JavaScript output based test – 1.1

March-24/ JS/001.1 Time: 02:00hrs

JavaScript

Diploma in Advance Computing

March 2024

***What will be the output of the following code snippet?***

1. let x = (1, 2, 3, 4);

console.log(x);

1. let x = [1, 2, 3, 4];

console.log(...x);

1. let [x, y] = [1, 2, 3, 4];

console.log(x, y);

1. let [x, y, ...z] = [1, 2, 3, 4];

console.log(x, y, z);

1. let x = [11, 12];

console.log(x.length);

x[7] = 'saleel';

1. let x = [11, 12]

x[7] = 'saleel'

console.log(x.length);

1. let x = [11, 12]

x[-1] = 'saleel'

console.log(x[-1]);

1. let x = [11, 12];

x[7] = 'saleel';

console.log(x);

1. let [a, b] = [10, 20];

console.log(a, b);

1. var [a, b] = [10, 20];

var [b, a] = [a, b]

console.log(a, b);

1. let x = [1, 2, 3]

let y1 = x;

let y2 = [...x]

x[0] = 200;

console.log(y1);

console.log(y2);

1. let x = [1, 2, 3]

let y = [10, 20, ...x];

console.log(y);

1. let x = [1, 2, 3]

let y = [0, [...x],4, 5,6];

console.log(y);

1. let x = [1, 2, 3, 4, 5, 6];

let [a, b] = [...x];

console.log(a, b);

1. let x = [1, 2, 3, 4, 5, 6];

let [a, b, c] = [...x];

console.log(a, b, c);

1. let x = [1, 2, 3, 4, 5, 6];

let [a, b, ...c] = [...x];

console.log(a, b, c);

1. function fn(x) {

console.log(x);

}

fn("One", "Two", "Three");

1. let y = ("One", "Two", "Three")

function fn(x) {

console.log(x);

}

fn(y);

1. let y = ["One", "Two", "Three"]

function fn(x) {

console.log(x);

}

fn(y);

1. let y = new Set(["One", "Two", "Three"]);

function fn(x) {

console.log(x);

}

fn(y);

1. function fn(...x) {

console.log(x);

}

fn(1, 2, 3);

1. function fn(...x) {

console.log(x);

}

fn([1, 2, 3]);

1. let x = [..."saleel"];

for (const doc of x) {

console.log(doc);

};

1. function fn(a, b, ...c) {

console.log(a, b, c);

}

fn(1, 2, 3, 4, 5, 6);

1. function fn1(a, b) {

console.log(a, b);

}

fn1(...[1, 2, 3, 4, 5, 6]);

1. var num = 10;

function increment() {

num++;

}

console.log(num);

increment();

1. var num = 10;

function increment() {

num++;

}

increment();

console.log(num);

1. console.log(...'saleel');
2. const [ a, , b ] = [1, 2, 3, 4, 5];

console.log(a, b);

1. const [a, , , b, ...c] = [1, 2, 3, 4, 5];

console.log(a, b, c);

1. const obj = {"\_id": 1001, "ename": "saleel", "city": "baroda"}

const { \_id, ename, city} = obj;

console.log(\_id, ename, city);

1. const obj = { "\_id": 1001, "ename": "saleel", "city": "baroda" }

const { \_id, ename, city, state = "GJ" } = obj;

console.log(\_id, ename, city, state);

1. const obj = { "\_id": 1001, "ename": "saleel", "city": "baroda" }

const { \_id, state = "GJ", ...r } = obj;

console.log(\_id, r, state);

1. const x = [1, 2, 3, 4, 5];

const [y, z] = x;

console.log(y);

console.log(z);

1. try {

const x = [1, 2, 3, 4, 5];

const [a, ...b, c] = [...x];

console.log(a + " " + b + " " + c);

}

catch (error) {

console.log('Rest element must be last element')

}

1. const obj = { a: "saleel", b: "sharmin" };

obj["c"] = "ruhan";

const { a, b, c } = obj;

console.log({ x: a, y: b, z: c });

console.log('This is an error');

1. const person = {

\_id: 1,

myName: 'saleel',

myCity: 'baroda',

myState:'GJ'

}

console.log(JSON.stringify(person));

1. console.log(JSON.stringify (

[ new String("Saleel"),

new Boolean(false),

new Number(9850884228) ])

);

1. console.log({ "myNameIs": JSON.stringify("saleel") });
2. ((...a) => {

console.log(a);

