

deck.js

Modern HTML Presentations

[Download](#) or [→](#) to learn more.

How Does It Work?

Use arrow keys to navigate

How Does It Work?

Slides are basic HTML.

```
<div class="deck-container">
  <section class="slide">
    <h1>My Presentation</h1>
  </section>

  <section class="slide">
    <h2>Slide Header</h2>
    <p>Here is a list of points:</p>
    <ul>
      <li>Point 1</li>
      <li>Point 2</li>
      <li>Point 3</li>
    </ul>
  </section>

  <section class="slide">
    <h2>Another Slide</h2>
    <blockquote cite="http://example.com">
      <p>Lorem ipsum dolor sit amet.</p>
      <p><cite>Cicero</cite></p>
    </blockquote>
  </section>
</div>
```

How Does It Work?

Slides are basic HTML.

The [deck.core](#) module keeps track of slide states

```
<div class="deck-container">
  <section class="slide deck-previous">
    <h1>My Presentation</h1>
  </section>

  <section class="slide deck-current">
    <h2>Slide Header</h2>
    <p>Here is a list of points:</p>
    <ul>
      <li>Point 1</li>
      <li>Point 2</li>
      <li>Point 3</li>
    </ul>
  </section>

  <section class="slide deck-next">
    <h2>Another Slide</h2>
    <blockquote cite="http://example.com">
      <p>Lorem ipsum dolor sit amet.</p>
      <p><cite>Cicero</cite></p>
    </blockquote>
  </section>
</div>
```

How Does It Work?

Slides are basic HTML.

The [deck.core](#) module keeps track of slide states and deck states

```
<div class="deck-container on-slide-1">
  <section class="slide deck-previous">
    <h1>My Presentation</h1>
  </section>

  <section class="slide deck-current">
    <h2>Slide Header</h2>
    <p>Here is a list of points:</p>
    <ul>
      <li>Point 1</li>
      <li>Point 2</li>
      <li>Point 3</li>
    </ul>
  </section>

  <section class="slide deck-next">
    <h2>Another Slide</h2>
    <blockquote cite="http://example.com">
      <p>Lorem ipsum dolor sit amet.</p>
      <p><cite>Cicero</cite></p>
    </blockquote>
  </section>
</div>
```

How Does It Work?

Slides are basic HTML.

The [deck.core](#) module keeps track of slide states and deck states, leaving CSS to define what each state look like...

```
<div class="deck-container on-slide-1">
  <section class="slide deck-previous">
    <h1>My Presentation</h1>
  </section>

  <section class="slide deck-current">
    <h2>Slide Header</h2>
    <p>Here is a list of points:</p>
    <ul>
      <li>Point 1</li>
      <li>Point 2</li>
      <li>Point 3</li>
    </ul>
  </section>

  <section class="slide deck-next">
    <h2>Another Slide</h2>
    <blockquote cite="http://example.com">
      <p>Lorem ipsum dolor sit amet.</p>
      <p><cite>Cicero</cite></p>
    </blockquote>
  </section>
</div>
```

How Does It Work?

Slides are basic HTML.

The [deck.core](#) module keeps track of slide states and deck states, leaving CSS to define what each state look like...

...and how to transition between them.

```
<div class="deck-container on-slide-1">
  <section class="slide deck-previous">
    <h1>My Presentation</h1>
  </section>

  <section class="slide deck-current">
    <h2>Slide Header</h2>
    <p>Here is a list of points:</p>
    <ul>
      <li>Point 1</li>
      <li>Point 2</li>
      <li>Point 3</li>
    </ul>
  </section>

  <section class="slide deck-next">
    <h2>Another Slide</h2>
    <blockquote cite="http://example.com">
      <p>Lorem ipsum dolor sit amet.</p>
      <p><cite>Cicero</cite></p>
    </blockquote>
  </section>
</div>
```

How Does It Work?

Slides are basic HTML.

The [deck.core](#) module keeps track of slide states and deck states, leaving CSS to define what each state look like...

...and how to transition between them.

