Skeleton Mammoth.

Or how I've been solving the problem of reusable skeleton loaders.

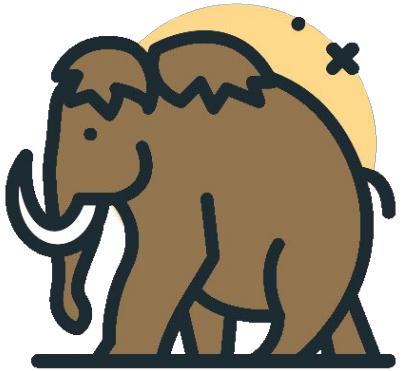
Oleksandr Tkachenko

About me:

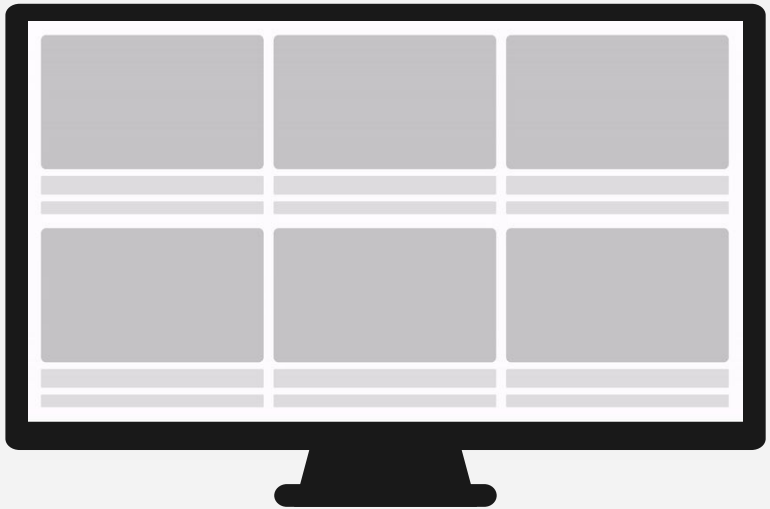
Oleksandr Tkachenko

- 5 years of experience in IT.
- Author of technical and scientific articles.
- Hackathons refereeing.
- Conferences speaker.
- Creator of the "**Skeleton Mammoth**" open-source CSS library.

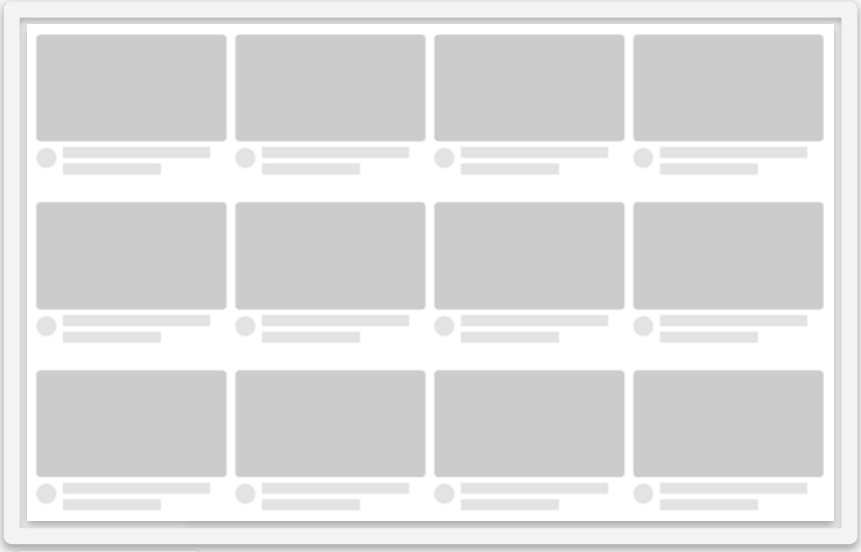
Introduction.



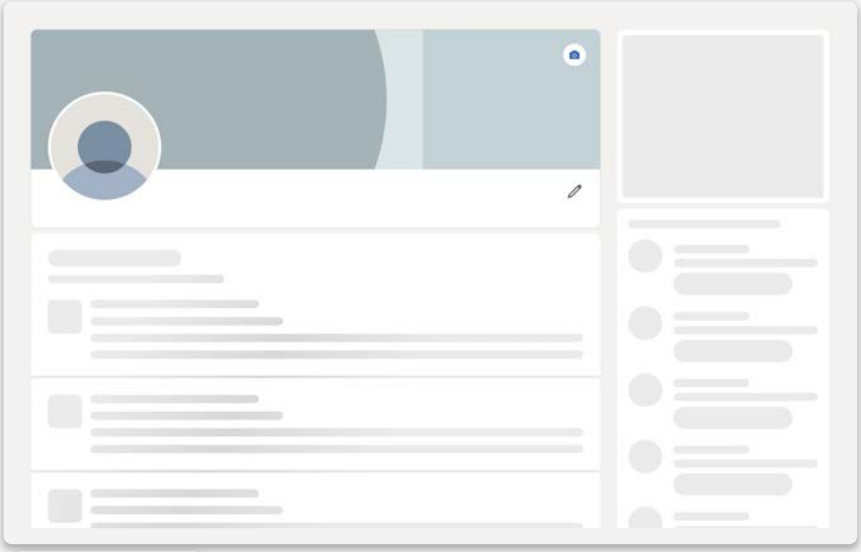
What are skeleton loaders?