}) (3, 4, 5);

1. ((...a) => {

const x = a[0] == a[1] ? {

\_id: 1,

myName: 'saleel',

myCity: 'baroda',

myState: 'GJ'

} : {

\_id: 2,

myName: 'sharmin',

myCity: 'pune',

myState: 'MH'

}

console.log(x);

}) (4, 4);

1. const person1 = {

\_id: 1,

myName: 'saleel',

myCity: 'baroda',

myState: 'GJ'

};

const person2 = {

\_id: 2,

myName: 'sharmin',

myCity: 'pune',

myState: 'MH'

};

((...a) => {

const x = a[0] == a[1] ? person1 : person2;

console.log(x);

}) (3, 4);

1. const arr = new Array();

arr.push(function f1() {

console.log("Sharmin")

});

arr[0]();

1. const arr = new Array();

arr.push(function f1() {

console.log("Sharmin")

});

arr[0].f1();

1. const arr = new Array();

arr.push((a) => {

console.log(a);

});

arr.push((b) => {

console.log(b);

});

arr[0]('Saleel...');

1. const arr = new Array();

arr.push((a) => {

console.log(a);

});

arr.push((b) => {

console.log(b);

});

const [x, y] = [...arr];

x("Sharmin");

1. let x = (midday) => {

console.log(midday == "AM" ? "Before Midday" : midday == "PM"? "After Midday": "Mid Night");

}

x("PM");

1. let x = (quarter) => {

return quarter == 'October' || quarter == 'November' || quarter == 'December' ? "4th Quarter" : "N/A";

}

console.log(x("October"));

1. const Q1 = ["January", "February", "March"]

const Q2 = ["April", "May", "June"]

const Q3 = ["July", "August", "September"]

const Q4 = ["October", "November", "December"]

const x = [...Q1, ...Q2, ...Q3, ...Q4];

console.log(x);

1. const obj = new Object();

obj.name = "Saleel",

obj.phone = 9850884228,

obj.city = "Baroda",

obj.state = "GJ"

console.log(obj);

1. let obj1 = { name: "Saleel" };

let obj2 = {city: "Baroda" };

let obj3 = { ...obj1, ...obj2 };

console.log(obj3);

**Answers**

1. 4

2. 1 2 3 4

3. 1 2

4. 1 2 [ 3, 4 ]

5. 2

6. 8

7. saleel

8. [ 11, 12, <5 empty items>, 'saleel' ]

9. 10 20

10. 20 10

11. [ 200, 2, 3 ]

[ 1, 2, 3 ]

12. [ 10, 20, 1, 2, 3 ]

13. [ 0, [ 1, 2, 3 ], 4, 5, 6 ]

14. 1 2

15. 1 2 3

16. 1 2 [ 3, 4, 5, 6 ]

17. One

18. Three

19. [ 'One', 'Two', 'Three' ]

20. Set(3) { 'One', 'Two', 'Three' }

21. [ 1, 2, 3 ]

22. [ [ 1, 2, 3 ] ]

23. s

a

l

e

e

l

24. 1 2 [ 3, 4, 5, 6 ]

25. 1 2

26. 10

27. 11

28. s a l e e l

29. 1 3

30. 1 4 [ 5 ]

31. 1001 saleel baroda

32. 1001 saleel baroda GJ

33. 1001 { ename: 'saleel', city: 'baroda' } GJ

34. 1

2

35. 'Rest element must be last element'

36. { x: 'saleel', y: 'sharmin', z: 'ruhan' }

This is an error

37. {"\_id":1,"myName":"saleel","myCity":"baroda","myState":"GJ"}

38. ["Saleel",false,9850884228]

39. { myNameIs: '"saleel"' }

40. [ 3, 4, 5 ]

41. { \_id: 1, myName: 'saleel', myCity: 'baroda', myState: 'GJ' }

42. { \_id: 2, myName: 'sharmin', myCity: 'pune', myState: 'MH' }

43. Sharmin

44. Error

45. Sharmin

46. Sharmin

47. After Midday

48. 4th Quarter

49. [

'January', 'February',

'March', 'April',

'May', 'June',

'July', 'August',

'September', 'October',

'November', 'December'

]

50. { name: 'Saleel', phone: 9850884228, city: 'Baroda', state: 'GJ' }

51. { name: 'Saleel', city: 'Baroda' }