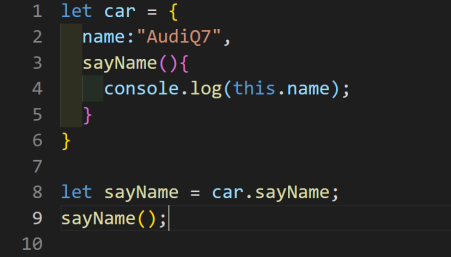


**JS Implementation**

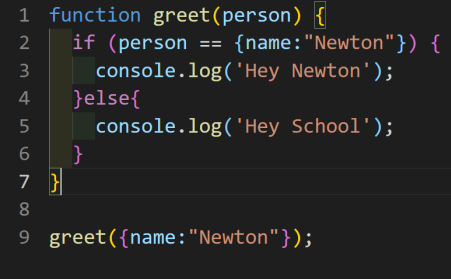
**Question Resources**

**Difficulty Level: Easy, Medium & Hard Mixed**

**1. What’s the output of this code?**

****

**2. What’s the output of this code?**

****

**3. What’s the output of this code?**

**var car = new Vehicle("Honda", "white", "2010", "UK");**

**console.log(car);**

**function Vehicle(model, color, year, country) {**

**this.model = model;**

**this.color = color;**

**this.year = year;**

**this.country = country;**

**}**

**● 1: Undefined**

**● 2: ReferenceError**

**● 3: null**

**● 4: {model: "Honda", color: "white", year: "2010", country: "UK"}**

**4. What’s the output of this code? function foo() {**

**let x = y = 0;**

**x++;**

**y++;**

**return x;**

**}**

**console.log(foo(), typeof x, typeof y);**

**● 1: 1, undefined and undefined**

**● 2: ReferenceError: X is not defined ● 3: 1, undefined and number**

**● 4: 1, number and number**

**5. What’s the output of this code? function main(){**

**console.log('A');**

**setTimeout(**

**function print(){ console.log('B'); } ,0);**

**console.log('C');**

**}**

**main();**

**● 1: A, B and C**

**● 2: B, A and C**

**● 3: A and C**

**● 4: A, C and B**

**6. What’s the output of this code?**

**var y = 1;**

**if (function f(){}) {**

**y += typeof f;**

**}**

**console.log(y);**

**● 1: 1function**

**● 2: 1object**

**● 3: ReferenceError**

**● 4: 1undefined**

**7. What’s the output of this code? var myChars = ['a', 'b', 'c', 'd'] delete myChars[0];**

**console.log(myChars);**

**console.log(myChars[0]);**

**console.log(myChars.length);**

**● 1: [empty, 'b', 'c', 'd'], empty, 3 ● 2: [null, 'b', 'c', 'd'], empty, 3**

**● 3: [empty, 'b', 'c', 'd'], undefined, 4 ● 4: [null, 'b', 'c', 'd'], undefined, 4**

**8. What’s the output of this code? console.log(1 < 2 < 3);**

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

**● 1: true, true**

**● 2: true, false**

**● 3: SyntaxError, SyntaxError,**

**● 4: false, false**

**9. What’s the output of this code in non-strict mode? function printNumbers(first, second, first) {**

**console.log(first, second, first);**

**}**

**printNumbers(1, 2, 3);**

**● 1: 1, 2, 3**

**● 2: 3, 2, 3**

**● 3: SyntaxError: Duplicate parameter name not allowed in this context ● 4: 1, 2, 1**

**10. What’s the output of the following code? const arrowFunc = () => arguments.length;**

**console.log(arrowFunc(1, 2, 3));**

**● 1: ReferenceError: arguments is not defined**

**● 2: 3**

**● 3: undefined**

**● 4: null**

**11. What’s the output of the following code? console.log(10 == [10]);**

**console.log(10 == [[[[[[[10]]]]]]]);**

**● 1: True, True**

**● 2: True, False**

**● 3: False, False**

**● 4: False, True**

**12. What’s the output of the following code? console.log([0] == false);**

**if([0]) {**

**console.log("I'm True");**

**} else {**

**console.log("I'm False");**

**}**

**● 1: True, I'm True**

**● 2: True, I'm False**

**● 3: False, I'm True**

**● 4: False, I'm False**

**13. What’s the output of the following code? async function func() {**

**return 10;**

**}**

**console.log(func());**

**● 1: Promise {<fulfilled>: 10}**

**● 2: 10**

**● 3: SyntaxError**

**● 4: Promise {<rejected>: 10}**

**14. What’s the output of the following code? async function func() {**

**await 10;**

**}**

**console.log(func());**

**● 1: Promise {<fulfilled>: 10}**

**● 2: 10**

**● 3: SyntaxError**

**● 4: Promise {<resolved>: undefined}**

**15. What’s the output of the following code? let myNumber = 100;**

**let myString = '100';**

**if (!typeof myNumber === "string") {**

**console.log("It is not a string!");**

**} else {**

**console.log("It is a string!");**

**}**

**if (!typeof myString === "number"){**

**console.log("It is not a number!")**

**} else {**

**console.log("It is a number!");**

**}**

**● 1: SyntaxError**

**● 2: It is not a string!, It is not a number!**

**● 3: It is not a string!, It is a number!**

**● 4: It is a string!, It is a number!**

**16. What’s the output of the following code? class A {**

**constructor() {**

**console.log(new.target.name)**

**}**

**}**

**class B extends A { constructor() { super() } }**

**new A();**

**new B();**

**1: A, A**

**2: A, B**

**17. What’s the output of the following code? const [x, ...y,] = [1, 2, 3, 4];**

**console.log(x, y);**

**1: 1, [2, 3, 4]**

**2: 1, [2, 3]**

**3: 1, [2]**

**4: SyntaxError**

**18. What’s the output of the following code? const {a: x = 10, b: y = 20} = {a: 30};**

**console.log(x);**

**console.log(y);**

**1: 30, 20**

**2: 10, 20**

**3: 10, undefined**

**4: 30, undefined**

**19. What’s the output of the following code? function area({length = 10, width = 20}) {**

**console.log(length\*width);**

**}**

**area();**

**1: 200**

**2: Error**

**3: undefined**

**4: 0**

**20. What’s the output of the following code? const props = [**

**{ id: 1, name: 