Skeleton Loader Component | WeWeb Documentation

|  |  |
| --- | --- |
| URL source | <https://docs.weweb.io/elements/skeleton-loader.html> |
| Date de scraping | 2025-04-08T15:54:32.284213+00:00 |

## Skeleton Loader ​

The Skeleton Loader component provides visual placeholders that mimic your content's layout during loading states. These animated placeholders improve perceived performance and reduce user frustration by indicating that content is loading.

## Getting Started ​

## Using AI ​

The quickest way to set up skeleton loaders is by using AI:  
First, think about which content areas you want to show loading states for Ask AI: "Create skeleton loaders for my user profile page" or "Add skeleton loaders to my product cards" Continue refining your skeleton loaders by asking AI for specific modifications: "Make the skeleton loaders use the pulse animation" "Create a skeleton layout for my article list with title and description" "Add rounded corners to match my card design" "Create a skeleton for a circular profile picture with two text lines"

## Manual Setup ​

Add a Skeleton Loader component to your page using the Add panel (search for "skeleton" or "loader") Style the component to match the dimensions and shape of your content (width, height, border radius) Choose an animation type that matches your design aesthetic

## Animation Types ​

The Skeleton Loader component offers three animation types via the Animation Type property:  
Wave - A subtle highlight that moves from left to right across the element  
Pulse - A gentle fading in and out effect  
Shimmer - A light reflection that sweeps across the element

## Styling Your Skeleton Loader ​

For the Skeleton Loader to be effective, it's essential to style it properly using the standard styling properties. Here are the key properties to consider:

## Essential Styling Properties ​

## Covering Existing Content ​

To create a loading state that covers existing content (like product images or cards that are already rendered):  
Place the Skeleton Loader in the same container as the elements you want to cover Give the Skeleton Loader a position of Absolute , with a top position of 0 and left position of 0 Set the width and height of the Skeleton Loader to 100% to fully cover the content Set the z-axis value of the Skeleton Loader to be higher than the contents you want to cover  
What these positioning properties do:

### Images

<https://docs.weweb.io/assets/skeleton-loader-card-example.nosG4sU9.gif>

## Conditionally Showing Skeleton Loaders ​

To show Skeleton Loaders only during loading states:  
Create a boolean variable (e.g., isLoading ) to use for controlling the display  
  
Bind the Skeleton Loader's display to this variable  
  
Set up logic to appropriately change the value of this variable when needed  
  
Once everything is set up, the Skeleton Loader will display only during loading, like this:

### Images

<https://docs.weweb.io/assets/skeleton-loader-create-var.DvZb6rDv.png>

<https://docs.weweb.io/assets/skeleton-loader-bind-display.BrnRkEJy.png>

<https://docs.weweb.io/assets/skeleton-loader-fetch-workflow.BRN5v9jB.png>

<https://docs.weweb.io/assets/skeleton-loader-example-fetch.B5dIlPIp.gif>

## Usage Examples ​

## Text Content Loader ​

To create a placeholder for text content:  
Place the Skeleton Loader in a container with the text element Give the Skeleton Loader a position of Absolute , with a top position of 0 and left position of 0 Set the width and height of the Skeleton Loader to 100% to fully cover the text Set the z-axis value of the Skeleton Loader to be higher than that of the text element

### Images

<https://docs.weweb.io/assets/skeleton-loader-text-example.BVqnN69U.gif>

## Card Loader ​

To create a placeholder for multiple elements in a card:  
Place the Skeleton Loader in a container with one of the desired elements to cover Give the Skeleton Loader a position of Absolute , with a top position of 0 and left position of 0 Set the width and height of the Skeleton Loader to 100% to fully cover the text Set the z-axis value of the Skeleton Loader to be higher than that of the card or image Duplicate the Skeleton Loader and place in containers with all needed elements

### Images

<https://docs.weweb.io/assets/skeleton-loader-card-example.nosG4sU9.gif>

<https://docs.weweb.io/assets/skeleton-loader-group-shortcut.DMwU6JqR.gif>

## Best Practices ​

## Do: ​

Match the skeleton's dimensions and shape to your actual content Use subtle, consistent animations across your interface Group related skeleton elements to mirror your content structure Display skeletons immediately when loading begins

## Don't: ​

Use bright or high-contrast colors for skeleton elements Mix different animation types in the same loading context Make animations too fast or distracting Keep skeletons visible for too long (aim for <2 seconds when possible)

## Common Use Cases ​

Article or blog post loading screens Product card loading in e-commerce Social media feed item placeholders Profile information loading Form field placeholders during data retrieval Dashboard widget loading states

## Forking ​

While the Skeleton Loader component offers flexibility through its animation types and styling options, there may be cases where you need functionality beyond what's natively available. In these situations, you can fork the component and modify it to meet your specific requirements.  
If you are unsure how to fork an element, you can learn more in the dedicated documentation .

## Forking Example: Custom Animation Type ​

One common reason to fork the Skeleton Loader is to create a specialized animation type not available in the standard component. For example, you might want to create a "Bounce" animation that adds a subtle up-and-down movement.  
Here's how you can use AI to help with this process:

### Code

Langage: unknown

Modify this forked Skeleton Loader to add a 'Bounce' animation type that adds a subtle vertical movement effect

## Forking Example: Changing Animation Speed ​

Another useful customization is having the ability to control the speed of the animation.  
Here's an example prompt for this:

### Code

Langage: unknown

Update this forked Skeleton Loader to add a property for controlling the animation speed

## Properties ​

## Settings ​