Skeleton loaders examples.



YouTube.

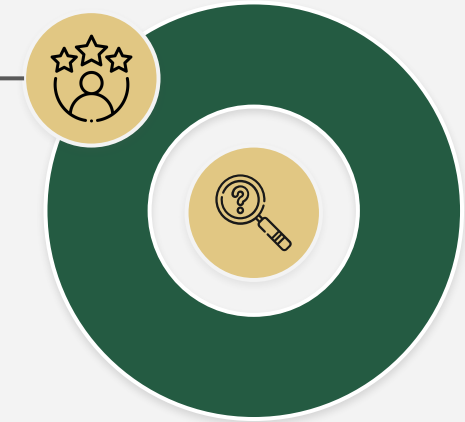


LinkedIn.

Why do you have to use skeleton loaders?

Why do you have to use skeleton loaders?

Improve user experience.



Why do you have to use skeleton loaders?

Improve user experience.

Reduce bounce rate.

Why do you have to use skeleton loaders?

Improve user experience.

Reduce bounce rate.

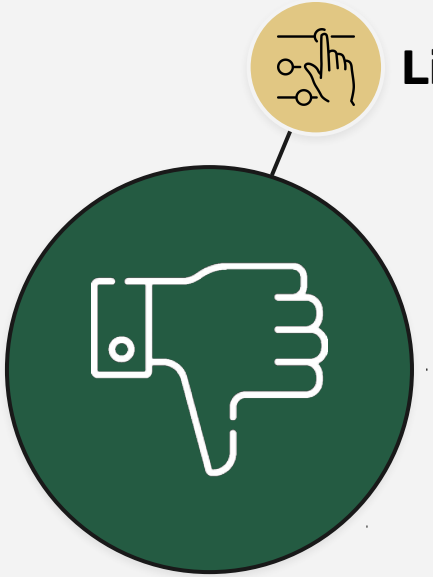
Smooth transitions.

Why do you have to use skeleton loaders?

- Improve user experience.
- Reduce bounce rate.
- Smooth transitions.
- Focus attention.

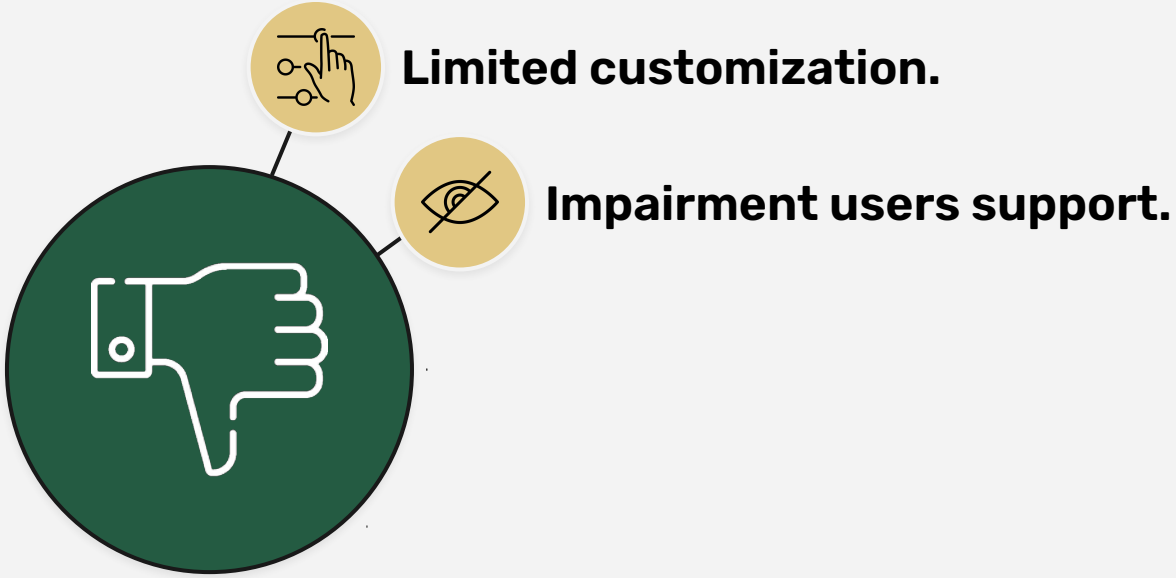
Problems of most existing skeletons.

