1. What’s the result of executing this code and why.

|  |
| --- |
| var a = 1; function b() {  a = 10;  return;  function a() {} } b(); console.log(a); |

2. Output of Following Code

|  |
| --- |
| [...[...'...']].length |

3. What is the final value of obj in the following example?

|  |
| --- |
| const obj = { foo: 1 }  obj.bar = 2 |

4. What would be the constructor of roger in the example below?

|  |
| --- |
| class Animal {  constructor(noise) {  this.noise = noise  }   speak() {  console.log(this.noise)  } }  class Dog extends Animal {  constructor(size) {  super('bark!')   this.size = size  } }  const roger = new Dog('large') |

5. What is the result of the following?

|  |
| --- |
| const speed = 'quick'  `The ${speed} brown fox jumps over the lazy dog.` |

6. Select the value of obj[123]

|  |
| --- |
| const obj = {  } |

7. What is the final value of array below?

|  |
| --- |
| const array = [1, 2, 3] const extension = [4, 5, 6]  array = [...extension, ...array] |

8. ES6 gives us a "let" statement - a new way to declare variables. Can you guess what it does?

|  |
| --- |
| let x = 150;   if(x > 100) {  let x = 1;  }   console.log(x); |

9. What is the alternative of following code in ES6.

|  |
| --- |
| var arr = [12,5,42,55,10];   function do\_something(arr) {  for (let i = 0; i < arr.length; i++) {  if(arr[i] > 32) {  return arr[i]  }  }  } |

10. Write a same function in es6 with minimum code and call the function to get output 15

|  |
| --- |
| function add(a = 10, b = 5) {  return a + b; } |

11. What will be the output from following function call?

|  |
| --- |
| function add(a = 10, b = 5) {  return a + b; }  add(11,2);  add(3);  add(undefined, 20);  add(); |

12. What will be the output of following code

|  |
| --- |
| let a = 12, b = 3;  [a, b] =[b, a];  console.log(a); console.log(b); |

13. What will be the output of following code

|  |
| --- |
| const point = [1,3]; const segment = [point , [5,5]];  const triangle = [...segment, [1, 8]] |

14. What will be the output of following code?

|  |
| --- |
| const myStory = (...params) => params;  let a = myStory(1,23,4); |

15. From following code which is the correct answer and why?

|  |
| --- |
| let arr = [16,22,44];   A. for (const iterator of arr) {  console.log(iterator) }   B. for (const key in arr) {  console.log(key) } |

16. What will be the output of following

|  |
| --- |
| function\* generator() {  var x = 0;   for (let i = 0; i < 10; i++) {  yield x++;  } }  var gen = generator();  gen.next(); gen.next();  console.log(gen.next().value); |

17. What will be the output of following

|  |
| --- |
| const compareToTen = (num) => {  p = new Promise((resolve, reject) => {  if(num > 10) {  resolve(num + " is greater than 10, success!")  } else {  reject(num + " is less than 10, error!")  }  })  return p }  compareToTen(15)  .then(result => console.log(result))  .catch(error => console.log(error))  compareToTen(8)  .then(result => console.log(result))  .catch(error => console.log(error)) |

18. What will be the output of following

|  |
| --- |
| const arrayOfWords = ['cucumber', 'tomatos', 'avocado'] const complicatedArray = ['cucumber', 44, true]  const makeAllCaps = (array) => {  return new Promise((resolve, reject) => {    let capsArray = array.map(word => {    if(typeof word === 'string'){  return word.toUpperCase()  } else {  reject('Error: Not all items in the array are strings!')  }  })  resolve(capsArray)  }) }   const sortWords = (array) => {  return new Promise((resolve, reject) => {  if(array) {  array.forEach((el) => {  if(typeof el !== 'string'){  reject('Error: Not all items in the array are strings!')  }  })  resolve(array.sort());  } else {  reject('Error: Something went wrong with sorting words.')  } }) }  makeAllCaps(arrayOfWords)  .then(sortWords)  .then((result) => console.log(result))  .catch(error => console.log(error))  makeAllCaps(complicatedArray)  .then(sortWords)  .then((result) => console.log(result))  .catch(error => console.log(error)) |

|  |
| --- |
| 19. What will be the output of following code? |

|  |
| --- |
| var fullname = 'John Doe'; var obj = {  fullname: 'Colin Ihrig',  prop: {  fullname: 'Aurelio De Rosa',  getFullname: function() {  return this.fullname;  }  } };  console.log(obj.prop.getFullname());  var test = obj.prop.getFullname;  console.log(test()); |

s