

HTML, CSS, Reasoning and Javascript

Anuj Sunil Ghadge

anujghadge14.10@gmail.com

Result

Status: **FAIL**

Completed: Nov 26, 2022

Time: 52min

Total:

29 %

Bottom 3% of all candidates

Breakdown

Bootstrap	<div><div></div></div>	67%
HTML/CSS	<div><div></div></div>	44%
Javascript	<div><div></div></div>	0%

Question 1

HTML/CSS . CSS SELECTORS . STYLING . PREMIUM

Score: 0%

Time: **17min 31sec**

Difficulty: Easy

Description:

Write CSS which implements the following rules for **all** tables:

- The text in the table header should be gray (HEX: 808080).
- The text in the first column should be bold.
- Odd rows in the table body (1, 3, 5, etc.) should have white text (HEX: FFFFFF) on a gray background (HEX: 808080).

Answer

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Format table</title>
  <style type="text/css">
    /* Write your CSS solution here (do not edit the surrounding HTML) */
    thead{
      color : '#808080';
    }

    tr{
```



```
font-weight : 'bold';
}

.id1{
color: '#FFFFFF';
bgcolor : '808080'
}

.id3{
color: '#FFFFFF';
bgcolor : '808080'
}

.id5{
color: '#FFFFFF';
bgcolor : '808080'
}

</style>
</head>
<body>
<table>
  <thead>
    <tr><th>Rank</th><th>Name</th></tr>
  </thead>
  <tbody>
    <tr><td id = 'id1'>1.</td><td>New York</td></tr>
    <tr><td id = 'id2'>2.</td><td>Los Angeles</td></tr>
    <tr><td id = 'id3'>3.</td><td>Chicago</td></tr>
    <tr><td id = 'id4'>4.</td><td>Houston</td></tr>
  </tbody>
</table>
</body>
</html>
```

Evaluation

- ☒ The header is styled correctly: Wrong Answer
- ☒ The first column's text is styled correctly: Wrong Answer
- ☒ Odd table body rows are styled correctly: Wrong Answer

Question 2

HTML/CSS . LANGUAGE . ELEMENTS AND ATTRIBUTES . PREMIUM

Score: 100%**Time:** 50sec**Difficulty:** Easy

Description:

Select all the HTML markups that would result in the following line of text:

A very **important** word.**Answer** Correct answer | ☒ Candidate's selection☒ A very important word.

- ☒ A very **important** word.
- ☒ A very important word.
- ☐ A very

important

 word.
- ☐ A very

important

 word.
- ☐ A very

important

 word.

Question 3

HTML/CSS . CSS SELECTORS . STYLING . PREMIUM

Score: 0%

Time: 1min 15sec

Difficulty: Hard

Description:

Given the following HTML code, which snippets of CSS code would result in green text?

```
<div class="label">Label</div>
```

Answer **Correct answer** | ☒ Candidate's selection

- ☐ .label { color: red; }
div.label { color: green; }
div { color: blue; }
- ☐ .label { color: green; }
div.label { color: red; }
div { color: blue; }
- ☒ .label { color: red; }
div.label { color: blue; }
div { color: green; }
- ☒ .label { color: blue; }
div.label { color: red; }
div { color: green !important; }

Question 4

JAVASCRIPT . ARITHMETIC . PREMIUM . NEW

Score: 0%

Time: 4min 38sec

Difficulty: Easy

Description:

A large package can hold five items, while the small package can hold only one item. The available number of both large and small packages is limited. All items must be placed in packages and used packages have to be filled up completely.

Write a function that calculates the minimum number of packages needed to hold a given number of items. If it's not possible to meet the requirements, return -1.

For example, if we have 16 items, 2 large and 10 small packages, the function should return 8 (2 large packages + 6 small packages).

Answer

```
function minimalNumberOfPackages(items, availableLargePackages, availableSmallPackages) {  
  
}  
  
console.log(minimalNumberOfPackages(16, 2, 10));
```

Evaluation

- ☒ Sample case: Wrong Answer
- ☒ Enough packages: Wrong Answer
- ☒ Not enough packages: Wrong Answer
- ☒ Performance test on a large storehouse: Wrong Answer

Question 5

JAVASCRIPT . HTTP . PROMISE . AJAX . PREMIUM

Score: 0%**Time:** 2min 52sec**Difficulty:** Hard**Description:**

The `loadURL` method given below creates a GET request for a passed `url` parameter.

