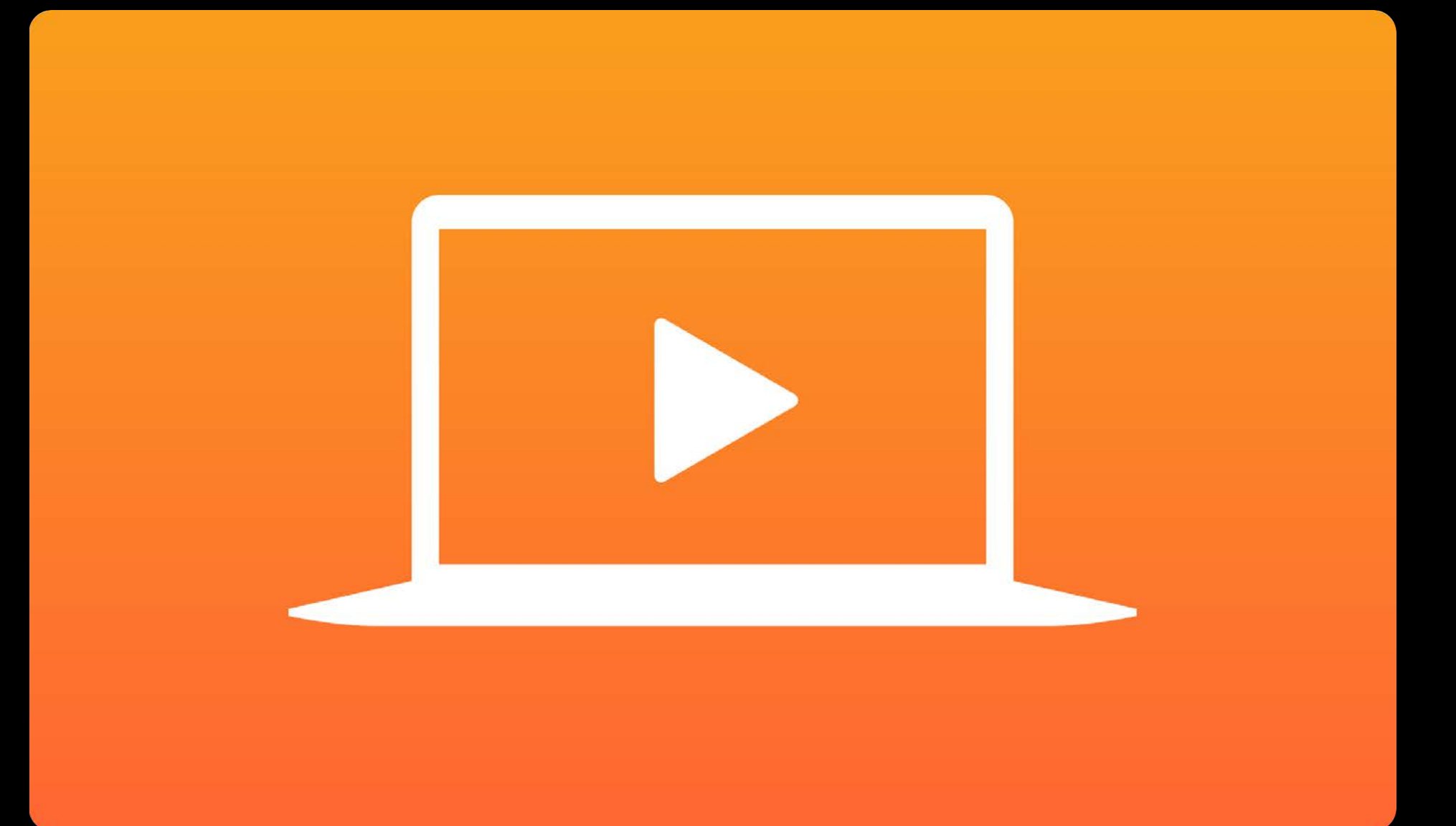
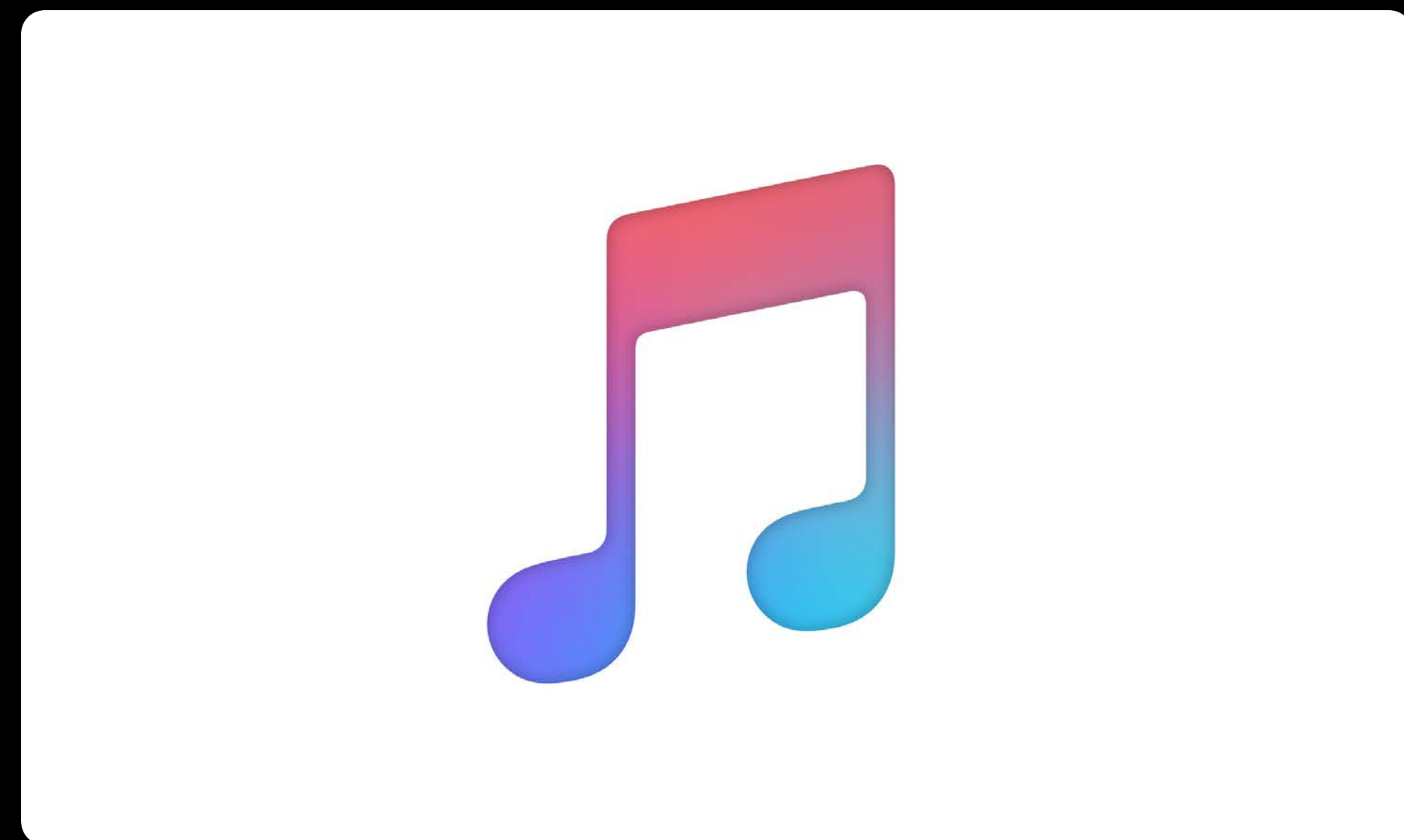
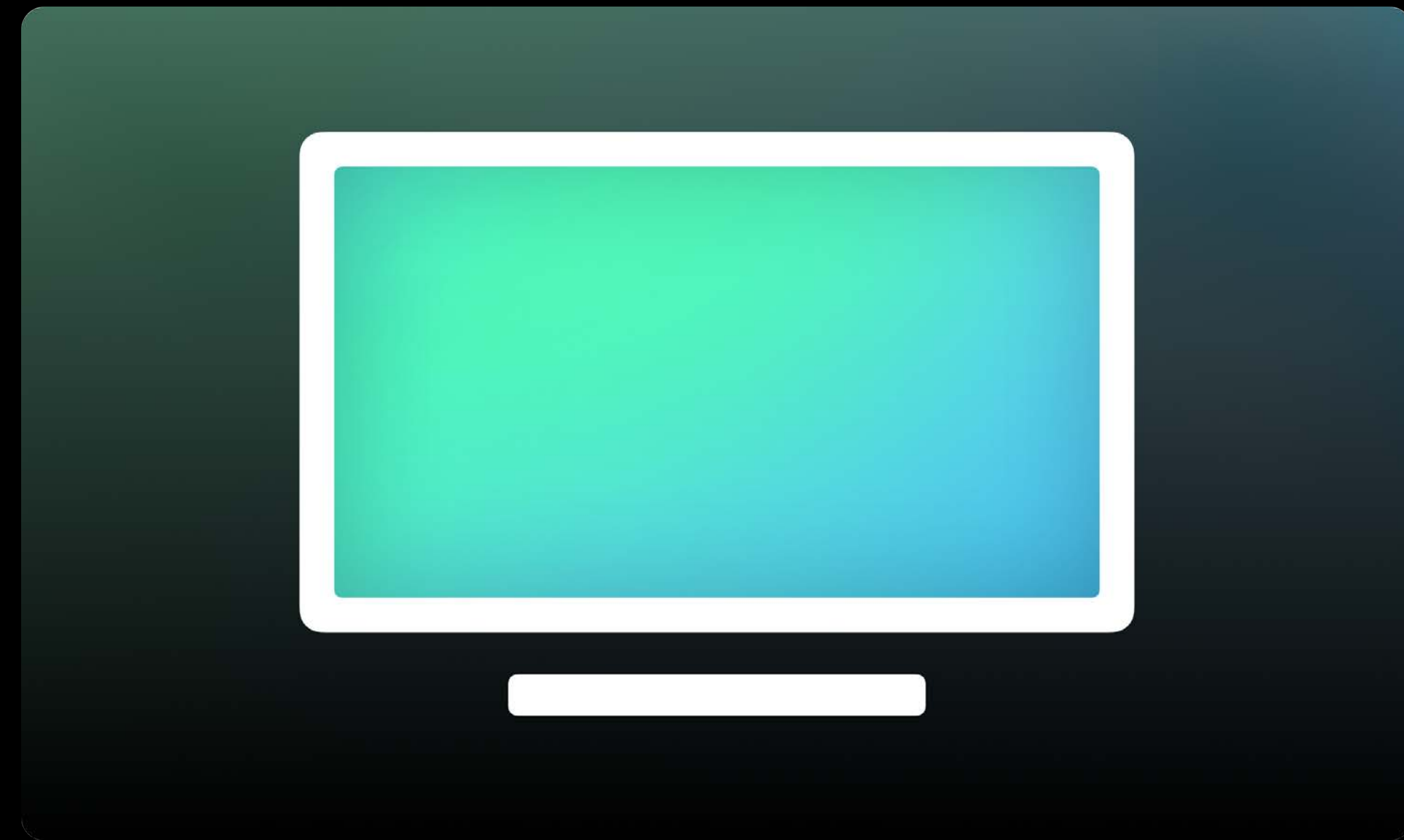


