

"Hello".charAt(1);

A) "H"
B) "e"
C) "l"
D) "o"

Answer: B
Explanation: Indexing starts at 0, so index 1 is "e".

---

14. How do you extract "Script" from the string "JavaScript"?

A) slice(4)
B) slice(0,6)
C) slice(4,10)
D) slice(5,10)

Answer: C
Explanation: slice(4,10) extracts characters from index 4 up to (but not including) 10.

---

15. What does this code return?

```
let str = "apple,banana,grape";
let parts = str.split(",");
console.log(parts[1]);
```

A) apple
B) banana
C) grape
D) banana,grape

Answer: B
Explanation: The string is split into an array, and index 1 is "banana".

---

16. What will str.indexOf("a") return for this code?

```
let str = "banana";
```

A) 0
B) 1
C) 2
D) 3

Answer: A
Explanation: indexOf() returns the index of the first occurrence, which is position 1 (zero-based index).

---

17. How do you replace "cats" with "dogs" in a string?

A) replace("dogs", "cats")
B) str.replace("cats", "dogs")
C) str.switch("cats", "dogs")
D) str.change("cats", "dogs")

Answer: B
Explanation: replace() replaces the first instance of a match with the new string.

---

18. What will this return?

```
let str = "hello world";
console.log(str.replace("world", "there"));
```

A) hello
B) hello world
C) hello there
D) Error

Answer: C
Explanation: replace() finds "world" and replaces it with "there".

---

19. What does this return?

```
let str = "abcdef";
console.log(str.slice(2, 4));
```

A) "bc"
B) "cd"
C) "de"
D) "ef"

Answer: B
Explanation: slice(2, 4) returns characters from index 2 to 3 (not including 4), which are "cd".

---

20. Which method checks if a string includes a certain word?

A) has()
B) contains()
C) includes()
D) find()

Answer: C
Explanation: The includes() method checks if the string contains the specified value and returns true or false.

21. What does Math.round(4.7) return?

A) 4
B) 5
C) 4.7
D) 6

Answer: B
Explanation: Math.round() rounds to the nearest integer. 4.7 becomes 5.

---

22. What will be the result of Math.floor(7.9)?

A) 8
B) 7
C) 9
D) Error

Answer: B
Explanation: Math.floor() always rounds down to the nearest integer.

---

23. What is the purpose of Math.ceil()?

A) Rounds to nearest whole number
B) Rounds down
C) Rounds up
D) Returns random number

Answer: C
Explanation: Math.ceil() rounds a number up to the next largest integer.

---

24. What does Math.random() return?

A) A number from 1 to 10

B) A decimal between 0 and 1
C) An integer from 0 to 100
D) Always 0

Answer: B
Explanation: Math.random() returns a floating-point number between 0 (inclusive) and 1 (exclusive).

---

25. What is the output of this code?

console.log(Math.floor(Math.random() * 5));

A) Always 5
B) Any decimal between 0 and 5
C) An integer between 0 and 4
D) An integer between 1 and 5

Answer: C
Explanation: Math.random() generates 0–0.999..., multiplied by 5 becomes 0–4.999..., and Math.floor() rounds it down to 0–4.

---

26. How do you convert a string "42" to a number?

A) Number("42")
B) parseInt("42")
C) +"42"
D) All of the above

Answer: D
Explanation: All options will convert a numeric string to a number in JavaScript.

---

27. What will this return?

parseInt("42.7");

A) 42.7
B) 43
C) 42
D) NaN

Answer: C
Explanation: parseInt() converts only the whole number part, so 42.7 becomes 42.

---

28. What does parseFloat("3.14") return?

A) "3.14"
B) 3
C) 3.14
D) NaN

Answer: C
Explanation: parseFloat() reads and returns the number including decimal points.

29. What will this output be?

```
let num = 7.45678;
console.log(num.toFixed(2));
```

A) 7
B) 7.45
C) 7.46
D) 7.45678

Answer: C
Explanation: toFixed(2) rounds the number to two decimal places. 7.45678 becomes 7.46.

