JavaScript Technical Test

Notes:

- There is no time limit
- We recommend you submit code solutions by using jsfiddle.net or jsbin.com (or any similar services) and sending us the code snippet link for each question.

Question 1: Callback

Explain what a callback function is and provide a simple example.

Question 2: Modules

Create a customer module that takes in a first name, last name and email address. The module should only expose a *getFullName* and *getEmailAddress* method and these method should return the respective values.

Demonstrate how to use this module.

Question 3: Arrays

Consider the following code:

```
var arrayList = ['a', 'b', 'c', 'd', 'e', 'f'];
var anotherArrayList = arrayList;
```

What is the content of **anotherArrayList** from the following command, explain why.

a.) arrayList = [];
b.) arrayList.length = 0;

Question 4: Scoping

Consider the following code:

```
function ScopingTest() {
   var a = 1;
   const b = 2;
   let c = 3;

   if (b < 10) {
      var a = 10;
      const b = 11;
      let c = 12;

      console.log(a, b, c);
   }

   console.log(a, b, c);
   console.log(d, e);

   var d = 4;
   const e = 5;
}

ScopingTest();
```

What will be printed in the console? Please provide an explanation.

Question 5: Event delegation

Consider the following HTML structure representing a tab system.

```
<div class="tabs">
   <ul class="tabs-header">
      <li class="active">
         <a href="#tab-1">Tab 1</a>
      </li>
      <li>
         <a href="#tab-2">Tab 2</a>
      </li>
      <li>
         <a href="#tab-3">Tab 3</a>
      </li>
   </ul>
   <article class="tab active" id="tab-1">
      Content 1
   </article>
   <article class="tab" id="tab-2">
      Content 2
   </article>
   <article class="tab" id="tab-3">
      Content 3
   </article>
</div>
```

Write the Javascript code needed to show the correct tab when the relevant tab header is selected. Both the tab and its header are considered visible if they possess the *active* CSS class name.

Question 6: Algorithms

Write an algorithm to determine if a number n is happy.

A **happy number** is a number defined by the following process:

- Starting with any positive integer, replace the number by the sum of the squares of its digits.
- Repeat the process until the number equals 1 (where it will stay), or it **loops endlessly in a cycle** which does not include 1.
- Those numbers for which this process **ends in 1** are happy.

Return true *if* n *is a happy number, and* false *if not*.

Example 1:

```
Input: n = 19
Output: true
Explanation:
1² + 9² = 82
8² + 2² = 68
6² + 8² = 100
1² + 0² + 0² = 1
```

Example 2:

```
Input: n = 2
Output: false
```

Question 7: Algorithms

Given an array of strings strs, group **the anagrams** together. You can return the answer in **any order**.

An **Anagram** is a word or phrase formed by rearranging the letters of a different word or phrase, typically using all the original letters exactly once.

Example 1:

```
Input: strs = ["eat","tea","tan","ate","nat","bat"]
Output: [["bat"],["nat","tan"],["ate","eat","tea"]]
```

Example 2:

```
Input: strs = [""]
Output: [[""]]
```

Example 3:

```
Input: strs = ["a"]
Output: [["a"]]
```