Problems of most existing skeletons.



Limited customization.

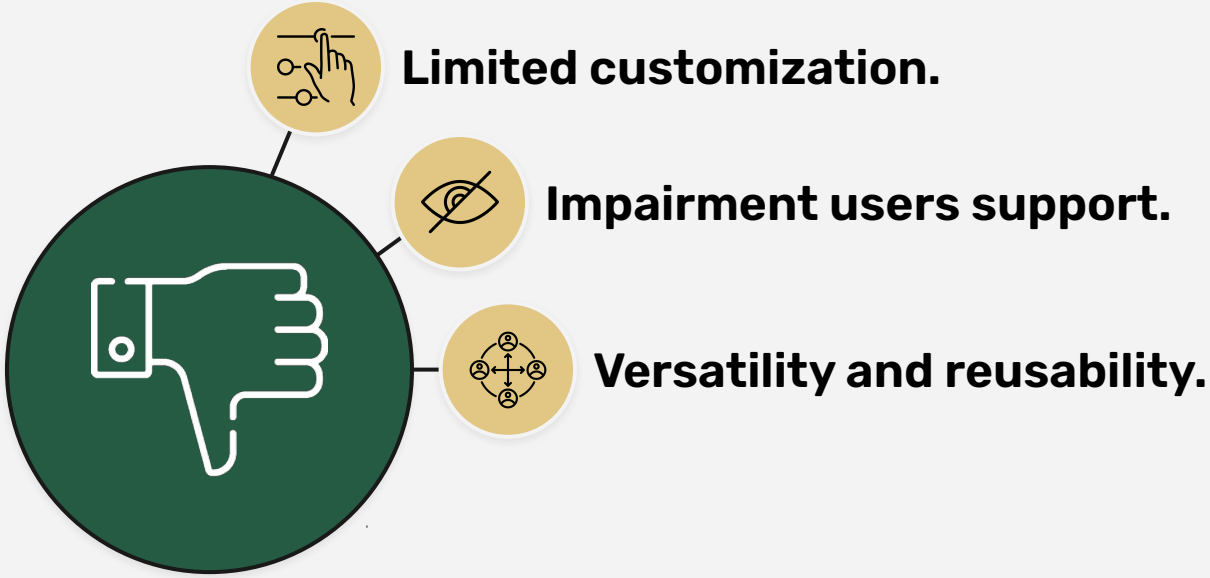
Problems of most existing skeletons.



Limited customization.

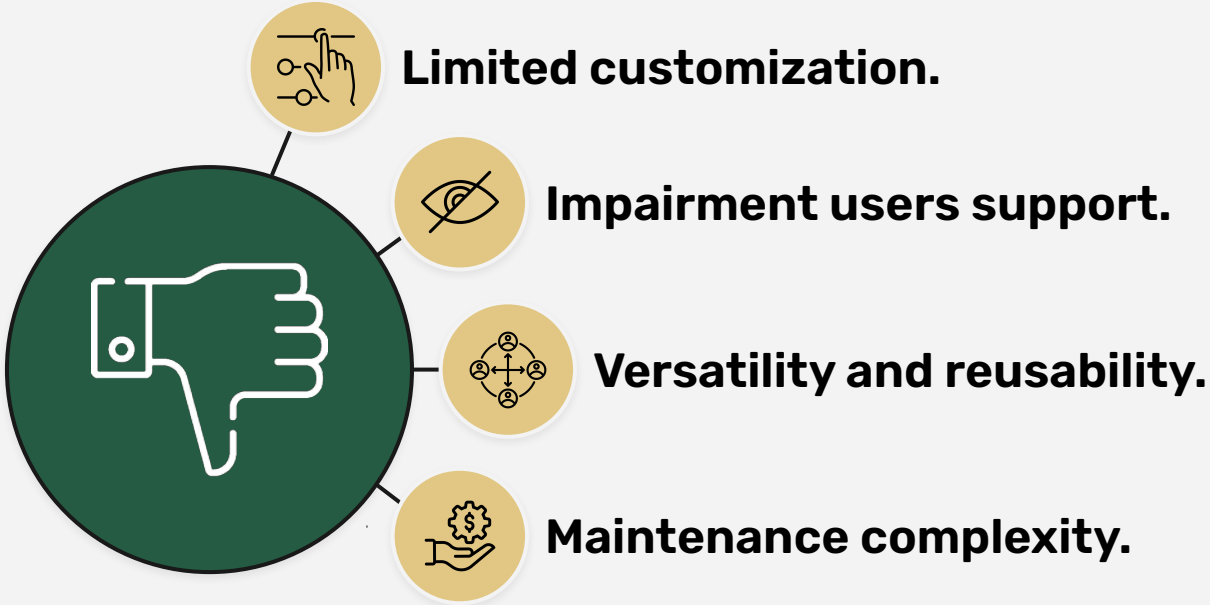
Impairment users support.