#WWDC18

What's New in TVMLKit

Session 238

Jeremy Foo, tvOS Engineering



Web Inspector Enhancements

Data Binding

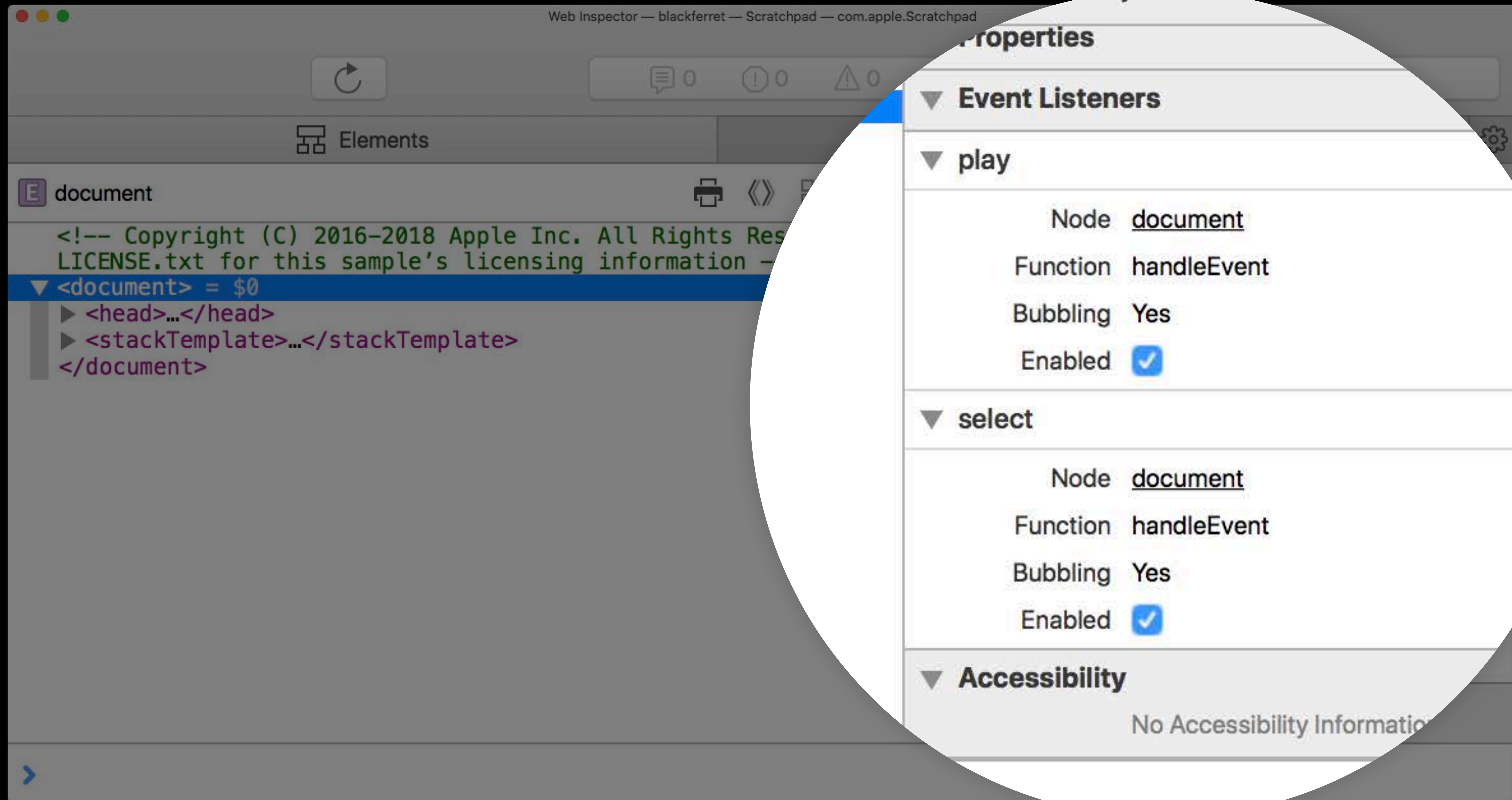
Customizing Playback

Web Inspector Enhancements

Web Inspector Enhancement

Toggle Event Listeners

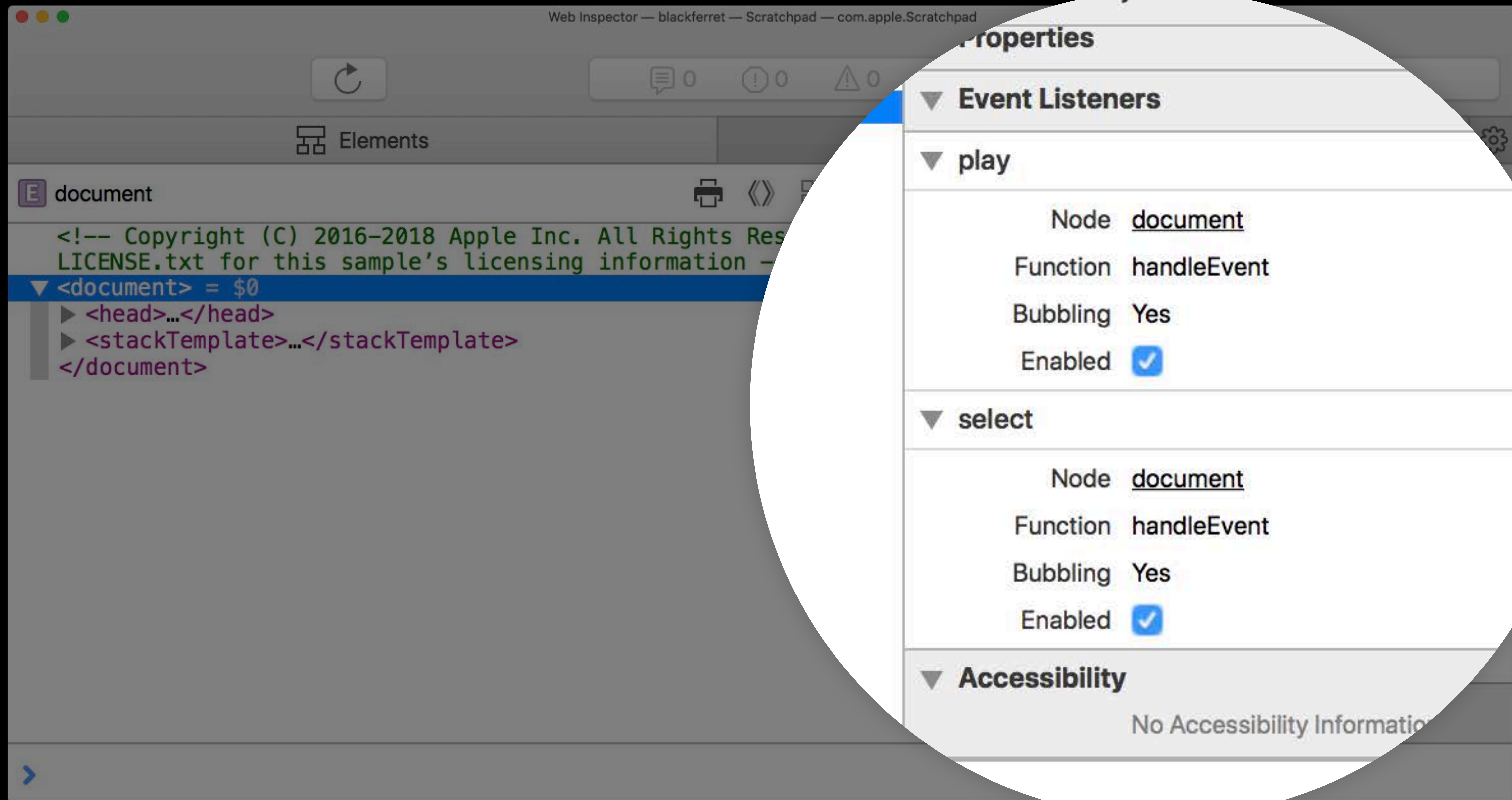
11.3



Web Inspector Enhancement

Toggle Event Listeners

NEW



Web Inspector Enhancement

Image Network Resources

NEW

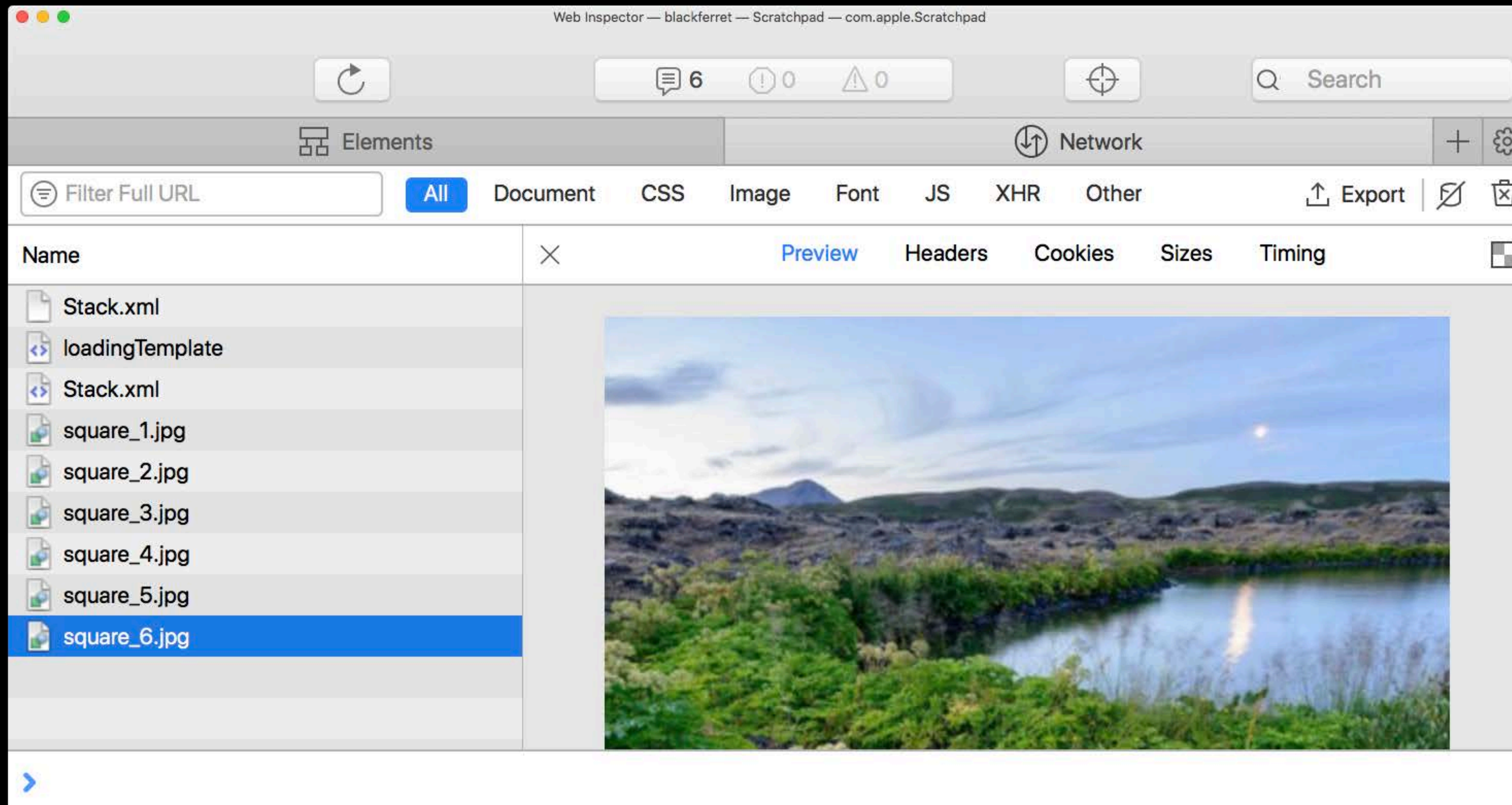
The screenshot shows the Web Inspector interface with the Network tab selected. The interface includes a search bar, a filter dropdown set to 'All', and a table of network resources. The table columns are Name, Domain, Type, Transfer Size, Time, and a column for visualizing the resource loading progress. The resources listed include XML files, a loading template, and several image files (square_1.jpg through square_6.jpg).

Name	Domain	Type	Transfer Size	Time	100.00ms	200.0ms	^
Stack.xml	localhost	xhr	12.72 KB	17.4...			
loadingTemplate	—	docu...	(memory)	0ms			
Stack.xml	localhost	docu...	(memory)	0ms			
square_1.jpg	localhost	jpg	73.12 KB	7.53...			
square_2.jpg	localhost	jpg	75.82 KB	9.32...			
square_3.jpg	localhost	jpg	67.37 KB	12.4...			
square_4.jpg	localhost	jpg	41.98 KB	14.2...			
square_5.jpg	localhost	jpg	66.62 KB	15.8...			
square_6.jpg	localhost	jpg	85.64 KB	17.5...			

