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

**function sayHi() {**

**console.log(name);**

**console.log(age);**

**var name = 'Lydia';**

**let age = 21;**

**}**

**sayHi();**

* 1. Lydia and undefined
  2. Lydia and ReferenceError
  3. ReferenceError and 21
  4. undefined and ReferenceError

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

**for (var i = 0; i < 3; i++) {**

**setTimeout(() => console.log(i), 1);**

**}**

**for (let i = 0; i < 3; i++) {**

**setTimeout(() => console.log(i), 1);**

**}**

* 1. 0 1 2 and 0 1 2
  2. 0 1 2 and 3 3 3
  3. 3 3 3 and 0 1 2
  4. None of above

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

**const shape = {**

**radius: 10,**

**diameter() {**

**return this.radius \* 2;**

**},**

**perimeter: () => 2 \* Math.PI \* this.radius,**

**};**

**console.log(shape.diameter());**

**console.log(shape.perimeter());**

* 1. 20 and 62.83185307179586
  2. **20 and NaN**
  3. 20 and 63
  4. NaN and 63

1. **Which one is true?**

const bird = {

size: 'small',

};

const mouse = {

name: 'Mickey',

small: true,

};

* 1. mouse.bird.size is not valid
  2. mouse[bird.size] is not valid
  3. mouse[bird["size"]] is not valid
  4. All of them are valid

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

**let c = { greeting: 'Hey!' };**

**let d;**

**d = c;**

**c.greeting = 'Hello';**

**console.log(d.greeting);**

* 1. Hello
  2. Hey!
  3. Undefined
  4. ReferenceErro
  5. TypeError

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

**let a = 3;**

**let b = new Number(3);**

**let c = 3;**

**console.log(a == b);**

**console.log(a === b);**

**console.log(b === c);**

* 1. true false true
  2. false false true
  3. true false false
  4. false true true

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

**let greeting;**

**greetign = {}; // Typo!**

**console.log(greetign);**

* 1. {}
  2. ReferenceError: greetign is not defined
  3. Undefined
  4. None of these

1. **What happens when we do this?**

**function bark() {**

**console.log('Woof!');**

**}**

**bark.animal = 'dog';**

* 1. Nothing, this is totally fine!
  2. SyntaxError. You cannot add properties to a function this way.
  3. "Woof" gets logged.
  4. ReferenceError

1. **All object have prototypes.**
   1. True
   2. False
2. **What's the output?**

**function sum(a, b) {**

**return a + b;**

**}**

**sum(1, '2');**

* 1. NaN
  2. TypeError
  3. “12”
  4. 3

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

**let number = 0;**

**console.log(number++);**

**console.log(++number);**

**console.log(number);**

* 1. 1 1 2
  2. 1 2 2
  3. 0 2 2
  4. 0 1 2

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

function checkAge(data) {

if (data === { age: 18 }) {

console.log('You are an adult!');

} else if (data == { age: 18 }) {

console.log('You are still an adult.');

} else {

console.log(`Hmm.. You don't have an age I guess`);

}

}

checkAge({ age: 18 });

* 1. You are an adult!
  2. You are still an adult.
  3. Hmm.. You don't have an age I guess

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

function getAge(...args) {

console.log(typeof args);

}

getAge(21);

* 1. “numbe”
  2. "array"
  3. "object"
  4. "NaN"

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

**function getAge() {**

**'use strict';**

**age = 21;**

**console.log(age);**

**}**

**getAge();**

* 1. 21
  2. undefined
  3. ReferenceError
  4. TypeError

1. **What's the value of sum?**

**const sum = eval('10\*10+5');**

* 1. 105
  2. "105"
  3. TypeError
  4. "10\*10+5"

1. **How long is cool\_secret accessible?**

**sessionStorage.setItem('cool\_secret', 123);**

* 1. Forever, the data doesn't get lost
  2. When the user closes the tab.
  3. When the user closes the entire browser, not only the tab.
  4. When the user shuts off their computer.

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

**var num = 8;**

**var num = 10;**

**console.log(num);**

* 1. 8
  2. 10
  3. SyntaxError
  4. ReferenceError

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

**const obj = { 1: 'a', 2: 'b', 3: 'c' };**

**const set = new Set([1, 2, 3, 4, 5]);**

**obj.hasOwnProperty('1');**

**obj.hasOwnProperty(1);**

**set.has('1');**

**set.has(1);**

* 1. false true false true
  2. false true true true
  3. true true false true
  4. true true true true

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

**const obj = { a: 'one', b: 'two', a: 'three' };**

**console.log(obj);**

* 1. { a: "one", b: "two" }
  2. { b: "two", a: "three" }
  3. { a: "three", b: "two" }
  4. SyntaxError

1. **The JavaScript global execution context creates two things for you: the global object, and the "this" keyword.**
   1. True
   2. false
   3. it depends
2. **What's the output?**

**for (let i = 1; i < 5; i++) {**

**if (i === 3) continue;**

**console.log(i);**

**}**

* 1. 1 2a
  2. 1 2 3
  3. 1 2 4
  4. 1 3 4

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

**const a = {};**

**const b = { key: 'b' };**

**const c = { key: 'c' };**

**a[b] = 123;**

**a[c] = 456;**

**console.log(a[b]);**

* 1. 123
  2. 456
  3. undefined
  4. ReferenceError

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

**const a = {};**

**const b = { key: 'b' };**

**const c = { key: 'c' };**

**a[b] = 123;**

**a[c] = 456;**

**console.log(a[b]);**

* 1. 123
  2. 456
  3. Undefined
  4. ReferenceError

1. **What is the event.target when clicking the button?**

<div onclick="console.log('first div')">

<div onclick="console.log('second div')">

<button onclick="console.log('button')">

Click!

</button>

</div>

</div>

* 1. Outer div
  2. Inner div
  3. Button
  4. An array of all nested elements.

1. **When you click the paragraph, what's the logged output?**

<div onclick="console.log('div')">

<p onclick="console.log('p')">

Click here!

</p>

</div>

* 1. p div
  2. div p
  3. p
  4. div

1. **Everything in JavaScript is either a...**
   1. primitive or object
   2. function or object
   3. trick question! only objects
   4. number or object
2. **What's the output?**

**console.log(3 + 4 + '5');**

* 1. "345"
  2. "75"
  3. 12
  4. "12"

1. **What's the value of num?**

**const num = parseInt('7\*6', 10);**

* 1. 42
  2. "42"
  3. 7
  4. NaN

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

const set = new Set([1, 1, 2, 3, 4]);

console.log(set);

1. [1, 1, 2, 3, 4]
2. [1, 2, 3, 4]
3. {1, 1, 2, 3, 4}
4. {1, 2, 3, 4}
5. **Is this a pure function?**

**function sum(a, b) {**

**return a + b;**

**}**

1. Yes
2. No