[Extensions](#) use core [events](#) and [methods](#) to add goodies

```
<div class="deck-container on-slide-1">
  <section class="slide deck-previous">
    <h1>My Presentation</h1>
  </section>

  <section class="slide deck-current">
    <h2>Slide Header</h2>
    <p>Here is a list of points:</p>
    <ul>
      <li>Point 1</li>
      <li>Point 2</li>
      <li>Point 3</li>
    </ul>
  </section>

  <section class="slide deck-next">
    <h2>Another Slide</h2>
    <blockquote cite="http://example.com">
      <p>Lorem ipsum dolor sit amet.</p>
      <p><cite>Cicero</cite></p>
    </blockquote>
  </section>
</div>
```


How Does It Work?

Slides are basic HTML.

The `deck.core` module keeps track of slide states and deck states, leaving CSS to define what each state look like...

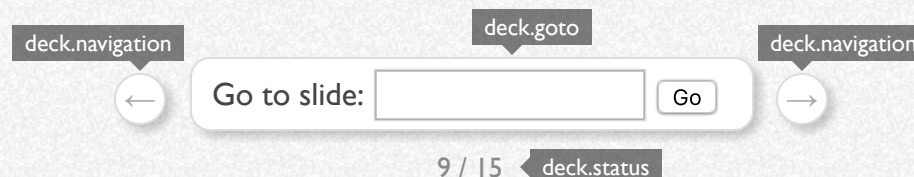
...and how to transition between them.

Extensions use core **events** and **methods** to add goodies, giving presenters the freedom to add what they want

```
<div class="deck-container on-slide-1">
  <section class="slide deck-previous">
    <h1>My Presentation</h1>
  </section>

  <section class="slide deck-current">
    <h2>Slide Header</h2>
    <p>Here is a list of points:</p>
    <ul>
      <li>Point 1</li>
      <li>Point 2</li>
      <li>Point 3</li>
    </ul>
  </section>

  <section class="slide deck-next">
    <h2>Another Slide</h2>
    <blockquote cite="http://example.com">
      <p>Lorem ipsum dolor sit amet.</p>
      <p><cite>Cicero</cite></p>
    </blockquote>
  </section>
</div>
```



How Does It Work?

Slides are basic HTML.

The [deck.core](#) module keeps track of slide states and deck states, leaving CSS to define what each state look like...

...and how to transition between them.

[Extensions](#) use core [events](#) and [methods](#) to add goodies, giving presenters the freedom to add what they want and leave out what they don't.

```
<div class="deck-container on-slide-1">
  <section class="slide deck-previous">
    <h1>My Presentation</h1>
  </section>

  <section class="slide deck-current">
    <h2>Slide Header</h2>
    <p>Here is a list of points:</p>
    <ul>
      <li>Point 1</li>
      <li>Point 2</li>
      <li>Point 3</li>
    </ul>
  </section>

  <section class="slide deck-next">
    <h2>Another Slide</h2>
    <blockquote cite="http://example.com">
      <p>Lorem ipsum dolor sit amet.</p>
      <p><cite>Cicero</cite></p>
    </blockquote>
  </section>
</div>
```

Great. Now What?



Great. Now What?

Only Know HTML?

Use included premade themes and templates to start making your deck immediately.

[Get Started](#)

Great. Now What?

Only Know HTML?

Use included premade themes and templates to start making your deck immediately.

[Get Started](#)

CSS+JS Badass?

Make custom decks with the API exposed by the core and extensions.

[View Docs](#)

Great. Now What?

Only Know HTML?

Use included premade themes and templates to start making your deck immediately.

[Get Started](#)

Something In-between?

Take a little from both worlds. Tweak a theme, hack an extension, or write some of your own.

CSS+JS Badass?

Make custom decks with the API exposed by the core and extensions.

[View Docs](#)

Great. Now What?

Only Know HTML?

Use included premade themes and templates to start making your deck immediately.

[Get Started](#)

Something In-between?

Take a little from both worlds. Tweak a theme, hack an extension, or write some of your own.

CSS+JS Badass?

Make custom decks with the API exposed by the core and extensions.

[View Docs](#)[Download](#)