```
function loadURL(url) {  
  return new Promise(function(resolve, reject) {  
    var request = new XMLHttpRequest();  
    request.open('GET', url);  
  
    request.onload = function() {  
      if (request.status === 200) {  
        resolve(request.response);  
      }  
      else {  
        reject(new Error(request.status));  
      }  
    };  
    request.onerror = function() {  
      reject(new Error("Network Error"));  
    };  
    request.send();  
  })  
  .then(res => console.log("Success:" + res))  
  .catch(err => console.log(err))  
  .then(res => console.log(res));  
}
```

Select all correct statements if the GET request to "http://www.example.com" returns "Done".

Answer Correct answer | ☒ Candidate's selection☒ The second `then` method always prints the same `res` object as the first `then` method.

- ☐ The first *then* method of the *loadURL* method will print "Success:Done" in the console for a successful GET request to "http://www.example.com".
- ☒ *catch* callback executes for any *reject* call.
- ☐ The second *then* callback method is never called.
- ☐ The second *then* callback method is always called.
- ☒ The first *then* method of the *loadURL* method will print "Success:undefined" in the console for a successful GET request to "http://www.example.com".

Question 6

BOOTSTRAP . FORMS . PREMIUM

Score: 67%

Time: 1min 11sec

Difficulty: Easy

Description:

Which statements are correct for forms made using Bootstrap:

Answer **Correct answer** | ☒ Candidate's selection

- ☐ The horizontal forms should be defined as `<form class="form-hrz">`.
- ☐ The submit button should be defined as `<button type="submit" class="submit">`.
- ☒ Forms can be inline, which allows multiple elements to appear side by side.
- ☒ The checkboxes should be defined as `<input class="form-check-input" type="checkbox">`.
- ☒ The submit button should be defined as `<button type="submit" class="btnSubmit">`.

Question 7

BOOTSTRAP . STYLING . BUTTONS . PREMIUM

Score: 67%

Time: 55sec

Difficulty: Easy

Description:

Which statements are correct when using the default Bootstrap configuration?

Answer **Correct answer** | ☒ Candidate's selection

- ☒ The `btn-secondary` class will set the button color to gray.
- ☐ The `btn-sm` class will increase the size of the button.
- ☒ An `` element will look like a button.
- ☐ The `btn-primary` class will set the button color to white.

✔ The btn-info class will set the button color to light blue.

Question 8

HTML/CSS . UI . RESPONSIVE DESIGN . PREMIUM . NEW

Score:  75%

Time: 2min 13sec

Difficulty: Easy

Description:

You are working on a travel blog which currently serves mostly desktop users but you want to attract more mobile users, which mostly browse in the portrait orientation.

The site is based on a 3 column layout and there's a big fixed-size image of people riding a gondola near the top of the page.

There's also a navigation menu made of main page, contact info, merchandise and gallery links arranged horizontally.

Which of the following help improve the mobile users' experience on the site?

Answer **Correct answer** | ✔ Candidate's selection

- ☐ Make the navigation menu appear at the bottom of the page, after the blog posts.
- ☐ Make sure to only use inline HTML elements.
- ✔ **Use a picture element with source children elements** so it shows versions of the gondola image optimized for each resolution.
- ✔ **Add a meta element with name="viewport" and content="width=device-width, initial-scale=1.0"**
- ✔ Add a horizontal scrollbar so the whole background can be seen, even on smaller devices.
- ✔ **Move the navigation links to a hamburger menu.**
- ☐ Making sure the columns are always 1/3rd of the screen's width.

Question 9

HTML/CSS . ACCESSIBILITY . PREMIUM . NEW

Score:  75%

Time: 3min 47sec

Difficulty: Hard

Description:

You're working on Curious Leaf, a news website for botanical information.

One of it's pages is not very friendly to screen-reader users.