Problems of most existing skeletons.

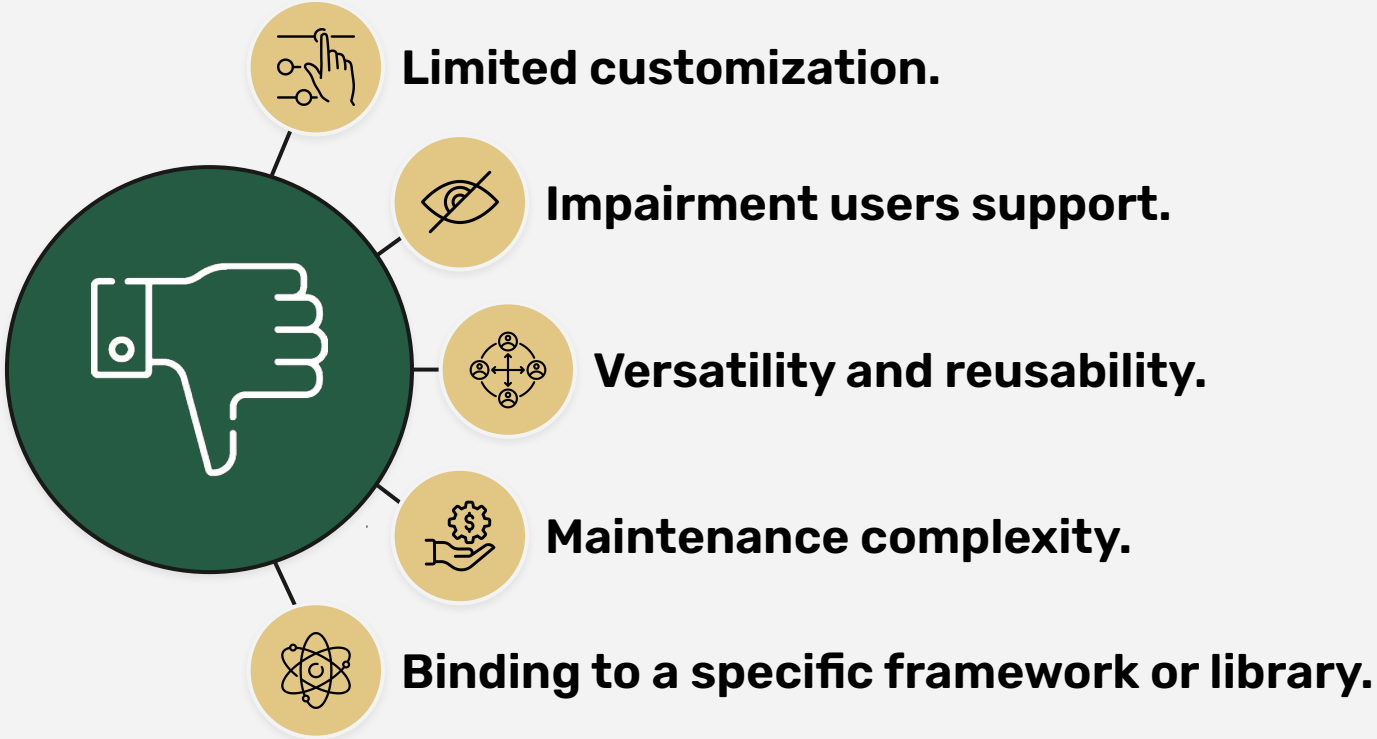


- Limited customization.
- Impairment users support.
- Versatility and reusability.

Problems of most existing skeletons.

