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);

**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.

35.

36.

37.

38.

39.

40.