**Question:** What is typeof []

**Answer:** Object. Actually Array is derived from Object. If you want to check array use Array.isArray(arr)

**Question:** What is typeof arguments

**Answer:** Object. arguments are array like but not array. it has length, can access by index but can't push pop, etc.

**Question:** What is 2+true

**Answer:** 3. The plus operator between a number and a boolean or two boolean will convert boolean to number. Hence, true converts to 1 and you get result of 2+1

**Question:** What is '6'+9

**Answer:** 69. If one of the operands of the plus (+) operator is string it will convert other number or boolean to string and perform a concatenation. For the same reason, "2"+true will return "2true"

**Question:** What is the value of 4+3+2+"1"

**Answer:** 91 . The addition starts from the left, 4+3 results 7 and 7+2 is 9. So far, the plus operator is performing addition as both the operands are number. After that 9 + "1" where one of the operands is string and plus operator will perform concatenation.

**Question:** What is the value of "1"+2+4

**Answer:** "124". For this one "1" + 2 will produce "12" and "12"+4 will generates "124".

**Question:** What is the value of -'34'+10

**Answer:** -24. minus(-) in front of a string is an unary operator that will convert the string to a number and will make it negative. Hence, -'34' becomes, -34 and then plus (+) will perform simple addition as bot

h the operands are number.

**Question:** What is the value of +'dude'

**Answer:** NaN. The plus (+) operator in front of a string is an unary operator that will try to convert the string to number. Here, JavaScript will fail to convert the "dude" to a number and will produce NaN.

**Question:** If you have var y = 1, x = y = typeof x; What is the value of x?

**Answer:** "undefined"

**Question:** for var a = (2, 3, 5); what is the value of a?

**Answer:** 5. The comma operator evaluates each of its operands (from left to right) and retuwrns the value of the last operand. ref: [MDN](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/Comma_Operator)

**Question:** for var a = (1, 5 - 1) \* 2 what is the value of a?

**Answer:** 8

**Question:** What is the value of !'bang'

**Answer:** false. ! is NOT. If you put ! in front of truthy values, it will return false. Using !! (double bang) is a tricky way to check anything truthy or falsy by avoiding implicit type conversion of == comparison.

**Question:** What is the value of parseFloat('12.3.4')

**Answer:** 12.3

**Question**var t = 9,7,6;

**Answer**Uncaught SyntaxError: Unexpected number

**Question:** What is the value of Math.max([2,3,4,5]);

**Answer:** NaN

**Question:** 3 instanceof Number

**Answer:** false

**Question:**null == undefined

**Answer:** true

**Question:**null === undefined

**Answer:** false

**Question:**[] == []&&[] === []

**Answer:** false

**Question:**What is the value of !!function(){};

**Answer:** true

**Question:** What is the value of typeof bar

**Answer:** "undefined"

**Question:** What is the value of typeof null

**Answer:** "object"

**Question:** If var a = 2, b =3 what would be value of a && b

**Answer:** 3

**Question:** If  10 && 4

Answer : 4

**Question:** What would be consoled var foo = 'outside'; function logIt(){console.log(foo); var foo = 'inside';} logIt();

**Answer:** undefined

**Question:** What is -5%2

**Answer:**-1. the result of remainder always get the symbol of first operand

**Question:** Why .1+.2 != .3

**Answer: true**

**Question: why .1+.2**

**Answer** 0.30000000000000004

**Question: 42..toString()**

**Anwser:** "42"

**Question:** 4.2..toString

**Anwser:** //SyntaxError: Unexpected token .

**Question:**42 . toString()

**Anwser:** "42"

**Question:** typeof(NaN)

**Anwser:**"number"

**Question:** 2 in [1,2]

**Anwser:** false. Because "in" returns whether a particular property/index available in the Object. In this case object has index 0 and 1 but don't have 2. Hence you get false.

**Question: 1.1+1.3**

**Anwser:** 2.4000000000000004

**Question: 1.1+1.8**

**Anwser:** 2.9000000000000004

**Question) what is the difference between an Anonymous Function and a named function?**

Anonymous functions exist only after they are called; whereas, Named functions to exist even if not called.

**Question)what is the difference between remove () and removeChild ()?**

The remove() function just removes the element. Whereas, the removeChild() returns the deleted element.

**Question)console.log(1 < 2 < 3);Ans) true**

**console.log(3 > 2 > 1);Ans) fasle**

The first statement returns true which is as expected.The second returns false because of how the engine works regarding operator associativity for < and >. It compares left to right, so 3 > 2 > 1 JavaScript translates to true > 1. true has value 1, so it then compares 1 > 1, which is false

**………..Question)for (let i = 0; i < 5; i++) {**

**setTimeout(function() { console.log(i); }, i \* 1000 );**

**}**

It will print 0 1 2 3 4, because we use **let** instead of **var** here. The variable i is only seen in the for loop’s block scope.

**Question)0.1+0.7**

0.7999999999999999

**0.1+0.2**

0.30000000000000004

**Soulation:** function areTheNumbersAlmostEqual(num1, num2) {

return Math.abs( num1 - num2 ) < Number.EPSILON;

}

console.log(areTheNumbersAlmostEqual(0.1 + 0.2, 0.3));

**Question)**

**console.log(1 + "2" + "2");** Ans) 122

**console.log(1 + +"2" + "2");** Ans) 32

**console.log(1 + -"1" + "2");** Ans) 02

**console.log(+"1" + "1" + "2");** Ans) 112

**console.log( "A" - "B" + "2");** Ans) NaN2

**console.log( "A" - "B" + 2);** Ans) NaN