'John'},**

**{ id: 2, name: 'Jack'},**

**{ id: 3, name: 'Tom'}**

**];**

**const [,, { name }] = props;**

**console.log(name);**

**1: Tom**

**2: Error**

**3: undefined**

**4: John**

**21. What’s the output of the following code? function add(item, items = []) {**

**items.push(item);**

**return items;**

**}**

**console.log(add('Orange'));**

**console.log(add('Apple'));**

**1: ['Orange'], ['Orange', 'Apple']**

**2: ['Orange'], ['Apple']**

**22. What’s the output of the following code? function myFun(x, y, ...manyMoreArgs) {**

**console.log(manyMoreArgs)**

**}**

**myFun(1, 2, 3, 4, 5);**

**myFun(1, 2);**

**1: [3, 4, 5], undefined**

**2: SyntaxError**

**3: [3, 4, 5], []**

**4: [3, 4, 5], [undefined]**

**23. What’s the output of the following code? function\* myGenFunc() {**

**yield 1;**

**yield 2;**

**yield 3;**

**}**

**var myGenObj = new myGenFunc;**

**console.log(myGenObj.next().value);**

**1: 1**

**2: undefined**

**3: SyntaxError**

**4: TypeError**

**24. What’s the output of the following code? let count = 10;**

**(function innerFunc() {**

**if (count === 10) {**

**let count = 11;**

**console.log(count);**

**}**

**console.log(count);**

**})();**

**1: 11, 10**

**2: 11, 11**

**3: 10, 11**

**4: 10, 10**

**25. What’s the output of the following code? let zero = new Number(0);**

**if (zero) {**

**console.log("If");**

**} else {**

**console.log("Else");**

**}**

**26. What’s the output of the following code? class Vehicle {**

**constructor(name) {**

**this.name = name;**

**}**

**start() {**

**console.log(`${this.name} vehicle started`); }**

**}**

**class Car extends Vehicle {**

**start() {**

**console.log(`${this.name} car started`);**

**super.start();**

**}**

**}**

**const car = new Car('BMW');**

**console.log(car.start());**

**● 1: SyntaxError**

**● 2: BMW vehicle started, BMW car started ● 3: BMW car started, BMW vehicle started ● 4: BMW car started, BMW car started**

**27. What’s the output of the following code? function Person() { }**

**Person.prototype.walk = function() {**

**return this;**

**}**

**Person.run = function() {**

**return this;**

**}**

**let user = new Person();**

**let walk = user.walk;**

**console.log(walk());**

**let run = Person.run;**

**console.log(run());**

**● 1: undefined, undefined**

**● 2: Person, Person**

**● 3: SyntaxError**

**● 4: Window, Window**

**28. What’s the output of the following code? const squareObj = new Square(10);**

**console.log(squareObj.area);**

**class Square {**

**constructor(length) {**

**this.length = length;**

**}**

**get area() {**

**return this.length \* this.length;**

**}**

**set area(value) {**

**this.area = value;**

**}**

**}**

**● 1: 100**

**● 2: ReferenceError**

**29. Is it a valid array?**

let arr = let arr = [2,'A',"B",true,[6,7,[9,0]]];

**30. What’s the output of the following code?**

let a = new String("abc");

let b = new Object("abc");

if(a==b){

console.log("yes");

}else{

console.log("no");

}

**31. What’s the output of the following code?**

let a = new String("abc");

let b = new String("abc");

if(a==b){

console.log("yes");

}else{

console.log("no");

}

**32. What’s the output of the following code?**

let a = new String("abc");

let b = "abc";

if(a==b){

console.log("yes");

}else{

console.log("no");

}

**33. What’s the output of the following code?** console.log(1);

console.log(2);

setTimeOut(() => {

console.log(3);

},0)

console.log(4);

**34. What’s the output of the following code?**

var num = 0;

function run(){

console.log(num);

var num = 1;

}

run();

**35. What will the code below output to the console and why?**

var myObject = {

foo: "bar",

func: function() {

var self = this;

console.log("outer func: this.foo = " + this.foo);

console.log("outer func: self.foo = " + self.foo);

(function() {

console.log("inner func: this.foo = " + this.foo);

console.log("inner func: self.foo = " + self.foo);

}());

}

};

myObject.func();

**36. In what order will the numbers 1-4 be logged to the console when the code below is executed? Why?**

**(function() {**

**console.log(1);**

**setTimeout(function(){console.log(2)}, 1000);**

**setTimeout(function(){console.log(3)}, 0);**

**console.log(4);**

**})();**

**37. Consider the code snippet below. What will the console output be and why?**

**(function(x) {**

**return (function(y) {**

**console.log(x);**

**})(2)**

**})(1);**

**38. Testing your this knowledge in JavaScript: What is the output of the following code?**

**var length = 10;**

**function fn() {**

**console.log(this.length);**

**}**

**var obj = {**

**length: 5,**

**method: function(fn) {**

**fn();**

**arguments[0]();**

**}**

**};**

**obj.method(fn, 1);**

**39. What will the following code output and why? var b = 1;**

**function outer(){**

**var b = 2**

**function inner(){**

**b++;**

**var b = 3;**

**console.log(b)**

**}**

**inner();**

**}**

**outer();**