1. var y = 1;

if(function f(){}){

y += typeof f;

}

console.log(y);

output

**1undefined**

1. **Write a mul function which will produce the following outputs when invoked:**

|  |  |
| --- | --- |
| 1  2 | console.log(mul(2)(3)(4)); // output : 24  console.log(mul(4)(3)(4)); // output : 48 |

Below is the answer followed by an explanation to how it works:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | function mul (x) {  return function (y) { // anonymous function  return function (z) { // anonymous function  return x \* y \* z;  };  };  } |

4.

function myFunction() {

var fruits = ["Banana", "Orange", "Apple", "Mango"];

var x = document.getElementById("demo");

x.innerHTML = Array.isArray(fruits);

}

5. var output = (function(x) {

delete x;

return x;

})(0);

console.log(output);

Output

**0**

**6. var trees = ["redwood", "bay", "cedar", "oak", "maple"];**

**delete trees[3];**

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

**output**

**5**

7. var bar = true;

console.log(bar + 0);

console.log(bar + "xyz");

console.log(bar + true);

console.log(bar + false);

**Output**

1

truexyz

2

1

9. var salary = "1000$";

(function () {

console.log("Original salary was " + salary);

var salary = "5000$";

console.log("My New Salary " + salary);

})();

**Output**

Original salary was **undefined**

My New Salary **5000$**

**10. Question:** Four people need to cross a rickety bridge at night. Unfortunately, they have only one torch and the bridge is too dangerous to cross without one. The bridge is only strong enough to support two people at a time. Not all people take the same time to cross the bridge. Times for each person:  1 min, 2 mins, 7 mins and 10 mins. What is the shortest time needed for all four of them to cross the bridge?

**Answer:** The initial solution most people will think of is to use the fastest person as an usher to guide everyone across. How long would that take? 10 + 1 + 7 + 1 + 2 = 21 mins. Is that it? No. That would make this question too simple even as a warm up question.

Let’s brainstorm a little further. To reduce the amount of time, we should find a way for 10 and 7 to go together. If they cross together, then we need one of them to come back to get the others. That would not be ideal. How do we get around that? Maybe we can have 1 waiting on the other side to bring the torch back. Ahaa, we are getting closer. The fastest way to get 1 across and be back is to use 2 to usher 1 across. So let’s put all this together.

1 and 2 go cross  
2 comes back  
7 and 10 go across  
1 comes back  
1 and 2 go across (done)

Total time = 2 + 2 + 10 + 1 + 2 = **17** mins

1. var str = "I want to remove the last word.";

var lastIndex = str.lastIndexOf(" ");

str = str.substring(0, lastIndex);

**Output**

**I want to remove the last**

**6. write a function that tests whether a string is a palindrome**

function palindrome(str) {

var len = str.length;

for ( var i = 0; i < Math.floor(len/2); i++ ) {

if (str[i] !== str[len - 1 - i]) {

return false;

}

}

return true;

}

**Output**

**palindrome(str); = false**

**palindrome(1); = true**