Web Inspector Enhancement

Image Network Resources

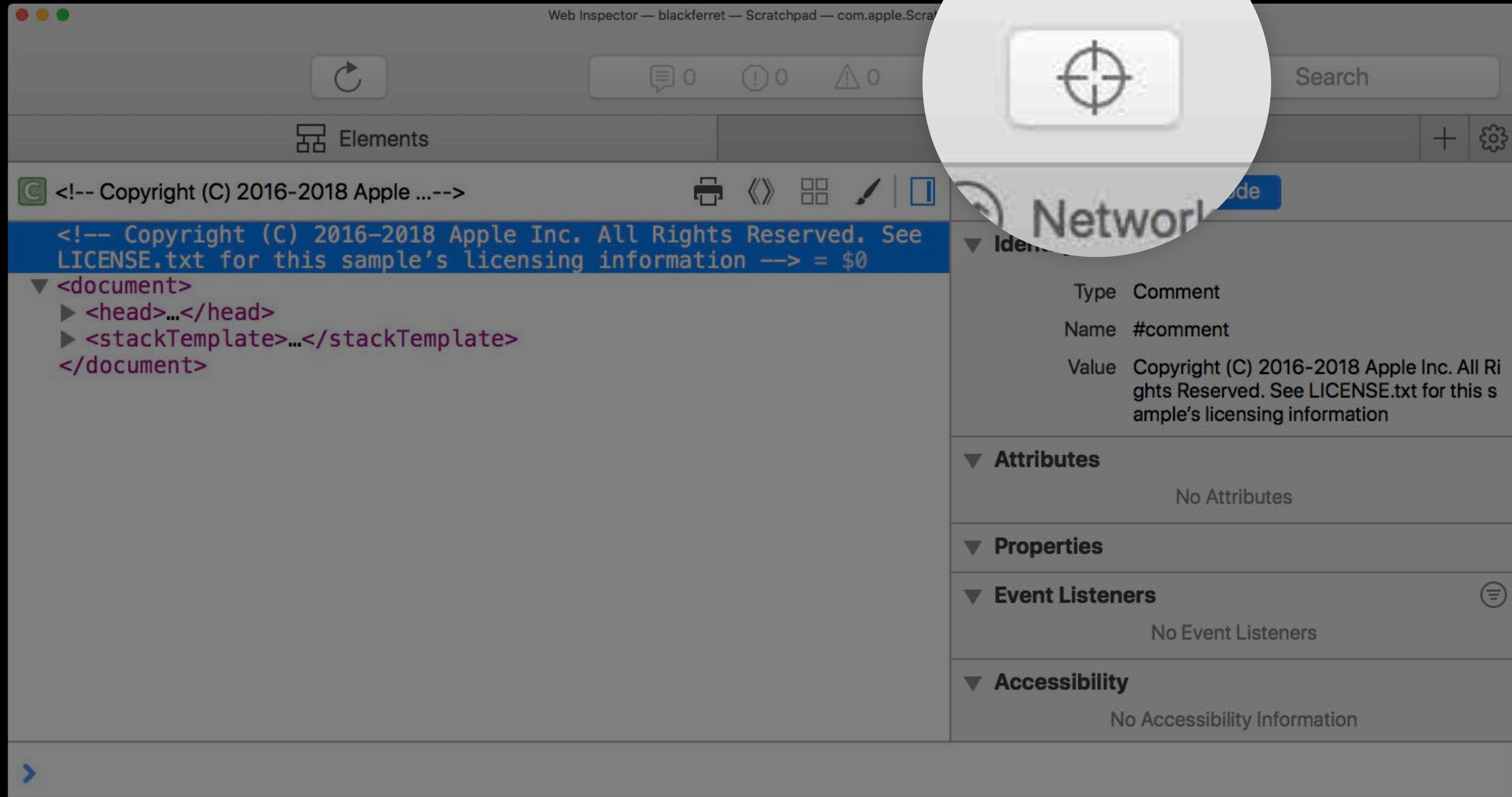
NEW



Web Inspector Enhancement

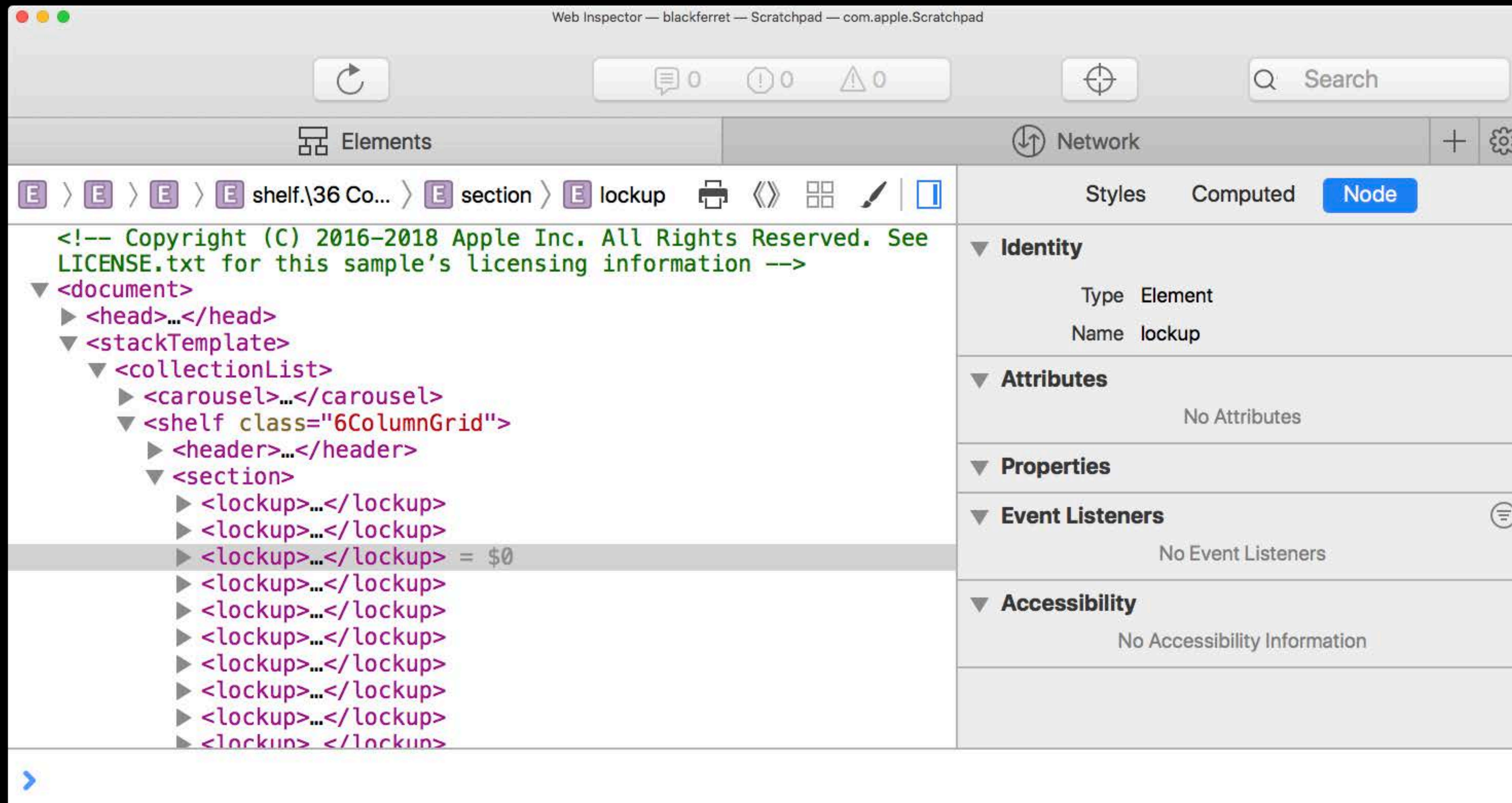
Show Focused Element

NEW



Web Inspector Enhancement

Show Focused Element



Web Inspector

Latest macOS

Safari technology preview

Data Binding

Data Binding

Transformation of data to user interface elements

Data Binding

Transformation of data to user interface elements

Separation of layout and application logic

Data Binding

Transformation of data to user interface elements

Separation of layout and application logic

Reduces JavaScript code

```
<banner>  
  <title>Title</title>  
  <description>Long description</description>  
</banner>
```


Data Binding

JS Data Fetch

```
{  
  title: "Title",  
  description: "Long Description"  
}
```

Data Binding

JS Data Fetch

JS Processing

```
{  
  title: "Title",  
  description: "Long Description"  
}
```

```
<banner>  
  <title>Title</title>  
  <description>Long Description</description>  
</banner>
```

Data Binding

JS Data Fetch

Data Bindings

```
{  
  title: "Title",  
  description: "Long Description"  
}
```

```
<banner>  
  <title binding="..." />  
  <description binding="..." />  
</banner>
```

```
<banner>  
  <title>Title</title>  
  <description>Long Description</description>  
</banner>
```

Data Binding

Binding keys

`attribute`

`textContent`

`items`

Data Binding

Attribute

Orange color not standard code highlight. Should it fit one of the existing code highlight styles?

Data

Template

Final Document

```
{  
  imageURL: "http://..."  
}
```

```

```

Data Binding

Attribute

Orange color not standard code highlight. Should it fit one of the existing code highlight styles?

Data

Template

Final Document

```
{  
  imageURL: "http://..."  
}
```



```
<img binding="@src:{imageURL}" />
```



```

```

Data Binding

Text Content

Orange color not standard code highlight. Should it fit one of the existing code highlight styles?

Data

Template

Final Document

```
{  
  title: "A nice title"  
}
```

```
<title>A nice title</title>
```

Data Binding

Text Content

Orange color not standard code highlight. Should it fit one of the existing code highlight styles?

