Juno College of Technology

Live Coding Answers

Note — these solutions are by no means the most efficient answers available. If you have a better option, feel free to go with that instead!

1. Define a function that takes a string, and returns whether it is a palindrome.

```
// create a function that will take the string as a param and determine whether it is
a palindrome
const palindrome = word => {
   let reverse = ""
   // split the string into individual letters — save as array
   const reverseArray = word.split("")
   // create a new string based on the array in reverse order
   for (let index in reverseArray) {
       reverse += reverseArray[reverseArray.length - index - 1]
   }
   // if new word equals test word, then return true
   return reverse.toLowerCase() === word.toLowerCase()
}
```

2. Define a function that takes an array of strings, and returns the most commonly occurring string in that array.

```
// create a function that takes an array of words as a param and returns most common
word
const getMostCommonWord = (array) => {
   // create empty variable to keep track of most common word
   const total = {}
   let mostCommonWord = ""
   // iterate through array
   array.map((word, i) => {
       // make the first item the most common word
       if (i === 0) mostCommonWord = word
       // if current word isn't in total object, add it and set count to 1
       total[word] = total[word] ? total[word] + 1 : 1
       // if current word has a higher count than the most common word, make it the
new most common word
       if (total[word] > total[mostCommonWord]) {
           mostCommonWord = word
       }
   })
   return mostCommonWord
```

```
}
```

3. Fizz Buzz! Define a function that takes an integer (*n*), and:
    - Console logs the number from 1 to *n*
    - Logs *fizz* instead of the number if the number is a multiple of 3
    - Logs *buzz* instead of the number if the number is a multiple of 5
    - Logs *fizzbuzz* instead of the number if the number is a multiple of 3 and 5

```
const fizzbuzz = (int) => {
   for (let n = 1; n <= int; n++) {
       if (n % 3 === 0 && n % 5 === 0) {
           console.log("fizzbuzz")
       } else if (n % 3 === 0) {
           console.log("fizz")
       } else if (n % 5 === 0) {
           console.log("buzz")
       } else {
           console.log(n)
       }
   }
}

fizzbuzz(15)
```

4. Assuming the following HTML, add functionality for an action to occur when a user clicks on of the list items:

```
<ul id="todo-app">
  <li class="item">Walk the dog</li>
  <li class="item">Pay bills</li>
  <li class="item">Make dinner</li>
  <li class="item">Code for one hour</li>
</ul>
```

Your initial instinct here would probably be to attach event listeners to each element with a class of `.item`. That's a good solution! However, in an interview it would be best to first ask the interviewer what the maximum number of elements the user can enter is. If it can never be more than 10, for example, then attaching event listeners to each `.item` would work fine. However, if there's no limit to the number of items the user can enter, then you'd want to use a more efficient solution.

If your application could end up with hundreds of event listeners, the more efficient solution would be to actually attach one event listener to the whole container, and then be able to access each item when it's actually clicked. This is called event delegation, and it's much more efficient than attaching separate event handlers.

Here's the code for event delegation:

```
document.addEventListener('DOMContentLoaded', function() {
   let app = document.getElementById('todo-app');

   // Attach event listener to whole container
   app.addEventListener('click', function(e) {
       if (e.target && e.target.nodeName === 'li') {
           let item = e.target;
           alert('You clicked on item: '+ item.innerHTML);
       }
   });
});
```

5. Define a function that takes a string and returns the number of vowels.

```
const vowels = "aeiouy" // assuming y is always a vowel for simplicity

const vowelCount = (string) => {
   let count = 0
   const stringLowerCase = string.toLowerCase() // to account for uppercase
   for (let i in stringLowerCase) {
       if (vowels.includes(stringLowerCase[i])) count ++
   }
   return count
}

console.log(vowelCount("Italy"))
```

6. Given a string, reverse every word in the sentence.

```
const reverseSentence = (sentence) => sentence
       .split(" ")
       .reverse()
       .join(" ")

console.log(reverseSentence("I love JavaScript"))
```

7. Write a function that calculates the score of a word in the game of Scrabble, following this point system:

Points	Letters
1	E, A, I, O, N, R, T, L, S, U
2	D, G
3	B, C, M, P
4	F, H, V, W, Y
5	K
8	J, X
10	Q, Z

When `countScore('quagmire');` is called, the output should be `20`.

```
const countScore = (word) => {
   // Create a place to hold the final score
   let finalScore = 0;
   // Create an array from the string
   const wordArr = [...word.toLowerCase()];
   // Loop over that new array
   wordArr.forEach(letter => {
       // the variable set here is going to be the keys/properties from the scoring
object
       // ie. 'eaionrtlsu' or 'dg'
       for (set in scoring) {
           // if that particular string of characters contains the letter
           if (set.includes(letter)) {
               // add to the final score
               // scoring[set] is accessing a value in the score function
               finalScore = scoring[set] + finalScore;
           }
       }
   })
   return finalScore;
}
```