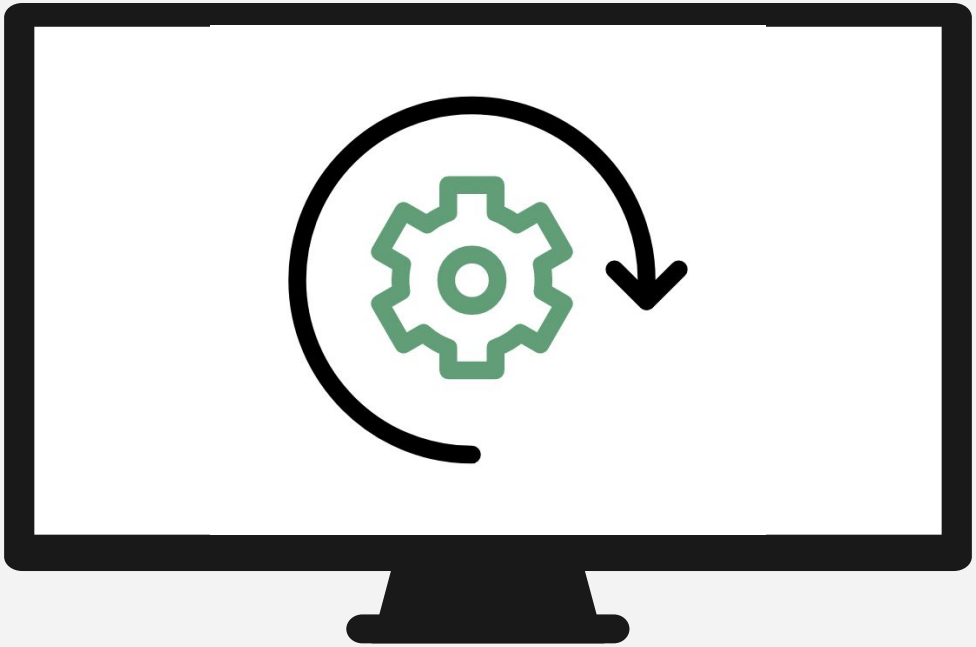


Problems of most existing skeletons.



Skeleton loader alternatives.

Skeleton loader alternatives.

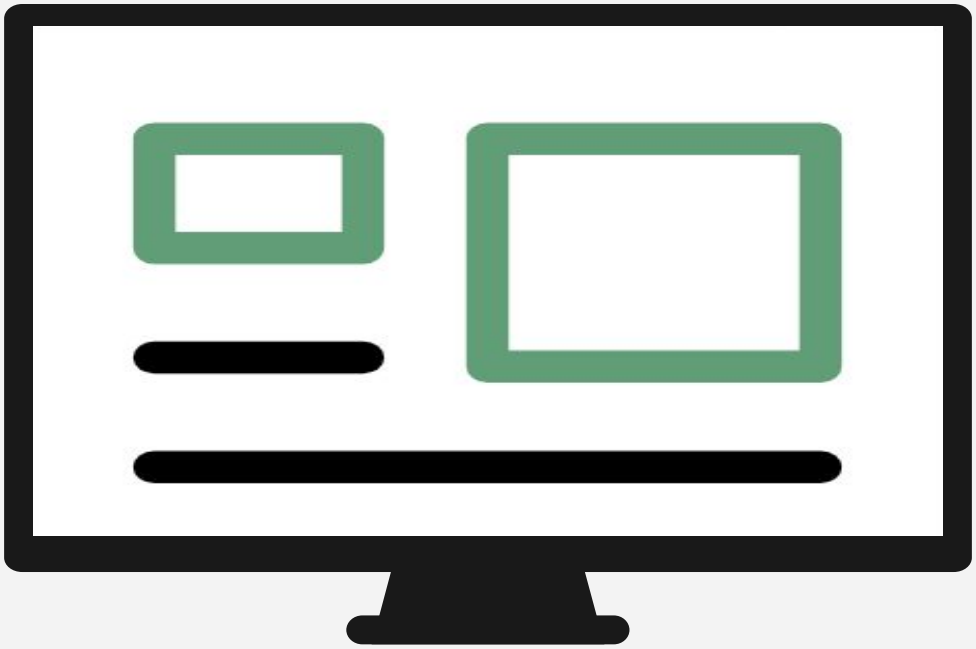


Spinner.

Skeleton loader alternatives.

Progress Bar.

Skeleton loader alternatives.



The absence of any visual.

Creating versatile and reusable skeleton loader.

- Versatile and reusable.
- Configuration flexibility.
- Feature-rich.
- Lightweight and dependencies-free.

Creating versatile and reusable skeleton loader.

BaseCard.jsx

```jsx
<div className="card">
  <div className="card__img-wrapper">
    <img className="card__img" src={require(`../../images/cards/${imgUrl}`)} />
  </div>
  <div className="card__body">
    <div className="card__details">
      <p className="card__title">{title}</p>
      <p className="card__subtitle">{subtitle}</p>
    </div>
  </div>
</div>
```

Creating versatile and reusable skeleton loader.

