IN512 2020 Skills Based Assessment

Due Date: Friday, 24th April, 1:00 pm – 3.00 pm – code freeze
Value: 30% of your final mark
Group Size: Individual

Learning Outcomes:

1. Use fundamental components of web pages and basic client/server communication.
2. Develop simple web-based applications using industry relevant client/server-side programming languages.
3. Use industry relevant tools and workflows in the development of web-based applications.

Submission:

- All SBA files must be submitted via GitHub Classroom. Here is the link to the repository you will be using for submission - https://classroom.github.com/a/sW48VTS1

Passing Criteria:

This assessment is criterion-referenced with a cumulative pass mark of 50%.

Fundamentals of Web Development SBA

TIME: 2 hours **TOTAL MARKS: 30**

1. **(2 marks)** You have been given the following HTML structure:

```
<div>first</div>
<div>second</div>
<div>third</div>
<div>fourth</div>
<div>fifth</div>
<div>sixth</div>
<div>seventh</div>
<div>eighth</div>
<div>ninth</div>
```

Which renders out like this in the browser:

first
second
third
fourth
fifth
sixth
seventh
eighth
ninth

Without changing the HTML (e.g. adding any classes or IDs), create **ONE** CSS rule that will make this HTML render like this in the browser (i.e., every 3rd div is now skyblue):

first
second
third
fourth
fifth
sixth
seventh
eighth
ninth

2. **(4 marks)** You have been given the following HTML structure:

```
<div class="cat">
  <input type="text" />
  <input type="text" />
  <input type="text" />
</div>

<div class="dog">
  <input type="text" />
  <input type="text" />
  <input type="text" />
</div>
```

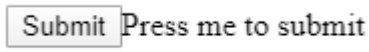
Which renders out like this in the browser:

Without changing the HTML (e.g. adding any classes or IDs), create **TWO** CSS rules that will make the HTML render like this in the browser (hint: the only style that has changed is the **background-color** – pink and grey):

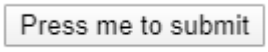
3. **(3 marks)** Each of the following HTML elements is written incorrectly. Rewrite each HTML element so they give the intended result.

```
<input type="submit">Press me to submit</input>
```

Looks like this in the browser:

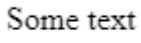


This is the intended result:




```
<div color="red">Some text</div>
```

Looks like this in the browser:



This is the intended result:

```
<img>https://images.pexels.com/photos/104827/cat-pet-animal-
domestic-104827.jpeg?auto=compress&cs=tinysrgb&dpr=1&w=500</img>
```

Looks like this in the browser:

https://images.pexels.com/photos/104827/cat-pet-animal-domestic-104827.jpeg?auto=compress&cs=tinysrgb&dpr=1&w=500

This is the intended result:

4. **(3 marks)** Consider the following code:

```
let myThing = [
    {
        "name" : "John",
        "age" : "26"
    },
    {
        "name" : "Sam",
        "age" : "18"
    },
    {
        "name" : "Billy",
        "age" : "30"
    }
];
```

Write some JavaScript code to update every **name** variable inside **myThing** to “Jason”.

5. **(6 marks)** Consider the following HTML and CSS:

```
.list-item{ padding:10px; color:maroon; }
```

```
<ol id="ageList">
    <li class="list-item">15 years old</li>
    <li class="list-item">19 years old</li>
    <li class="list-item">43 years old</li>
</ol>
```

Use the **myThing** variable from Question 4, and using **JavaScript**, append **three new list items** to this list with the ages specified in **myThing**. Your final list should look like this:

1. 15 years old
2. 19 years old
3. 43 years old
4. 26 years old
5. 18 years old
6. 30 years old

Hint:

- Use the .classList.add() method to apply the list-item class -> look it up for how to use.

6. **(10 marks)** You are going to use the Hexbot from https://noopschallenge.com/challenges/hexbot to create a beautiful dot painting on your page.

You will:

- Fetch 100 random colours using the Hexbot (read the documentation for how).
- Provide the width and height parameters to get coordinates returned to you also (300 will be fine for the width and height values).
- Iterate through the returned list, and create a new div for each colour. Apply the class **dot** and use the following CSS style for creating your 'dots'.

```
.dot{ padding:10px;position:absolute;border-radius:100% }
```

- Finally, as you iterate through your colours, you need to use **JavaScript** to apply the returned colour as the **background-color** style to your div.
- You also need to apply the returned coordinates (**x** and **y**) as the **left** and **top** properties of each div.
- Append each div to the body of the doc.
- Your final page should look similar to this:

Hint:

- You need to include "px" when you are assigning your **left** and **top** values -> you will need to concatenate this to the end of your values, e.g. 100px, 56px, etc, etc

7. **(2 marks)** The following code results in *undefined* being logged to the console:

```
let test = [
    { "message" : "The SBA is over!" }
];
console.log(test["message"]);
```

1) Why?
2) How can you fix the code to correctly log “The SBA is over!” to the console?