In Angular, two concepts are related to loading images: `loading` and `lazy loading`. Additionally, Angular provides a way to specify the `priority` for loading images. Let's explore each concept:

1. **Loading**: The `loading` attribute in Angular allows you to specify how the browser should handle the loading of an image. It can take three values:
    - `auto` (default): The browser chooses the best strategy for loading the image.
    - `eager`: The image is loaded immediately, regardless of whether it is currently visible on the screen.
    - `lazy`: The image is loaded only when it becomes visible in the viewport, i.e., when it is about to be displayed on the screen.
2. The `loading` attribute is set on the `<img>` tag and can be used to optimize the loading of images, especially when dealing with large or numerous images.

```
   <img ngSrc="cat.jpg" width="400" height="200"
loading="eager">
```

3. **Lazy Loading**: Lazy loading is a technique used to defer the loading of non-critical resources, such as images until they are needed. In the context of Angular, lazy loading typically refers to loading images only when they become visible in the viewport. This can significantly improve the initial page load time and reduce the amount of data transferred.
4. **Priority for Images**: In Angular, you can set the `priority` for loading images using the `loading` attribute.

```
   <img width="400" height="400" priority ngSrc="pic1.jpg">
```

   The `priority` attribute allows you to specify the importance of an image in terms of loading priority. For example, you might want to give higher priority to images that are critical for the initial rendering of the page, while deferring the loading of less important images. The exact behavior and implementation of the `priority` attribute can vary depending on the library or custom solution you are using.

   By default, `NgOptimizedImage` sets `loading=lazy` for all images that are not marked `priority`.