30. Which statement turns a number into a string?

A) String(123)
B) 123.toString()
C) "" + 123
D) All of the above

Answer: D
Explanation: All three are valid ways to convert a number to a string in JavaScript.

31. What will Math.round(2.5) return?

A) 2
B) 3
C) 2.5
D) 4

Answer: B
Explanation: Math.round() rounds to the nearest integer. 2.5 rounds up to 3.

---

32. What's the difference between Math.floor(2.9) and Math.ceil(2.1)?

A) Both return 2
B) floor gives 2, ceil gives 3
C) floor gives 3, ceil gives 2
D) They both return 3

Answer: B
Explanation: floor() always rounds down, ceil() always rounds up.

---

33. What is the result of this code?

Math.floor(-3.5);

A) -3
B) -4
C) 3
D) 4

Answer: B
Explanation: Math.floor() goes to the lower integer, even for negatives. So it becomes -4.

---

34. Which method would you use to get the absolute value of a number?

A) Math.abs()
B) Math.absolute()
C) Math.pos()
D) Math.unsigned()

Answer: A
Explanation: Math.abs() returns the non-negative version of any number.

---

35. What does Math.pow(2, 3) return?

A) 6
B) 8
C) 9
D) 4

Answer: B
Explanation: Math.pow(x, y) returns x raised to the power of y. So $2^3 = 8$.

---

36. What is the value of Math.max(3, 7, 2)?

A) 3
B) 2
C) 7
D) Error

Answer: C
Explanation: Math.max() returns the largest of the numbers provided.

---

37. What will be printed?

```
let x = "9.81";
console.log(parseInt(x));
```

A) 9.81
B) 10
C) 9
D) NaN

Answer: C
Explanation: parseInt() only reads the whole number part before the decimal.

---

38. Which method returns a floating-point number from a string like "7.45"?

A) parseFloat("7.45")
B) parseInt("7.45")
C) Number("7.45")
D) A and C

Answer: D
Explanation: Both parseFloat() and Number() return 7.45 as a float.

---

39. What will this return?

```
let str = "abc";
console.log(Number(str));
```

A) abc
B) 0
C) undefined
D) NaN

Answer: D
Explanation: Number("abc") fails to convert and returns NaN (Not a Number).

---

40. Which of the following converts 123 to "123"?

A) String(123)
B) 123.toString()

C) "" + 123
D) All of the above

Answer: D
Explanation: Each option turns a number into a string.

---

41. What will typeof parseInt("123abc") return?

A) "string"
B) "number"
C) "NaN"
D) "object"

Answer: B
Explanation: Even though the input has non-numeric characters, parseInt("123abc") returns 123, which is a number.

---

42. What's the result of parseFloat("123.45xyz")?

A) 123.45
B) NaN
C) 0
D) Error

Answer: A
Explanation: parseFloat() reads up to the first invalid character and stops. "123.45xyz" returns 123.45.

---

43. What will be the output?

```
let a = 5.6789;
console.log(a.toFixed(3));
```

A) "5.679"
B) 5.679

C) "5.678"
D) 5.6789

Answer: A
Explanation: toFixed(3) returns the number rounded to 3 decimal places as a string.

---

44. Which method would you use to ensure a number only has 2 decimal places in calculations?

A) Math.ceil()
B) toFixed(2)
C) round(2)
D) truncate()

Answer: B
Explanation: toFixed(2) ensures the result is rounded and formatted to 2 decimal places.

45. What does new Date() return in JavaScript?

A) A number representing time
B) A string of today's date
C) A Date object with the current date and time
D) The date only, without time

Answer: C
Explanation: new Date() creates a Date object representing the current date and time.

---

46. Which method retrieves the current year from a Date object?

let today = new Date();

A) today.year()
B) today.getFullYear()
C) today.getYear()
D) today.getCurrentYear()

Answer: B

Explanation: getFullYear() returns the 4-digit year from a Date object.