Data

Template

Final Document

```
{
```

```
  title: "A nice title"
```

```
}
```



```
<title binding="textContent:{title}" />
```



```
<title>A nice title</title>
```


Data Binding

Items

Data

Template

Final Document

```
{
  items: [
    { /* list item 1 */ },
    { /* list item 2 */ },
    { /* list item 3 */ }
  ]
}
```

```
<section>
  <listItemLockup />
  <listItemLockup />
  <listItemLockup />
</section>
```

Data Binding

Items

Data

Template

Final Document

```
{
  items: [
    { /* list item 1 */ },
    { /* list item 2 */ },
    { /* list item 3 */ }
  ]
}
```

→

```
<section binding="items:{items}" />
  <prototypes>
    <listItemLockup />
  </prototypes>
</menuBar>
```

→

```
<section>
  <listItemLockup />
  <listItemLockup />
  <listItemLockup />
</section>
```

Data Binding

Binding keys

`attribute`

`textContent`

`items`

Data Binding

NEW

Binding keys

```
children
```

Data Binding

NEW

Binding keys

children

Special elements

fragment

rules

Data Binding

Children



NEW

Generic form of `items` binding

Data Binding

Children



NEW

Generic form of `items` binding

Generates children of any element

Data Binding

Children



NEW

Generic form of `items` binding

Generates children of any element

Works like `items` binding

Data Binding

Children

Data

```
{
  items: [
    { /* menu item 1 */ },
    { /* menu item 2 */ },
    { /* menu item 3 */ }
  ]
}
```

Data Binding

Children

Data

```
{  
  items: [  
    { /* menu item 1 */ },  
    { /* menu item 2 */ },  
    { /* menu item 3 */ }  
  ]  
}
```

Final Document

```
<menuBar>  
  <menuItem />  
  <menuItem />  
  <menuItem />  
</menuBar>
```

Data Binding

Children

Data

Template

Final Document

```
{
  items: [
    { /* menu item 1 */ },
    { /* menu item 2 */ },
    { /* menu item 3 */ }
  ]
}
```

→

```
<menuBar binding="children:{items}" />
  <prototypes>
    <menuItem />
  </prototypes>
</menuBar>
```

→

```
<menuBar>
  <menuItem />
  <menuItem />
  <menuItem />
</menuBar>
```

```
<menuBar>  
  <nowPlayingMenuItem />  
  <menuItem />  
  <menuItem />  
  <menuItem />  
</menuBar>
```

Data Binding

Fragment



NEW

Invisible element to help compartmentalize DOM

Data Binding

Fragment



NEW

Invisible element to help compartmentalize DOM

Children of a `fragment` are visible

Data Binding

Fragment



NEW

Invisible element to help compartmentalize DOM

Children of a `fragment` are visible

Works with `children` binding

Data Binding

Fragment

Data

```
{
  items: [
    { /* menu item 1 */ },
    { /* menu item 2 */ },
    { /* menu item 3 */ }
  ]
}
```

Final Document

```
<menuBar>
  <nowPlayingMenuItem />
  <fragment>
    <menuItem />
    <menuItem />
    <menuItem />
  </fragment>
</menuBar>
```


Data Binding

Fragment

Data

Template

Final Document

```
{
  items: [
    { /* menu item 1 */ },
    { /* menu item 2 */ },
    { /* menu item 3 */ }
  ]
}
```



```
<menuBar>
  <nowPlayingMenuItem />
  <fragment binding="children:{items}">
    <prototypes>
      <menuItem />
    </prototypes>
  </fragment>
</menuBar>
```



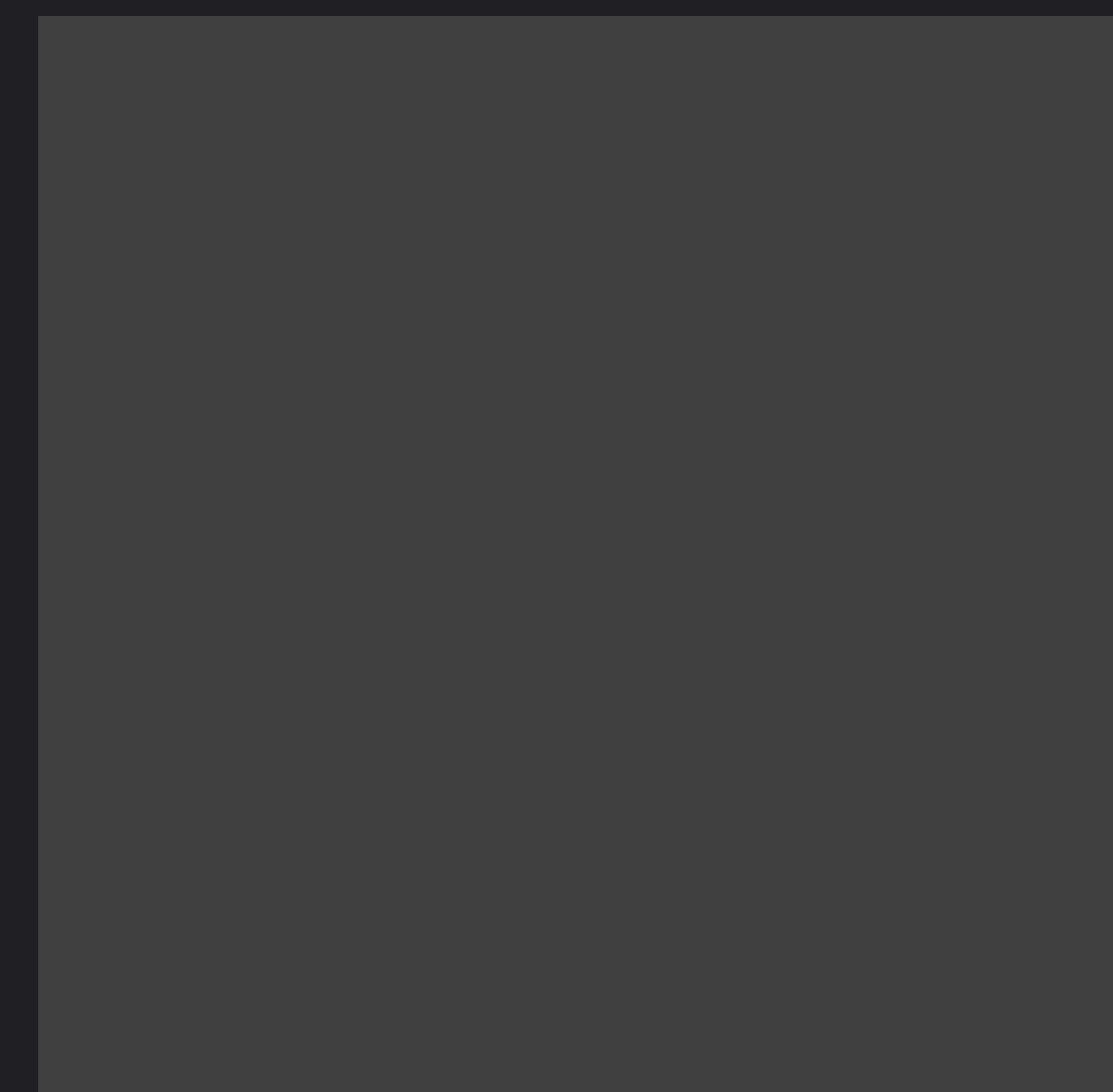
```
<menuBar>
  <nowPlayingMenuItem />
  <fragment>
    <menuItem />
    <menuItem />
    <menuItem />
  </fragment>
</menuBar>
```

```
{  
  img: "...",  
  title: "...",  
  progress: 0.60  
}
```

```
{  
  img: "...",  
  title: "...",  
  progress: 0.60  
}
```

