1. HTML

```
<button id="change-appearance">change
Appearance</button>
<p id="paragraph">Lorem ipsum dolor
<span class="highlight">sit amet</span>
consectutor adipsicing elit.</p>
```

CSS:

```
font size: 18px;
}
.highlight {
background-color: orange;
}
:highlight .bover {
background-color: orange;
}
.changed-appearance {
font-size: 24px;
colour: blue;
}
```

Java script:

```
const paragraph = document.getElementById('paragraph');
const button = document.getElementBuild('change-
                                        appearance');
button.addEventListener('click', () => {
paragraph.classList.toggle('changed-appearance');
});
```

2. HTML:

```
input id="num1" type="number">
input id="num2" type="number">
button id="check"> Check Equal
input id="word type:"text">
button id="search"> Search Word
```

pid = "result">

Javascript:

```
const num 1 =
document.get Element Byld ("num 1")
const num 2 = document.get Element Byid ("num 2");
Const check = document.get Element Byid ('check');
const word = document.get Element By id ('word');
Const search = document.get Element by id ('search');
const result = document.get Element by id ('result');
```

3. HTML:

```
<form id = "form">
Name : <input type = "text"
    id = "name">
Email : <input type = "email"
    id = "email">
<button id = "submit">
    Submit </button>
</form>
```

Javascript:

```
const form = document
get Element Byid ('form');
Const submit button =
document.get Element By id
        ('submit');
Submit Button.addevent
Listener ('click', (e) => {
const name =
document.get Email by id
('name').value;
Const email =
document.get Email By id
('email').value;
if (!name || !email) {
alert ("please fillout all
fields");
e.percent Default ();
}
});
```

4. HTML:

```
<div class="red" onmouseover="document.body.style.
background color="red"></div>
<div class="green" onmouse over=" document.body.style.
Background color="green"></div>
<div class="blue" onmouseover="document.body.style.
background color="Window blue"></div>
<button onclick="window.scrollBy(1,0):setTimeout.
(() => window.scrollBy(-1,0), 50)">Shake</button>
```

CSS:

```
.red {background-color: red; width: 50px; height: 150px;}
.green {background-color: green; width: 50px; height: 50px;}
.blue {background-color: blue; width: 50px; height: 50px;}
```

5. What is DOM?

Document ObjectModel is like a Map of an HTML document. It helps Javascript interact with the document. Creating HTML Elements Dynamically:

1. Create element: const element = document.createElement('tagName')
2. Add text: element.textContent = 'text'
3. Append to parent:

   parent.appendChild(element)

Traversing DOM:

1. document.getElementById('id')
2. document.getElementbyid('classname')

6. What are Regular Expressions?

Regular Expressions (regex) are patterns to match text.

Why use Regex in JavaScript?

1. Validate data (email, password)
2. Search text
3. Format input

password Validation Example:-

Regex pattern: ^(?=.*[A-Z])(?=.*\d).{8,}$,

- 8 + characters
- 1+ uppercase
- 1+ lowercase/number

3. Essential Regex Methods:

1. test() - check if pattern Matches
2. exec() - Find Match and Subgroups
3. Match() - Find Match and Subgroups.

7. HTML:

```
<button onclick="showText()">show</button>
<button onclick="hideText()">Hide</button>
<div id="text" style="display more;">Hidden Text
```

CSS:

```
.red { Background color: red; width: 50px; height:
       50px; }
.blue { Background-color: blue; width: 50px;
        height: 50px; }
```

8.

HTML:

```
<img src="image1.jpg" onclick="this.src=this.src=
image1.jpg?'image2.jpg':'image1.jpg">
<span onmouseover="this.style.color='red';this.style.
color='black',this.style.font-weight='Normal'">
Hoo over me! </span>
```

9.

HTML:

```
<input type="text" id="search" onkey="autoComplete()">
<div id="suggestions"></div>
```

JavaScript:

```
function autoComplete(){
const input = document.getElementbyid("search");
const query = input.value.trim();
if (query.length>=2){
fetch('/autocomplete?q='+query)
.then(response => response.json())
.then(data => {
document.getElementbyid("suggestions").intextHTML
=data.ajoin("");});}}
server-sideCode(Node.js);
const http = require('http');
http.create server((req,res) => {
const query = req.url.split("==")[1];
const suggestions = ['Apple','Google','Amazon'];
```

```
res.end(JSON.stringify(suggestions.filter(s=>s.includes
(query))); }).listen(3000);
```

10. HTML:

```
<input id="num" type="number">
<button onClick="calculate()">Calculate</button>
<p id="result"></p>
```

JavaScript:

```
function calculate() {
const num = num.value;
result.innerHTML = ${factorial(num)} | ${sum
    Digits(num)}
 | ${isPalindrome(num)};
}
function factorial(n) {
return n == 0 || 1;
}
function sumDigits(n)
return [...n].reduce((a,b) => parseInt(a) + parseInt
    (b), 0);
function isPalindrome(n){
  return n === [...n].reverse().join('') ? 'Yes' :
      No;
}
```

2. Tablet (480px - 767px)
3. Desktop (768px+)

Design Considerations:

1. Simpler Navigation
2. Clear Typography
3. Minimal Content

Tools:

1. Google Chrome Dev Tools
2. Mozilla Responsive Design Mode
3. Mobile-Friendly Test tool

---

Webpage Design

- Header
  - Content (text, images
  - Footer

HTML ELEMENTS.

- h1-h6 (headings)