BaseCard.jsx

```jsx
<div className={`card ${dataState.dataStatus === "loading" ? "sm-loading" : ""}`}>
    <div className="card__img-wrapper sm-item-primary">
        <img className="card__img" src={require(`../../images/cards/${imgUrl}`)} />
    </div>
    <div className="card__body">
        <div className="card__details">
            <p className="card__title sm-item-secondary">{title}</p>
            <p className="card__subtitle sm-item-secondary">{subtitle}</p>
        </div>
    </div>
</div>
```

Creating versatile and reusable skeleton loader.

Root variables.
Colors.

colors.scss

```scss
/* Root variables.
Needed here to be able to override them after compilation:
https://github.com/WOLFRIEND/skeleton-mammoth#overriding-styles-with-global-variables
-------------------------------------------------------------------------------*/
:root {
  /* Light theme colors. */
  --sm-color-light-primary: 204, 204, 204, 1;
  --sm-color-light-secondary: 227, 227, 227, 1;
  --sm-color-light-animation-primary: color-mix(in srgb, #fff 15%, rgba(var(--sm-color-light-primary)) 85%);
  --sm-color-light-animation-secondary: color-mix(in srgb, #fff 15%, rgba(var(--sm-color-light-secondary)) 85%);

  /* Dark theme colors. */
  --sm-color-dark-primary: 37, 37, 37, 1;
  --sm-color-dark-secondary: 41, 41, 41, 1;
  --sm-color-dark-animation-primary: color-mix(in srgb, #fff 2%, rgba(var(--sm-color-dark-primary)) 98%);
  --sm-color-dark-animation-secondary: color-mix(in srgb, #fff 2%, rgba(var(--sm-color-dark-secondary)) 98%);
}

/* Light theme colors. */
$--sm-color-light-primary: var(--sm-color-light-primary);
$--sm-color-light-secondary: var(--sm-color-light-secondary);
$--sm-color-light-animation-primary: var(--sm-color-light-animation-primary);
$--sm-color-light-animation-secondary: var(--sm-color-light-animation-secondary);

/* Dark theme colors. */
$--sm-color-dark-primary: var(--sm-color-dark-primary);
$--sm-color-dark-secondary: var(--sm-color-dark-secondary);
$--sm-color-dark-animation-primary: var(--sm-color-dark-animation-primary);
$--sm-color-dark-animation-secondary: var(--sm-color-dark-animation-secondary);
```

Creating versatile and reusable skeleton loader.

Root variables.
Animations.

animations.scss

```scss
/* Root variables.
Needed here to be able to override them after compilation:
https://github.com/WOLFRIEND/skeleton-mammoth#overriding-styles-with-global-variables
------------------------------------------------------------------------------*/
:root {
  /* Animations values. */
  --sm-animation-duration: 1.5s;
  --sm-animation-timing-function: linear;
  --sm-animation-iteration-count: infinite;

  /* Animations. */
  --sm-animation-none: none;
  --sm-animation-wave: --sm--animation-wave var(--sm-animation-duration) var(--sm-animation-timing-function)
    var(--sm-animation-iteration-count);
  --sm-animation-wave-reverse: --sm--animation-wave-reverse var(--sm-animation-duration)
    var(--sm-animation-timing-function) var(--sm-animation-iteration-count);
  --sm-animation-pulse: --sm--animation-pulse var(--sm-animation-duration) var(--sm-animation-timing-function)
    var(--sm-animation-iteration-count);
}

/* Animations values. */
$--sm-animation-duration: var(--sm-animation-duration);
$--sm-animation-timing-function: var(--sm-animation-timing-function);
$--sm-animation-iteration-count: var(--sm-animation-iteration-count);
$--sm-animation-wave-background-position-x: -200%;
$--sm-animation-wave-reverse-background-position-x: 200%;
$--sm-animation-pulse-percentage-0: 1;
$--sm-animation-pulse-percentage-50: 0.6;
$--sm-animation-pulse-percentage-100: 1;

/* Animations. */
$--sm-animation-none: var(--sm-animation-none);
$--sm-animation-wave: var(--sm-animation-wave);
$--sm-animation-wave-reverse: var(--sm-animation-wave-reverse);
$--sm-animation-pulse: var(--sm-animation-pulse);
```

Creating versatile and reusable skeleton loader.

