Name : Ramanda Danny

Email : 21102313@ittelkom-pwt.ac.id

No. : 085888892380

1. JavaScript Framework and Vue.js

A JavaScript Framework is a collection of tools and functions that aid in developing web applications using JavaScript. Frameworks provide a structured way and built-in functionalities to ease the development of applications.

Vue.js is a lightweight and progressive JavaScript framework used for building interactive user interfaces. It allows developers to easily create complex web applications using concepts like data binding, components, and reactivity.

Example using Vue.js:

```
<!DOCTYPE html>
<html>
<head>
  <title>Vue.js Example</title>
  <script src="https://cdn.jsdelivr.net/npm/vue@2.6.14/dist/vue.js"></script>
</head>
<body>
  <div id="app">
    <p>{{ message }}</p>
  </div>
  <script>
    var app = new Vue({
      el: '#app',
      data: {
        message: 'Hello, this is an example of using Vue.js!'
      }
    });
  </script>
</body>
</html>
```

2. Use of Ellipsis

Ellipsis (the three-dot symbol) is used to indicate that displayed text or content is truncated and not fully shown. It is generally used to indicate that there is more content not being displayed and may be accessible in some way (e.g., on click).

CSS example to use ellipsis on truncated text:

```
.ellipsis-text {
  white-space: nowrap;
  overflow: hidden;
  text-overflow: ellipsis;
  width: 150px; /* Set width as needed */
}
```

3. Animation CSS Properties

a. @keyframes

Used to define animations by specifying style changes at specific points in time during the animation.

Example usage:

```
@keyframes example {
  0%   { opacity: 0; }
  50%  { opacity: 0.5; }
  100% { opacity: 1; }
}
```

b. animation-name

Specifies the name of the @keyframes to be used for the animation.

Example usage:

```
.element {
  animation-name: example;
}
```

c. animation-duration

Specifies the duration of the animation in a specified time unit (e.g., seconds or milliseconds).

Example usage:

```
.element {
  animation-duration: 2s;
}
```

d. animation-iteration

Specifies the number of times the animation will repeat.

Example usage:

```
.element {
  animation-iteration: 3; /* Set the number of iterations */
}
```

e. animation-direction

Specifies the direction of the animation (forward, backward, alternate, alternate-reverse).

Example usage:

```
.element {
  animation-direction: alternate;
}
```

4. Lazy Load in JavaScript

Lazy loading is a technique where a web application loads content only when it's needed. In the case of images, for example, images are loaded only when a user scrolls to the area where the image appears.

Implementation example of lazy loading with JavaScript:

```
<img data-src="lazyload-image.jpg" class="lazyload" width="200" height="200">
<script>
  document.addEventListener("DOMContentLoaded", function() {
    var lazyloadImages = document.querySelectorAll(".lazyload");
    var lazyloadThrottleTimeout;

    function lazyload() {
      if (lazyloadThrottleTimeout) {
        clearTimeout(lazyloadThrottleTimeout);
      }
      lazyloadThrottleTimeout = setTimeout(function() {
```

```
    var scrollTop = window.pageYOffset;

    lazyloadImages.forEach(function(img) {

     if (img.offsetTop < (window.innerHeight + scrollTop)) {

      img.src = img.dataset.src;

     }

    });

   }, 20);

  }

  window.addEventListener('scroll', lazyload);

  window.addEventListener('resize', lazyload);

  window.addEventListener('orientationChange', lazyload);

 });

</script>
```

5. Git Commands and Their Functions

Git is a version control system used to manage changes in a software project. Here are some Git commands and their functions:

- `git init`: Initializes a new Git repository.

- `git add <file>`: Adds changes in a file to the staging area.

- `git commit -m "commit message"`: Creates a snapshot of the staging area and saves it to the repository.

- `git push`: Sends local changes to the remote repository.

- `git pull`: Fetches changes from the remote repository to the local repository.

As a student seeking an internship, having an understanding of these concepts and practical skills in using Git, JavaScript, and understanding basic concepts like CSS animations will greatly assist in handling day-to-day tasks in software development projects.