

# Lazy loading

Last Modified on 10/21/2021 11:58 am EDT

Lazy loading is one more way for publishers to optimize their websites. Rather than immediately loading all ads on a page when it's opened, a publisher can build triggers into the code that prevent ads from loading until a certain condition is met (for example, when a user scrolls to a specific part of the page).

**Note:** The method described in this article does not include a mechanism for measuring scroll positions or other on-page events to trigger lazy loading. These instructions only cover the actions that you should take to implement a lazy-loaded ad unit.

## Implement lazy loading on initial page load

1. Change the class on the tags from the Freestar system from "ad-tag" to "no-ad" to turn them into placeholder div tags.
2. To make it easier to find the correct div later, you may want to add a unique id to the class name, for example, "no-ad-1".
3. Add the placeholder divs for the ads in the location you would like the ads to appear.

## Example

A publisher can take a regular looking div and change it so it won't be found.

- Regular div:

```
<div class="ad-tag"></div>
```

- Lazy load div:

```
<div class="no-ad-1"></div>
```

## Implement lazy loading on page scroll

Once the user scrolls down to the point where you want to trigger the ad load, change the class from "no-ad-1" to "ad-tag" on the div(s) that you want to load, and then call the `deployads.push()` function.

## Example

- Before user scrolls:

```
<div class="no-ad-1"></div>
```

- After user scrolls:

```
<div class="ad-tag"></div>
```

To actually trigger the ad to load, execute our push function:



You can add additional classes and parent elements to assist in the locating and targeting of the ad units.

For industry insights and information about our product offerings, [check out our blog!](#)

Want to see our products in action? For a demo, fill out a form [here](#).