**Function**

1. What methods of function definition are correct?

|  |  |
| --- | --- |
| a) | **function** Sum(a, b) {  **return** a + b;  }; |
| b) | **const** sum = **function** (a, b) {  **return** a + b;  }; |
| c) | **var** sum = **function** Sum(a, b) {  **return** a + b;  }; |
| d) | **const** obj = {  x: **3**,  y: **5**,  sum(a, b) {  **return** a + b;  }  }; |
| e) | **const** sumMethod = 'sum';  **const** obj = {  x: **3**,  y: **5**,  [sumMethod](a, b) {  **return** a + b;  }  }; |
| f) | **const** sum = (a, b) => {  **return** a + b;  } |
| g) | **function**\* increaseCount() {  **var** count = **0**;  **while** (**true**) {  yield count++;  }  } |
| h) | **let** a = 'a', b = 'b';  **var** sum = **new** Function(a, b, 'return a+ b'); |

1. In what cases is an error displayed in the console?

|  |  |
| --- | --- |
| a) | alert(x());  **var** x = **function** () { **return** **15**; } |
| b) | alert(x());  **function** x() { **return** **15**; } |
| c) | **var** x = **function** X() { **return** **15**; }  alert(X()); |

1. What will be displayed in cases **(1)** and **(2)**?

|  |
| --- |
| **class** User {  constructor(name) { **this**.name = name }  sayHi() {  alert('Hi ' + **this**.name);  }  }  alert(**typeof** User); **// 1**  User('Petr').sayHi(); **//2** |

|  |  |
| --- | --- |
| a) | class  error |
| b) | class  Hi Petr |
| c) | class  undefined |
| d) | object  error |
| e) | object  Hi Petr |
| f) | object  undefined |
| g) | function  error |
| h) | function  Hi Petr |
| i) | function  undefined |

1. What will **alert** display?

index.html

|  |
| --- |
| <!doctype html>  <script type="module" src="1.js"></script>  <script type="module" src="2.js"></script> |

admin.js

|  |
| --- |
| **export** **let** admin = {  name: "John"  }; |

1.js

|  |
| --- |
| **import** { admin } from './admin.js';  admin.name = "Stevie"; |

2.js

|  |
| --- |
| **import** { admin } from './admin.js';  alert(admin.name); |

|  |  |
| --- | --- |
| a) | John |
| b) | Stevie |
| c) | undefined |

1. What will **rabbit.walk()** display?

|  |
| --- |
| 'use strict';  **let** animal = {  walk() {  alert("I'm walking");  }  };  **let** rabbit = {  \_\_proto\_\_: animal,  walk: **function** () {  **super**.walk();  }  };  rabbit.walk(); |

|  |  |
| --- | --- |
| a) | I'm walking |
| b) | error |
| c) | undefined |

1. What value will be assigned to **x**?

|  |
| --- |
| **var** f = **function** ff() { **return** { name: "name" } }; **var** x = f.name; |

|  |  |
| --- | --- |
| a) | error |
| b) | “name” |
| c) | "f" |
| d | “ff” |
| e) | undefined |

1. In what cases does a variable exist in **block scope**?

|  |
| --- |
| **function** foo() {  **if** (**true**) {  **var** fruit1 = 'apple'; // 1  **const** fruit2 = 'banana'; // 2  **let** fruit3 = 'strawberry'; // 3  }  } |

|  |  |
| --- | --- |
| a) | 1 |
| b) | 2 |
| c) | 3 |
| d) | no one |

1. What will be displayed in the console?

|  |
| --- |
| (**function** () {  **var** a = b = !!**1**;  alert(a);  })(); |

|  |  |
| --- | --- |
| a) | 1 |
| b) | 0 |
| c) | true |
| d) | false |
| e) | error |
| f) | undefined |

1. What will **alert** display?

|  |
| --- |
| **function** f(x) {  "use strict";  arguments[**0**] = **5**;  alert(x);  }  f(**1**); |

|  |  |
| --- | --- |
| a) | 1 |
| b) | 5 |
| c) | error |
| d) | undefined |

1. What will **user.foo()** display?

|  |
| --- |
| **function** foo() {  'use strict';  console.log(**this** === window);  }  **let** user = {  count: **10**,  foo: foo,  foo1: **function** () {  console.log(**this** === window);  }  }  user.foo(); |

|  |  |
| --- | --- |
| a) | true |
| b) | false |
| c) | error |
| d) | undefined |

1. What will **user.foo()** display?

|  |
| --- |
| **function** foo() {  'use strict';  console.log("Simple function call")  console.log(**this** === window);  }  **let** user = {  count: **10**,  foo: foo,  foo1: **function** () {  console.log(**this** === window);  }  }  **let** fun1 = user.foo1;  fun1(); |

|  |  |
| --- | --- |
| a) | true |
| b) | false |
| c) | error |
| d) | undefined |

1. What value will be assigned to **x**?

|  |
| --- |
| **var** person = {  fullName: **function** (city, country) {  **return** **this**.firstName + " " + **this**.lastName + "," + city + "," + country;  }  }  **var** person1 = {  firstName: "John",  lastName: "Doe"  }  var x = person.fullName.apply(person1, ["Oslo", "Norway"]); |

|  |  |
| --- | --- |
| a) | John Doe,Oslo,Norway |
| b) | error |
| c) | undefined undefined,Oslo,Norway |

1. What will be displayed in the console?

|  |
| --- |
| **var** obj = {};  obj.add = **function** (val1, val2) {  **var** t = **this**;  **var** inner = **function** () {  **this**.sum = val1 + val2;  }  inner();  }  obj.add(**2**, **4**);  console.log(obj.sum); |

|  |  |
| --- | --- |
| a) | 6 |
| b) | error |
| c) | undefined |

1. What value will be assigned to **x**?

|  |
| --- |
| **function** f(str) {  **if** (str === '') **return** '';  **return** f(str.substr(**1**)) + str[**0**];  }  var x = f('cat'); |

|  |  |
| --- | --- |
| a) | ‘cat’ |
| b) | ‘tac’ |
| c) | error |
| d) | undefined |

1. What will **alert** display?

|  |
| --- |
| **let** objectToString = Object.prototype.toString;  **let** arr = [];  alert(objectToString.call(arr)); |

|  |  |
| --- | --- |
| a) | [object Object] |
| b) | [object Array] |
| c) | [object] |
| d) | error |

1. What will **alert** display?

|  |
| --- |
| alert("hello Tom!", "How are you doing?"); |

|  |  |
| --- | --- |
| a) | hello Tom! |
| b) | How are you doing? |
| c) | hello Tom!How are you doing? |
| d) | empty line |
| e) | error |