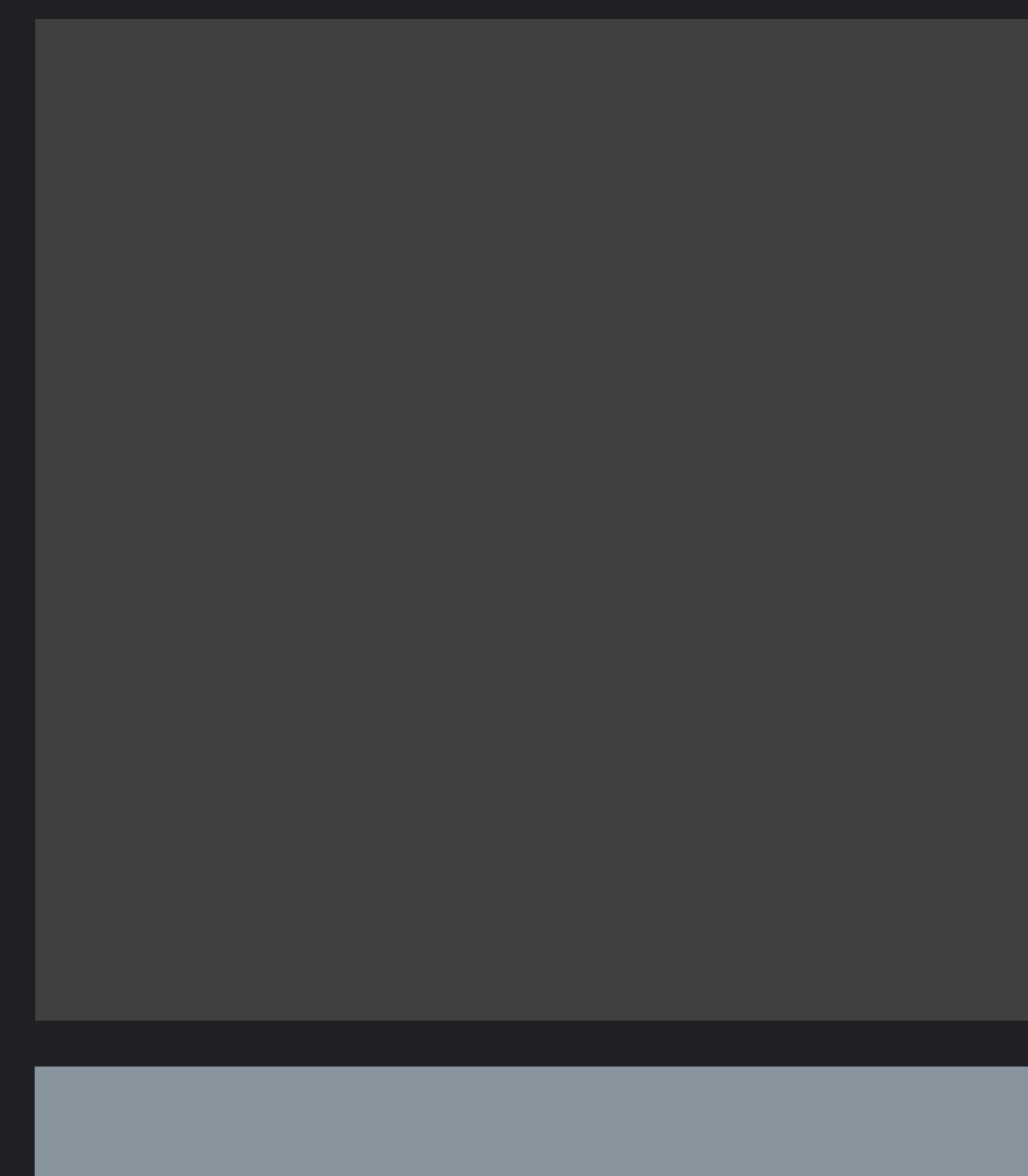
progress = 0%

```
{  
  img: "...",  
  title: "...",  
  progress: 0.60  
}
```



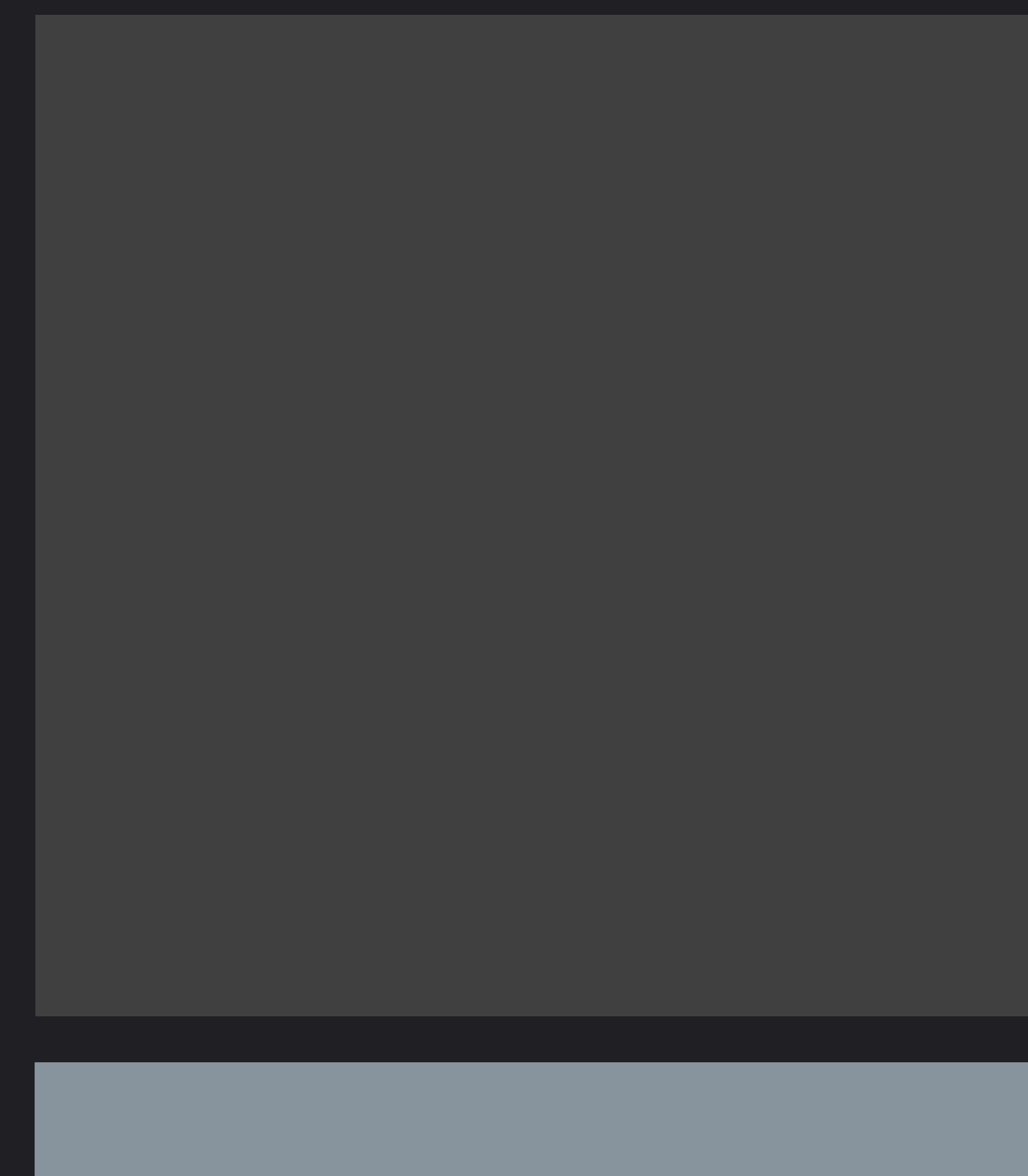
progress = 0%

```
{  
  img: "...",  
  title: "...",  
  progress: 0.60  
}
```