---

47. What will this output?

```
let d = new Date("2025-01-01");
console.log(d.getMonth());
```

A) 1
B) 12
C) 0
D) January

Answer: C
Explanation: In JavaScript, months are zero-indexed. So January is 0.

---

48. What does the following function do?

```
function greet() {
  console.log("Hello!");
}
```

A) Logs Hello immediately
B) Returns a string
C) Defines a reusable block of code
D) Declares a variable

Answer: C
Explanation: A function defines reusable code that runs when called.

---

49. What is the correct way to call the above greet function?

A) greet;
B) call greet();
C) greet();
D) function greet()

Answer: C
Explanation: Functions are invoked by writing the name followed by parentheses.

---

50. What will this output?

```
function add(x, y) {
  return x + y;
}
console.log(add(3, 4));
```

A) 7
B) 34
C) undefined
D) "3 + 4"

Answer: A
Explanation: add(3, 4) adds the two arguments and returns 7.

---

51. Which keyword is used to send a result out of a function?

A) send
B) output
C) return
D) pass

Answer: C
Explanation: The return statement sends the result back to where the function was called.

---

52. What is the role of default in a switch statement?

A) It stops the switch
B) It handles unmatched cases
C) It defines a constant
D) It breaks the loop

Answer: B
Explanation: The default case runs if none of the case conditions are met.

---

53. What does this switch block return?

```
let color = "green";
switch (color) {
  case "red":
    console.log("Stop");
    break;
  case "green":
    console.log("Go");
    break;
  default:
    console.log("Caution");
}
```

A) Stop
B) Go
C) Caution
D) Error

Answer: B
Explanation: Since color is "green", the green case is matched and "Go" is printed.

---

54. What will happen without break in a switch case?

A) It breaks the browser
B) Only the matched case runs
C) All cases run after the match
D) It stops the switch before matching

Answer: C
Explanation: Without break, JavaScript will execute the matched case and all cases that follow (“fall through”).

55. What will this output?

let now = new Date();
console.log(now.getDay());

A) Name of the day (e.g., "Monday")
B) Current date
C) A number from 0 to 6
D) An error

Answer: C
Explanation: getDay() returns the day of the week as a number (0 = Sunday, 6 = Saturday).

---

56. Which method gets the current hour from a Date object?

A) getHour()
B) getHours()
C) currentHour()
D) now.getHours()

Answer: B
Explanation: getHours() retrieves the hour (0–23) from a Date object.

---

57. What will this return?

let birthday = new Date("1995-06-15");
console.log(birthday.getFullYear());

A) 1995
B) 95
C) 06
D) Error

Answer: A
Explanation: getFullYear() returns the full four-digit year.

---

58. What does getTime() return?

A) The time as a string
B) The number of seconds since 1970
C) The number of milliseconds since Jan 1, 1970
D) A time object

Answer: C
Explanation: getTime() returns milliseconds since the Unix epoch (Jan 1, 1970).

---

59. Which function definition is correct?

A) function = myFunc() {}
B) def myFunc() {}
C) function myFunc() {}
D) func myFunc() {}

Answer: C
Explanation: The correct JavaScript syntax is function functionName() {}.

---

60. What will this output?

```
function multiply(a, b) {
  console.log(a * b);
}
multiply(2, 3);
```

A) 5
B) 6
C) a * b
D) undefined

Answer: B
Explanation: The function logs the product of a and b, which is 6.

---

61. What will this return?

```
function sayHello() {
  return "Hello!";
}
console.log(sayHello());
```

A) Hello!
B) undefined
C) Function reference
D) Error

Answer: A
Explanation: The function returns the string "Hello!" and console.log prints it.

---

62. If a function doesn't have a return statement, what does it return by default?

A) 0
B) null
C) undefined
D) ""

Answer: C
Explanation: A function without a return statement returns undefined.