```
<head>
<style>
p.heading {
  /* Default p size is too small */
  font-size: 30px;
```

```
    }
  </style>
</head>
<body>
  <p class="heading" id="heading">
    Cabbage
  </p>
  <p id="news-content">
    One of several cultivars of Brassica oleracea (<a href= "/species?id=1990377">Read more</a>), it is
    <b>edible</b>.
  </p>
  <button onclick="subscribe()"></button>
  <span>&lt;-Subscribe</span>
  
</body>
```

Which of the following measures would help the screen-readers give more appropriate information?

Answer **Correct answer** | ☒ Candidate's selection

- ☐ Add *aria-role="presentation"* to the tag with the *news-content* id.
- ☐ Add *aria-labelledby="heading"* to the tag with the *news-content* id.
- ☒ Change the element with the id *heading* to *h1* instead of *p*.
- ☒ Replace 'edible' with all uppercase letters so it's said strongly.
- ☒ Put the 'Subscribe' text inside the *button*.
- ☒ Remove the 'Read more' text and instead put the hyperlink over 'Brassica oleracea'.
- ☐ Add a style with the attribute *speak:never* to the *head* element.
- ☒ Add an *alt* attribute to the *img* tag containing a description of the photo.

Question 10

JAVASCRIPT . ARITHMETIC . ARRAYS . ITERATION . PREMIUM

Score: 0%

Time: **9min 14sec**

Difficulty: Easy

Description:

Write a function that takes an array of 3D vectors and returns the shortest one. Each vector is represented with an array that contains 3 elements (x, y, and z). If multiple vectors have the same length, the function should return any one of them.

To determine the length of a vector use the formula: $\sqrt{x^2 + y^2 + z^2}$.

For example, for the array of 3D vectors [[1, 1, 1], [2, 2, 2], [3, 3, 3]] *findShortest* should return the first vector (array [1, 1, 1]) because it is the shortest.

Answer

```
function findShortest(vectors) {
  // Write the code that goes here
  for(vector in vectors){
    array = vector;
  }
  length = math.root.(array(x*2 + y^2 + z^2));
```

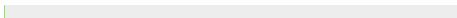
```
}  
  
var vectors = [[1, 1, 1], [2, 2, 2], [3, 3, 3]];  
var shortest = findShortest(vectors);  
console.log(shortest);
```

Evaluation

- ❌ Example case: SyntaxError at line 6
- ❌ Vectors with different lengths: SyntaxError at line 6
- ❌ Vectors with the same lengths: SyntaxError at line 6

Question 11

JAVASCRIPT . ASSOCIATIVE ARRAYS . PREMIUM . NEW

Score:  0%

Time: **6min 57sec**

Difficulty: Hard

Answer

In a graphics-based web application, the *colors* array holds color-names and their corresponding color-codes. Users can suggest and add new color-names that correspond to the color-codes, like below:

```
let colors = [];  
colors['Red'] = '#FF0000'; // suggested by user - valid color code  
colors['SomeColor'] = '#GFFF00'; // suggested by user - invalid color code (G is invalid hex number)
```

The function *isValidHexColor* correctly returns *true* if a color code is valid, else it returns *false*.

```
function isValidHexColor(colorCode) {  
  // Correct implementation. Returns true for a valid color code like #00AAFF  
}
```

Fill in the blanks in the following function that filters out array elements with invalid color codes and returns an array with valid elements:

```
function filterValid(colors) {  
  let filteredColors = [];  
  Object.keys(colors).forEach(function(colorName) {  
    if(isValidHexColor(Red)) {  
      filteredColors[#GFFF00] = #FF0000;  
    }  
  });  
  return filteredColors;  
}
```

For example, the code below should print *Red: #FF0000*:

```
let filteredColors = filterValid(colors);  
for (const [key, value] of Object.entries(filteredColors)) {  
  console.log(`${key}: ${value}`);  
}
```

Proctoring images (0)