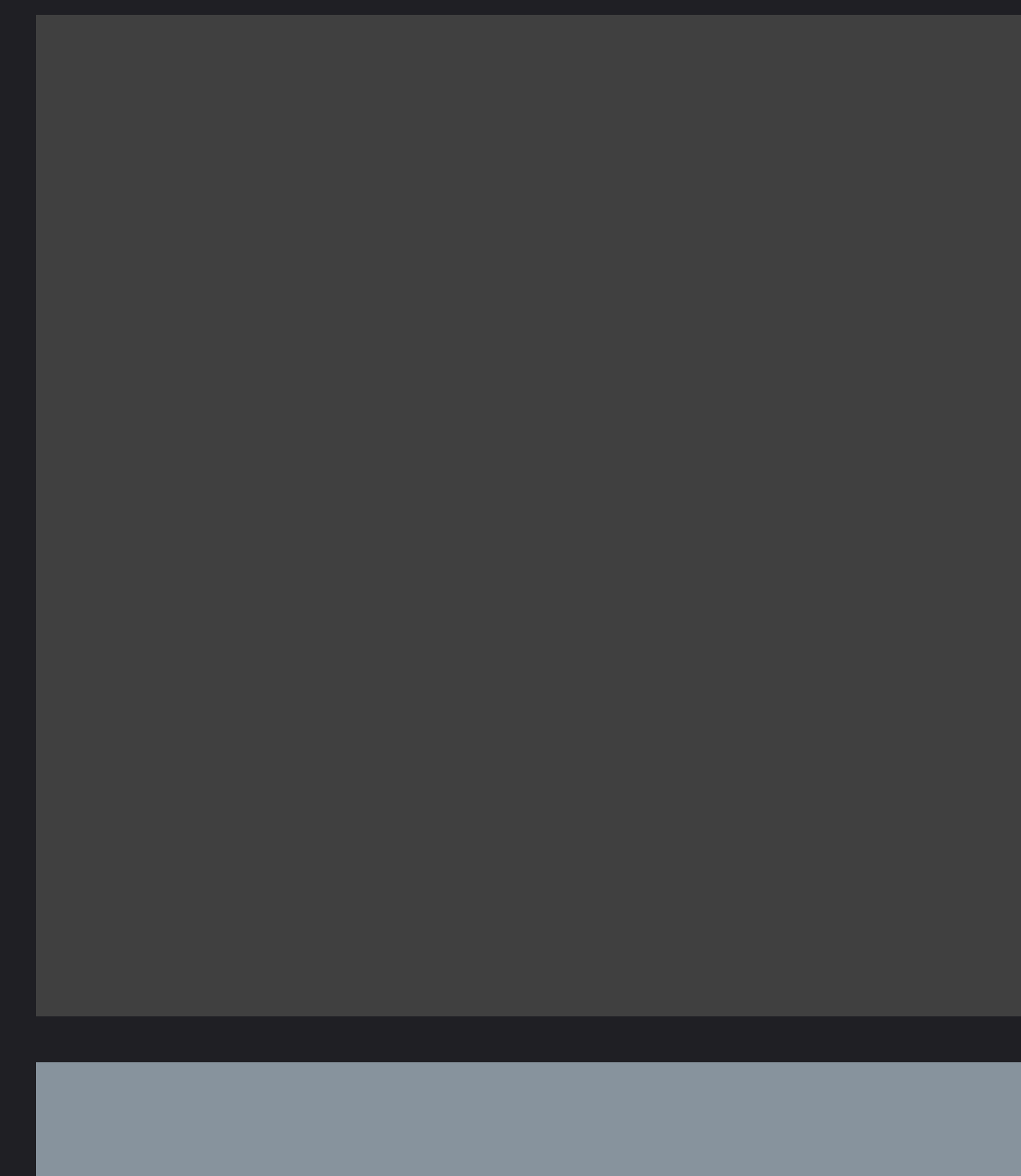


progress = 0%

```
{  
  img: "...",  
  title: "...",  
  progress: 0.60  
}
```

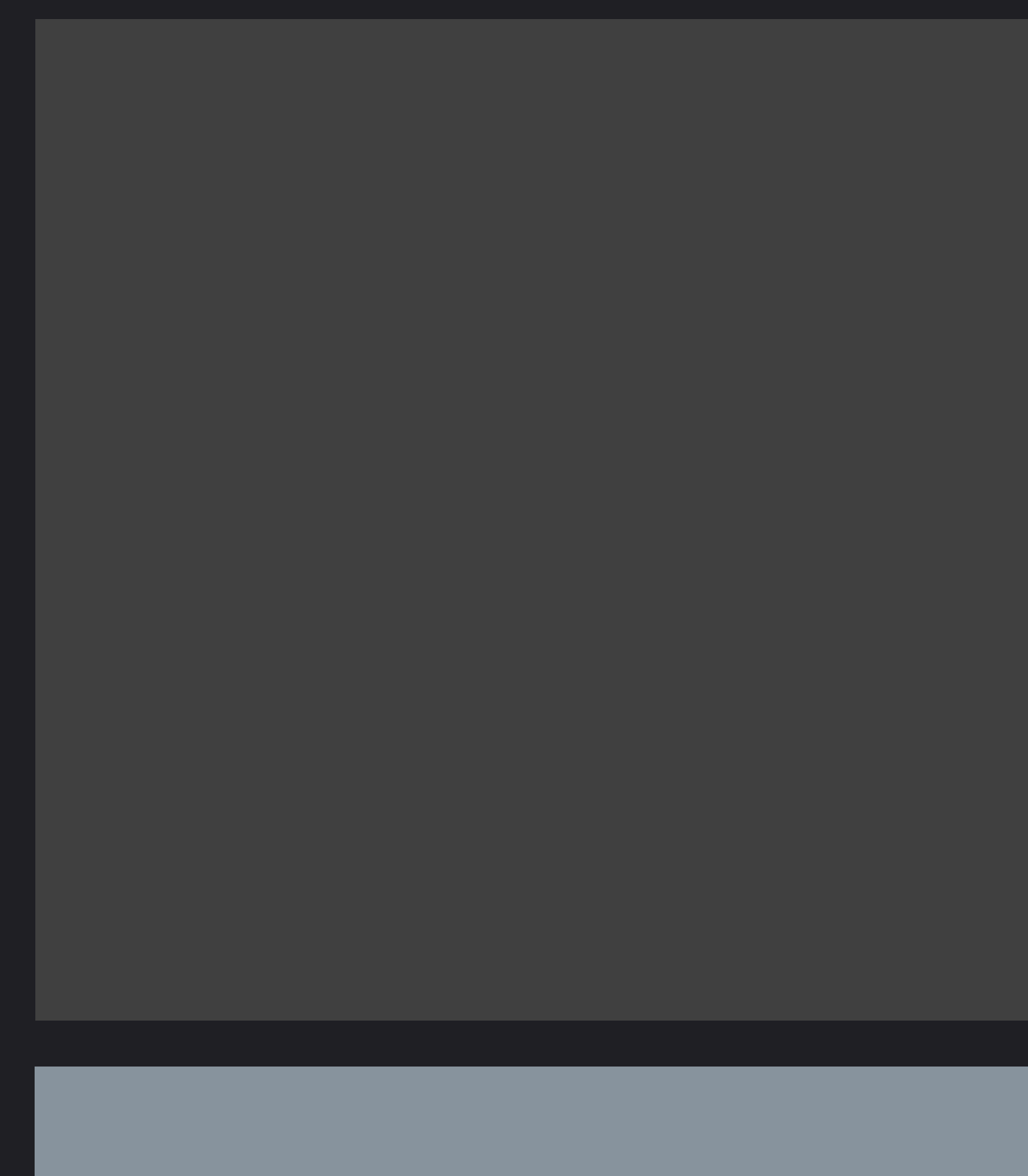


progress = 0%

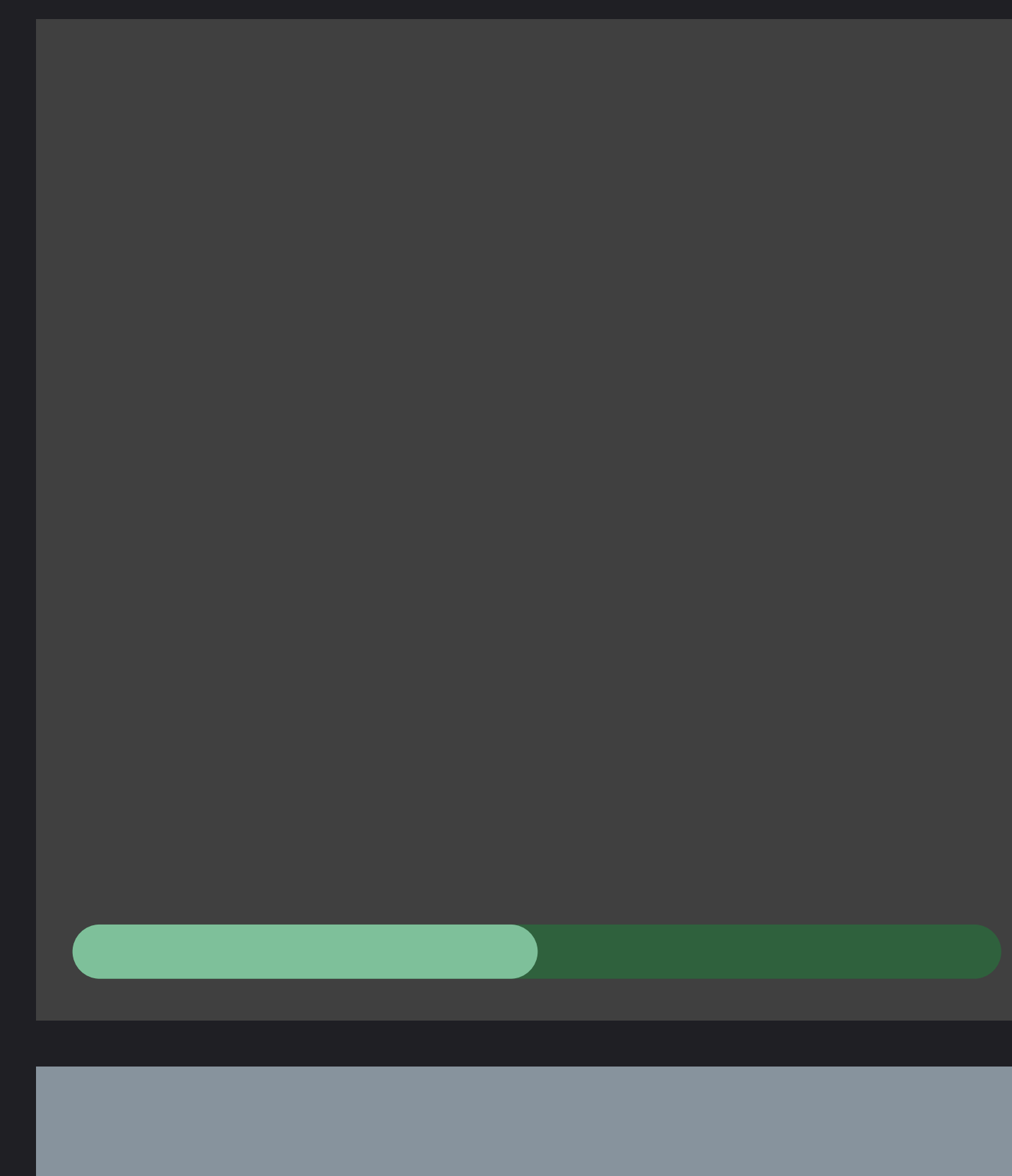


progress > 0%

```
{  
  img: "...",  
  title: "...",  
  progress: 0.60  
}
```



progress = 0%



progress > 0%

Data Binding

Data

```
{  
  img: "...",  
  title: "...",  
  progress: 0.0  
}
```



Final Document

```
<lockup>  
    
  <title>...</title>  
</lockup>
```

```
{  
  img: "...",  
  title: "...",  
  progress: 0.60  
}
```



```
<lockup>  
    
  <title>...</title>  
  <overlay>  
    <progressBar value="0.60" />  
  </overlay>  
</lockup>
```


