1. **What's the output?**

!!null;

!!"";

!!1;

* A: false true false
* B: false false true
* C: false true true
* D: true true false

###### What does this return?

[..."Lydia"]; //… - spread operator – spreads elements of the following array/string

* A: ["L", "y", "d", "i", "a"]
* B: ["Lydia"]
* C: [[], "Lydia"]
* D: [["L", "y", "d", "i", "a"]]

###### What does this return?

const firstPromise = new Promise((res, rej) => {

setTimeout(res, 500, "one"); //the method res will be called in 500 milleseconds with argument “one”

});

const secondPromise = new Promise((res, rej) => {

setTimeout(res, 100, "two");

});

Promise.race([firstPromise, secondPromise]).then(res => console.log(res));

//race – the method of Promise that gets several promises and performs an one that is resolved first

* A: "one"
* B: "two"
* C: "two" "one"
* D: "one" "two"

###### What's the output?

let person = { name: "Lydia" };

const members = [person];

person = null;

console.log(members);

* A: null
* B: [null]
* C: [{}]
* D: [{ name: "Lydia" }]

###### What's the output?

const person = {

name: "Lydia",

age: 21

};

for (const item in person) { //iterates keys of an object

console.log(item);

}

* A: { name: "Lydia" }, { age: 21 }
* B: "name", "age"
* C: "Lydia", 21
* D: ["name", "Lydia"], ["age", 21]

###### What's the output?

console.log(3 + 4 + "5");

* A: "345"
* B: "75"
* C: 12
* D: "12"

###### What's the output?

const a = {};

const b = { key: "b" };

const c = { key: "c" };

//Any object key should be a string ; if an object doesn’t have the method

//toString , JS will convert the object to [object Object]

a[b] = 123;

a[c] = 456;

console.log(a[b]);

* A: 123
* B: 456
* C: undefined
* D: ReferenceError

###### What's the output?

const numbers = [1, 2, 3];

numbers[10] = 11;

console.log(numbers.length);

* A: 11
* B: 4
* C: Error

###### What's the value of num?

const num = parseInt("7\*6");

* A: 42
* B: "42"
* C: 7
* D: NaN

###### What's the output?

function getInfo(member, year) {

member.name = "Lydia";

year = "1998";

}

const person = { name: "Sarah" };

const birthYear = "1997";

getInfo(person, birthYear);

console.log(person, birthYear);

* A: { name: "Lydia" }, "1997"
* B: { name: "Sarah" }, "1998"
* C: { name: "Lydia" }, "1998"
* D: { name: "Sarah" }, "1997"

###### What's the output?

function greeting() {

throw "Hello world!";

}

function sayHi() {

try {

const data = greeting();

console.log("It worked!", data);

} catch (e) {

console.log("Oh no an error:", e);

}

}

sayHi();

* A: It worked! Hello world!
* B: Oh no an error: undefined
* C: SyntaxError: can only throw Error objects
* D: Oh no an error: Hello world!

###### What's the output?

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

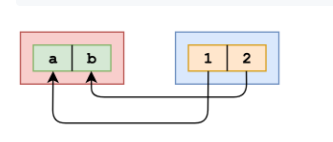
const [y] = numbers;

console.log(y);

* A: [[1, 2, 3, 4, 5]]
* B: [1, 2, 3, 4, 5]
* C: 1
* D: [1]

JS may unpack values from arrays through destructuring. For example:

[a, b] = [1, 2];



###### What's the output?

const user = { name: "Lydia", age: 21 };

const admin = { admin: true, ...user };

console.log(admin);

* A: { admin: true, user: { name: "Lydia", age: 21 } }
* B: { admin: true, name: "Lydia", age: 21 }
* C: { admin: true, user: ["Lydia", 21] }
* D: { admin: true }

###### What's the output?

const settings = {

username: "lydiahallie",

level: 19,

health: 90

};

const data = JSON.stringify(settings, ["level", "health"]);//considered only // specified keys

console.log(data);

* A: "{"level":19, "health":90}"
* B: "{"username": "lydiahallie"}"
* C: "["level", "health"]"
* D: "{"username": "lydiahallie", "level":19, "health":90}"

###### What's the output?

let num = 10;

const increaseNumber = () => num++;

const increasePassedNumber = number => number++;

const num1 = increaseNumber();

const num2 = increasePassedNumber(num1);

console.log(num1);

console.log(num2);

* A: 10, 10
* B: 10, 11
* C: 11, 11
* D: 11, 12

###### What's the output?

[1, 2, 3, 4].reduce((x, y) => console.log(x, y));

* A: 1 2 and 3 3 and 6 4
* B: 1 2 and 2 3 and 3 4
* C: 1 undefined and 2 undefined and 3 undefined and 4 undefined
* D: 1 2 and undefined 3 and undefined 4

###### What's the output?

function addToList(item, list) {

return list.push(item);

}

const result = addToList("apple", ["banana"]);

console.log(result);

* A: ['apple', 'banana']
* B: 2
* C: true
* D: undefined

###### What is the output?

const list = [1 + 2, 1 \* 2, 1 / 2]

console.log(list)

* A: ["1 + 2", "1 \* 2", "1 / 2"]
* B: ["12", 2, 0.5]
* C: [3, 2, 0.5]
* D: [1, 1, 1]

###### What is the output?

function sayHi(name) {

return `Hi there, ${name}`

}

console.log(sayHi())

* A: Hi there,
* B: Hi there, undefined
* C: Hi there, null
* D: ReferenceError

###### What's the output?

console.log("I want pizza"[0])

* A: """
* B: "I"
* C: SyntaxError
* D: undefined

###### What is the output?

function checkAge(age) {

if (age < 18) {

const message = "Sorry, you're too young."

} else {

const message = "Yay! You're old enough!"

}

return message

}

console.log(checkAge(21))

* A: "Sorry, you're too young."
* B: "Yay! You're old enough!"
* C: ReferenceError
* D: undefined

###### What's the output?

function sum(num1, num2 = num1) {

console.log(num1 + num2)

}

sum(10)

* A: NaN
* B: 20
* C: ReferenceError
* D: undefined

###### What's the output?

let newList = [1, 2, 3].push(4)

console.log(newList.push(5))

* A: [1, 2, 3, 4, 5]
* B: [1, 2, 3, 5]
* C: [1, 2, 3, 4]
* D: Error