---

63. What's the output?

```
function greet(name = "Guest") {
  return "Hello, " + name;
}
console.log(greet());
```

A) Hello,
B) Hello, Guest
C) Guest
D) undefined

Answer: B

Explanation: Default parameter "Guest" is used when no argument is passed.

---

64. Which is a valid function expression?

A) function sayHi() {}
B) let greet = function() {}
C) const fun = function greet() {}
D) Both B and C

Answer: D
Explanation: Both are function expressions. A is a function declaration.

---

65. What's the difference between return and console.log() in a function?

A) return displays to screen, log ends function
B) return exits function and gives result back; console.log() just prints
C) No difference
D) return prints and logs to console

Answer: B
Explanation: return sends back a value to where the function was called. console.log() just outputs to the console.

---

66. What is function hoisting?

A) Functions can't be moved
B) Function expressions are moved to top
C) Function declarations can be used before they're defined
D) Only arrow functions are hoisted

Answer: C
Explanation: Function declarations are hoisted, meaning they can be used before they appear in code.

---

67. What will this output?

```
let result = function(x, y) {
  return x - y;
};
console.log(result(10, 3));
```

A) 13
B) 7
C) -7
D) undefined

Answer: B
Explanation: Function expression subtracts y from x, so 10 - 3 = 7.

---

68. Which function type is not hoisted?

A) Function declarations
B) Function expressions
C) Named functions
D) All of the above

Answer: B
Explanation: Function expressions are not hoisted. You must define them before use.

---

69. What is the output?

```
function testScope() {
  let x = 5;
}
console.log(x);
```

A) 5
B) undefined
C) ReferenceError
D) null

Answer: C
Explanation: x is declared inside the function and not accessible outside due to block scope.

70. What is the key difference between a while loop and a do...while loop?

A) do...while executes only if the condition is true
B) while always executes once
C) do...while always runs at least once
D) No difference

Answer: C
Explanation: do...while executes the loop once before checking the condition.

---

71. What is the output of this loop?

```
let i = 0;
while (i < 3) {
  console.log(i);
  i++;
}
```

A) 0 1 2
B) 1 2 3
C) 0 1 2 3
D) 1 2

Answer: A
Explanation: It prints values of i from 0 to 2 while i < 3.

---

72. How many times will this run?

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

A) 0
B) 5
C) 6
D) Infinite loop

Answer: C
Explanation: The do...while runs once before the condition is checked. So i becomes 6.

---

73. What's the output?

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

A) 0
B) 1
C) false
D) undefined

Answer: A
Explanation: The loop never runs because the condition is false initially.

---

74. Which loop guarantees at least one execution?

A) for
B) while
C) do...while
D) All of the above

Answer: C
Explanation: do...while always runs once, even if the condition is false.

---

75. What's the correct way to declare an array in JavaScript?

A) let arr = "apple", "banana";
B) let arr = ["apple", "banana"];
C) let arr = {apple, banana};
D) let arr = (apple, banana);

Answer: B
Explanation: Arrays use square brackets with comma-separated values.

---

76. What is the output?

```
let fruits = ["apple", "banana"];
console.log(fruits[1]);
```

A) apple
B) banana
C) 1
D) undefined

Answer: B
Explanation: Array indexing starts at 0. fruits[1] is "banana".

---

77. What will this print?

```
let numbers = [1, 2, 3];
numbers.push(4);
console.log(numbers);
```

A) [1, 2, 3]
B) [4, 1, 2, 3]
C) [1, 2, 3, 4]
D) 4

Answer: C
Explanation: push() adds a value to the end of the array.

---

78. What does pop() do in an array?

A) Removes first element
B) Removes last element
C) Adds to start
D) Adds to end

Answer: B
Explanation: pop() removes and returns the last item in an array.

---

79. What's the result of this code?

```
let arr = [10, 20, 30];
console.log(arr.length);
```

A) 2
B) 3
C) 30
D) undefined

Answer: B
Explanation: length returns the total number of elements in the array.