Data Binding

Data

```
{  
  img: "...",  
  title: "...",  
  progress: 0.0  
}
```

Template

Final Document

```
<lockup>  
    
  <title>...</title>  
</lockup>
```

```
{  
  img: "...",  
  title: "...",  
  progress: 0.60  
}
```

```
<lockup>  
    
  <title>...</title>  
  <overlay>  
    <progressBar value="0.60" />  
  </overlay>  
</lockup>
```

Data Binding

Rules



NEW

Refinement of UI based on data states

Data Binding

Rules



NEW

Refinement of UI based on data states

Invisible element

Data Binding

Rules



NEW

Refinement of UI based on data states

Invisible element

Operates on sibling elements

Data Binding

Rules

We are not allowed to alter your code in any way. Please make adjustments

```
<prototypes>
  <lockup>
    <img binding="@src:{imgURL}" />
    <title binding="textContent:{title}" />
    <placeholder tag="progress" />
  </lockup>
</prototypes>
```

Data Binding

Rules

We are not allowed to alter your code in any way. Please make adjustments

```
<prototypes>
  <lockup>
    <img binding="@src:{imgURL}" />
    <title binding="textContent:{title}" />
    <placeholder tag="progress" />
  </lockup>
</prototypes>
```

Data Binding

Rules

We are not allowed to alter your code in any way. Please make adjustments

```
<prototypes>
  <lockup>
    <img binding="@src:{imgURL}" />
    <title binding="textContent:{title}" />
    <placeholder tag="progress" />
    <rules>
      <specialize state="{progress}-greater-than:0">
        <overlay tag="progress">
          <progressBar binding="@value:{progress}" />
        </overlay>
      </specialize>
    </rules>
  </lockup>
</prototypes>
```

Data Binding

Rules

We are not allowed to alter your code in any way. Please make adjustments

```
<prototypes>
  <lookup>
    <img binding="@src:{imgURL}" />
    <title binding="textContent:{title}" />
    <placeholder tag="progress" />
    <rules>
      <specialize state="{progress}-greater-than:0">
        <overlay tag="progress">
          <progressBar binding="@value:{progress}" />
        </overlay>
      </specialize>
    </rules>
  </lookup>
</prototypes>
```


Data Binding

Rules

We are not allowed to alter your code in any way. Please make adjustments

```
<prototypes>
  <lookup>
    <img binding="@src:{imgURL}" />
    <title binding="textContent:{title}" />
    <placeholder tag="progress" />
    <rules>
      <specialize state="{progress}-greater-than:0">
        <overlay tag="progress">
          <progressBar binding="@value:{progress}" />
        </overlay>
      </specialize>
    </rules>
  </lookup>
</prototypes>
```

Data Binding

Rules

We are not allowed to alter your code in any way. Please make adjustments

```
<prototypes>
  <lookup>
    <img binding="@src:{imgURL}" />
    <title binding="textContent:{title}" />
    <placeholder tag="progress" />
    <rules>
      <specialize state="{progress}-greater-than:0">
        <overlay tag="progress">
          <progressBar binding="@value:{progress}" />
        </overlay>
      </specialize>
    </rules>
  </lookup>
</prototypes>
```

Data Binding

Rules

We are not allowed to alter your code in any way. Please make adjustments

```
<prototypes>
  <lockup>
    <img binding="@src:{imgURL}" />
    <title binding="textContent:{title}" />
    <placeholder tag="progress" />
    <rules>
      <specialize state="{progress}-greater-than:0">
        <overlay tag="progress">
          <progressBar binding="@value:{progress}" />
        </overlay>
      </specialize>
    </rules>
  </lockup>
</prototypes>
```

Data Binding

We cannot make code any smaller than it is on slide now without losing legibility.

Data

```
{
  img: "...",
  title: "...",
  progress: 0.0
}
```

<lockup>

```
  <img binding="@src:{imgURL}" />
  <title binding="textContent:{title}" />
  <placeholder tag="progress" />
  <rules>
```

```
    <specialize state="{progress}-greater-than:0">
```

```
      <overlay tag="progress">
```

```
        <progressBar binding="@value:{progress}" />
```

```
      </overlay>
```

```
    </specialize>
```

```
  </rules>
```

```
</lockup>
```

Final Document

```
<lockup>
  
  <title>...</title>
</lockup>
```

```
{
  img: "...",
  title: "...",
  progress: 0.60
}
```

</lockup>

```
<lockup>
  
  <title>...</title>
  <overlay>
    <progressBar value="0.60" />
  </overlay>
</lockup>
```

Customizing Playback

Customizing Playback

Works with all playback experiences

Customizing Playback

Works with all playback experiences

Provide `TVPlayer` and `UIViewController`

Customizing Playback

Works with all playback experiences

Provide `TVPlayer` and `UIViewController`

Close analogue to TVMLKit JS APIs

JS Player

TVPlayer

JS Playlist

TVPlaylist

JS MediaItem

TVMediaItem

Customizing Playback

Works with all playback experiences

Provide `TVPlayer` and `UIViewController`

Close analogue to TVMLKit JS APIs

JS Player

TVPlayer

JS Playlist

TVPlaylist

JS MediaItem

TVMediaItem

Limited JavaScript Bridge

Customizing Playback

TVPlayer

Customizing Playback

TVPlayer

Public `AVPlayer` adaptor to Playback Pipeline

Customizing Playback

TVPlayer

Public `AVPlayer` adaptor to Playback Pipeline

Dispatch custom events to JavaScript

Customizing Playback

TVPlayer

Public `AVPlayer` adaptor to Playback Pipeline

Dispatch custom events to JavaScript

KVO properties for changes from JavaScript

Customizing Playback

TVPlayer

Public `AVPlayer` adaptor to Playback Pipeline

Dispatch custom events to JavaScript

KVO properties for changes from JavaScript

Sequential playlist

Customizing Playback

TVPlayer

Public `AVPlayer` adaptor to Playback Pipeline

Dispatch custom events to JavaScript

KVO properties for changes from JavaScript

Sequential playlist

```
// TVApplicationControllerDelegate

func player(for appController: TVApplicationController) -> TVPlayer? {
    return TVPlayer()
}
```

Customizing Playback

Playback User Interface

Implement your own playback UI

Customizing Playback

Playback User Interface

Implement your own playback UI

```
// TVInterfaceCreating

func playerViewController(for player: TVPlayer) -> UIViewController? {
    return MyPlaybackViewController.init(player: player)
}
```

Customizing Playback

Caveats

Handle “should” events yourself

```
shouldHandleStateChange (pause, fast forward, etc)  
shouldChangeToMediaAtIndex
```

Customizing Playback

Caveats

Handle “should” events yourself

```
shouldHandleStateChange (pause, fast forward, etc)  
shouldChangeToMediaAtIndex
```

Use `AVContentKeySession` for secure key loading

Customizing Playback

Caveats

Handle “should” events yourself

```
shouldHandleStateChange (pause, fast forward, etc)  
shouldChangeToMediaAtIndex
```

Use AVContentKeySession for secure key loading

Advances in HTTP Live Streaming

WWDC 2017

AVContentKeySession Best Practices

WWDC 2018

Customizing Playback

Caveats

Handle “should” events yourself

```
shouldHandleStateChange (pause, fast forward, etc)  
shouldChangeToMediaAtIndex
```

Use AVContentKeySession for secure key loading

Bring your own additional user interface

```
overlay  
interactiveOverlay
```

Summary

Data Binding is now more powerful

Build your own native playback experience

More Information

<https://developer.apple.com/wwdc18/238>

 **WWDC18**