Base styles.

base-styles.scss

```scss
.sm-loading {
  .sm-item-primary,
  .sm-item-secondary {
    border-color: transparent !important;
    color: transparent !important;
    cursor: wait;
    outline: none;
    position: relative;
    user-select: none;
```

base-styles.scss

```scss
    &:before {
      clip: rect(1px, 1px, 1px, 1px);
      content: "Loading, please wait.";
      inset: 0;
      overflow: hidden;
      position: absolute;
      white-space: nowrap;
    }
```

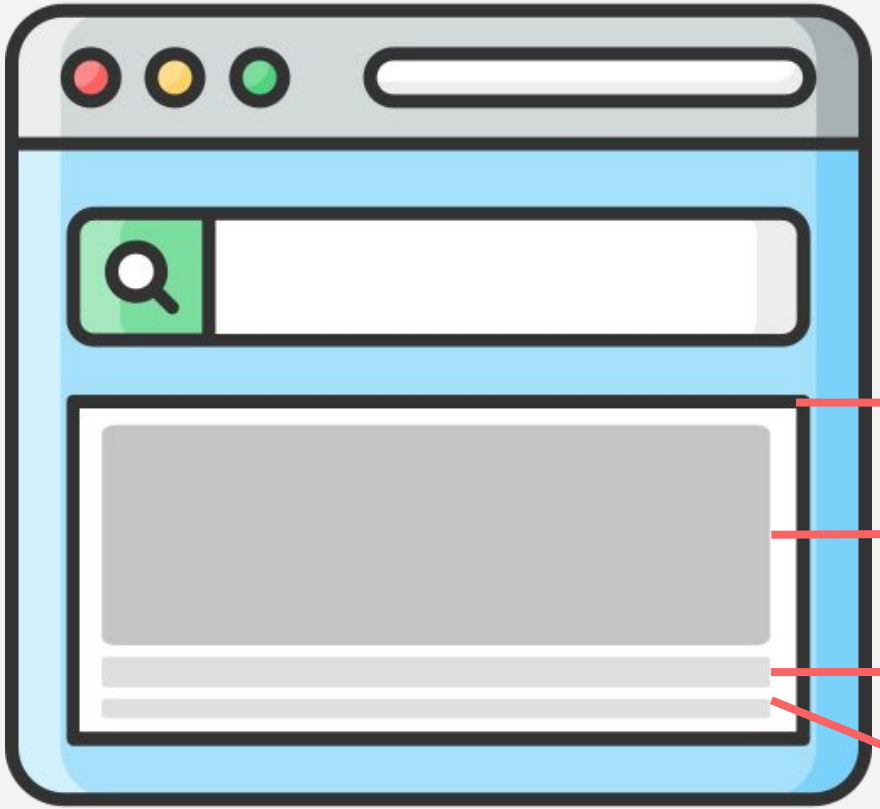
base-styles.scss

```scss
    &::placeholder {
      color: transparent !important;
    }

    * {
      visibility: hidden;
    }

    &:empty:after {
      content: "\00a0";
    }
  }
}
```

Creating versatile and reusable skeleton loader.



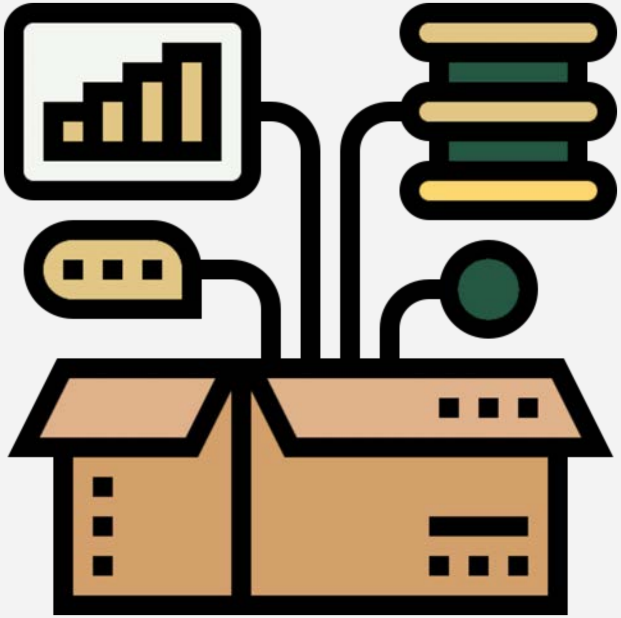
```
<div class="card sm-loading">
```

```
<div class="card__img-wrapper sm-item-primary">
```

```
<div class="card__title sm-item-secondary">
```

```
<div class="card__subtitle sm-item-secondary">
```

Advanced features.