80. What will be the output of this code?

```
let i = 1;
while (i <= 3) {
  console.log("Hi");
  i++;
}
```

A) Hi
B) Hi Hi
C) Hi Hi Hi
D) Infinite loop

Answer: C
Explanation: Loop runs three times because the condition i <= 3 is true for i = 1, 2, and 3.

---

81. Which of these is the correct syntax for a do...while loop?

A) do (code) while (condition);
B) do { code } while (condition);
C) while { code } do (condition);
D) do while (code) { condition };

Answer: B
Explanation: Proper syntax uses do {} followed by while(condition);.

---

82. What will this code output?

```
let i = 0;
do {
  console.log(i);
  i++;
} while (i < 2);
```

A) 0 1
B) 1 2
C) 0 1 2
D) Nothing

Answer: A
Explanation: Loop runs while i < 2, printing 0 and 1.

---

83. What is a potential risk when using while loops?

A) Too many print statements
B) Variables not declared
C) Forgetting to update the condition
D) Syntax errors

Answer: C

Explanation: If the condition is never updated, it can cause an infinite loop.

---

84. What's the output?

```
let i = 2;
while (i > 0) {
  console.log(i);
  i--;
}
```

A) 2
B) 2 1
C) 2 1 0
D) 1 0

Answer: B
Explanation: The loop prints 2 and 1, then stops when i is 0.

Here are 25 MCQs covering the specified chapters from "The Smarter Way to Learn JavaScript":

85. What is the primary purpose of a while loop?

a) To execute a block of code repeatedly
b) To skip a block of code
c) To exit a program
d) To declare a variable

Answer: a) To execute a block of code repeatedly
Explanation: A while loop allows you to execute a block of code repeatedly while a certain condition is true.

86. What is the syntax for a while loop?

a) while (condition) { code }
b) if (condition) { code }
c) for (var i = 0; i < 10; i++) { code }
d) do { code } while (condition)

Answer: a) while (condition) { code }

Explanation: The syntax for a while loop includes the while keyword, a condition in parentheses, and a block of code in curly brackets.

87. What happens if the condition in a while loop is false?

a) The loop will execute indefinitely
b) The loop will skip the code block
c) The loop will throw an error
d) The loop will execute once

Answer: b) The loop will skip the code block
Explanation: If the condition in a while loop is false, the code block will be skipped.

88. Can a while loop be used to iterate over an array?

a) Yes
b) No

Answer: a) Yes
Explanation: A while loop can be used to iterate over an array by using a counter variable and checking the length of the array.

89. What is an example of a while loop?

a) while (i < 10) { console.log(i); i++; }
b) for(var i = 0;i < 10; i++) { console.log(i); }
c) if (i < 10) { console.log(i); }
d) do { console.log(i); } while (i < 10)

Answer: a) while (i < 10) { console.log(i); i++; }
Explanation: This is an example of a while loop that logs the numbers 0 to 9 to the console.

90. What is the primary difference between a while loop and a do...while loop?

a) The condition is checked before the code block in a while loop
b) The condition is checked after the code block in a do...while loop
c) The code block is executed only once in a do...while loop
d) The code block is skipped in a while loop

Answer: b) The condition is checked after the code block in a do...while loop
Explanation: In a do...while loop, the condition is checked after the code block, whereas in a while loop, the condition is checked before the code block.

91. What is the syntax for a do...while loop?

a) do { code } while (condition)
b) while (condition) { code }
c) for (var i = 0; i < 10; i++) { code }
d) if (condition) { code }

Answer: a) do { code } while (condition)
Explanation: The syntax for a do...while loop includes the do keyword, a code block in curly brackets, and a condition in parentheses.

92. What happens if the condition in a do...while loop is false?

a) The loop will execute indefinitely
b) The loop will skip the code block
c) The loop will throw an error
d) The loop will execute once

Answer: d) The loop will execute once
Explanation: If the condition in a do...while loop is false, the code block will still be executed once.

93. Can a do...while loop be used to iterate over an array?

a) Yes
b) No

Answer: a) Yes
Explanation: A do...while loop can be used to iterate over an array by using a counter variable and checking the length of the array.

94. What is an example of a do...while loop?

a) do { console.log(i); i++; } while (i < 10)
b) while (i < 10) { console.log(i); i++; }
c) for(var i = 0; i < 10; i++){ console.log(i); }
d) if (i < 10) { console.log(i); }

Answer: a) do { console.log(i); i++; } while (i < 10)
Explanation: This is an example of a do...while loop that logs the numbers 0 to 9 to the console.

95. Where can JavaScript code be placed in an HTML document?

a) In the head section
b) In the body section
c) In an external file
d) All

Answer: b)

96. What is the purpose of commenting in JavaScript code?

a) To add functionality to the code
b) To explain the code to other developers
c) To debug the code
d) To optimize the code

Answer: b) To explain the code to other developers
Explanation: Commenting helps other developers understand the code, making it easier to maintain and collaborate.

97. What are the two types of comments in JavaScript?

a) Single-line and multi-line
b) Block and inline
c) HTML and CSS
d) Java and Python

Answer: a) Single-line and multi-line
Explanation: JavaScript supports single-line comments (//) and multi-line comments (/* */).

98. How do you write a 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

Answer: a) // This is a comment
Explanation: Single-line comments start with two forward slashes (//).

99. What is an event in JavaScript?

a) A function that is called repeatedly
b) A variable that is declared globally
c) A action that occurs when a user interacts with a web page

d) A loop that iterates over an array

Answer: c) A action that occurs when a user interacts with a web page
Explanation: Events are triggered by user interactions, such as clicking a button or hovering over an element.

100. What is the purpose of the addEventListener method?

a) To remove an event listener
b) To add an event listener to an element
c) To trigger an event
d) To cancel an event

Answer: b) To add an event listener to an element
Explanation: The addEventListener method attaches an event listener to an element, allowing the code to respond to events.

101. What is the difference between a link event and a button event?

a) A link event is triggered by clicking a link, while a button event is triggered by clicking a button
b) A link event is triggered by hovering over a link, while a button event is triggered by clicking a button
c) A link event is triggered by submitting a form, while a button event is triggered by clicking a button
d) A link event is triggered by loading a page, while a button event is triggered by clicking a button

Answer: a) A link event is triggered by clicking a link, while a button event is triggered by clicking a button
Explanation: Link events and button events are triggered by different user interactions.

102. What is the purpose of the mouse event?

a) To track the user's keyboard input
b) To track the user's mouse movements
c) To track the user's touch input
d) To track the user's scroll position

Answer: b) To track the user's mouse movements
Explanation: Mouse events allow the code to respond to user interactions with the mouse.

103. How do you read the value of a form field using JavaScript?

a) Using the value property
b) Using the text property
c) Using the innerHTML property
d) Using the outerHTML property

Answer: a) Using the value property
Explanation: The value property returns the current value of a form field.

104. How do you set the value of a form field using JavaScript?

a) Using the value property
b) Using the text property
c) Using the innerHTML property
d) Using the outerHTML property

Answer: a) Using the value property
Explanation: The value property can be used to set the value of a form field.

105. What is the difference between reading and setting a field value?

a) Reading a field value retrieves the current value, while setting a field value changes the current value
b) Reading a field value changes the current value, while setting a field value retrieves the current value
c) Reading a field value is used for text fields, while setting a field value is used for checkbox fields
d) Reading a field value is used for checkbox fields, while setting a field value is used for text fields

Answer: a) Reading a field value retrieves the current value, while setting a field value changes the current value
Explanation: Reading a field value retrieves the current value, while setting a field value changes the current value.

<<<[Prepared by : "Ismail Shah"]>>>