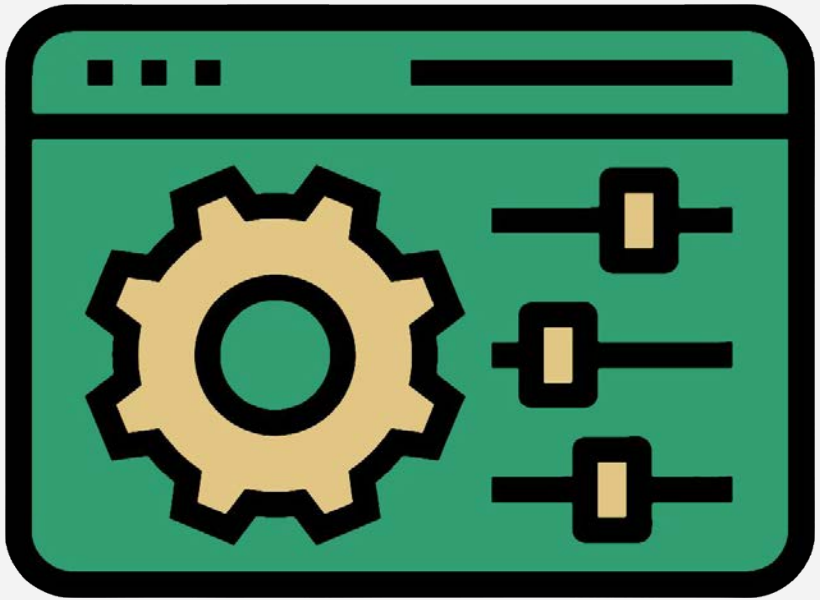
Color scheme.

```
@media (prefers-color-scheme: dark) {
    /* Dark color scheme styles .*/
}
```

Reduced motion.

```
@media (prefers-reduced-motion) {
    /* Animations styles .*/
}
```

Configuration.



BaseCard.jsx

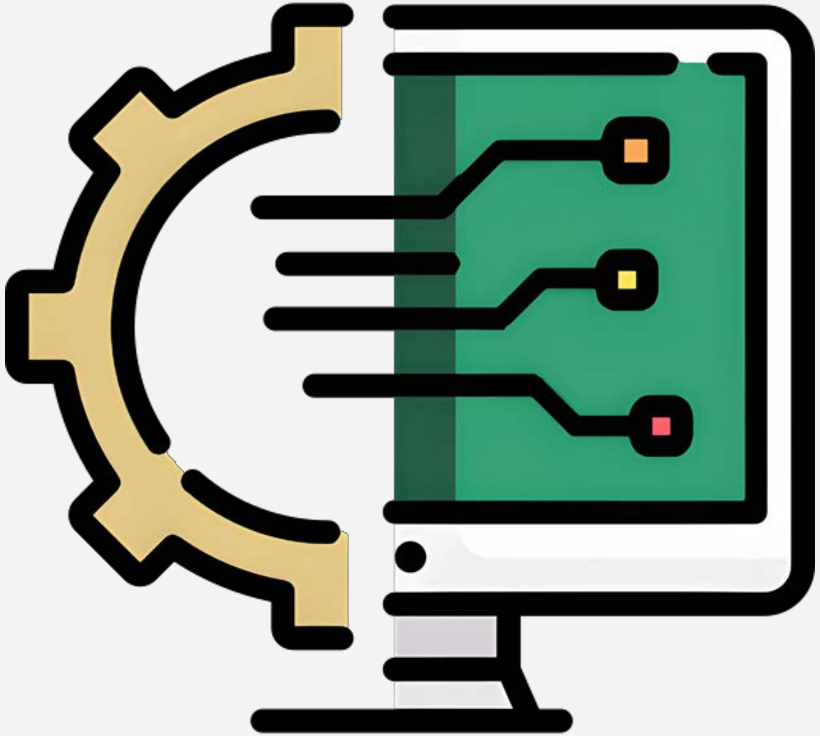
```jsx
const config = JSON.stringify({
  animation: "wave",
  theme: "dark",
  opacity: "0.7",
})
```

BaseCard.jsx

```jsx
<div className="card sm-loading" data-sm-config={config}>
    {/* Omitted pieces of code. */}
</div>
```

BaseCard.jsx
1 const config = JSON.stringify({
2 animation: "wave",
skeleton-mammoth.scss
1 .sm-loading[data-sm-config*='"theme":"dark"'] .sm-item-primary,
2 .sm-loading[data-sm-config*='"theme":"dark"'] .sm-item-secondary {
3 /* Omitted pieces of code. */
4 }
2 {/* Omitted pieces of code. */}
3

Advanced usage.



Demo.

Demo: main flow.

Demo: data loading.

Demo: color scheme.

Demo: animations.

Demo: opacity.

Let's wrap it up!

Thanks.

Scan for useful links.

Oleksandr Tkachenko