

P2H Player Basic

API

Instantiation

The player needs to be instantiated before it can run. This can be done by simply calling the element that contains all the slides. In the code example below, the slides are contained within an element with the id “player1”.

```
$('#player1').p2h();
```

Public properties

A public property needs to be passed as an option in the player's constructor.

repeat

A boolean property that is false by default. When false, the prev button is disabled if the first slide is in view and the next button is disabled when the last slide is in view.

```
$('#player1').p2h({repeat: true});
```

enableSlider

A boolean property that is true by default. This property is for debugging only and it speeds up the slider when value is false. This allows moving from one slide to another without the sliding animation getting in the way.

```
$('#player1').p2h({enableSlider: false});
```

enableKeyboardNav

A boolean property that is true by default. When false, keyboard navigation is disabled.

```
$('#player1').p2h({enableKeyboardNav: false});
```

enableClickNav

A boolean property that is true by default. When false, clicking or tapping a slide does not go to the next slide.

```
$('#player1').p2h({enableClickNav: false});
```

Public methods

A public method can be called in the same way the player was instantiated. Note that method names are case sensitive.

first

Navigates to the first slide.

```
$('#player1').p2h('first');
```

prev

Navigates to the previous slide, moving backwards. If a Boolean argument of true is provided, then sub animations are not respected. Calling next method with an argument of true, this will cause the next slide to show regardless of whether or not sub animations are present in the currently viewed slide.

```
$('#player1').p2h('prev');// respects animations
```

or

```
$('#player1').p2h('prev', true); // does not respects animation.
```

next

Navigates to the next slide, moving forward. If a Boolean argument of true is provided, then sub animations are not respected. Calling next method with an argument of true, this will cause the next slide to show regardless of whether or not sub animations are present in the currently viewed slide.

```
$('#player1').p2h('next');// respects animations
```

or

```
$('#player1').p2h('next', true); // does not respect animation.
```

last

Navigates to the last slide.

```
$('#player1').p2h('last');
```

goto

Allows navigation to a slide by slide index. If the index provided does not correspond to an existing slide, the method will simply do nothing. The code sample below navigates to the second slide, note the second parameter of 2.

```
$('#player1').p2h('goto', 2);
```

toggleSidePane

Toggles the side pane. i.e. if the sidepane is expanded then it is collapsed, otherwise expanded.

```
$('#player1').p2h('toggleSidePane');
```

count

Returns the number of slides.

```
var count = $('#player1').p2h('count');
```

```
window.console.log(count);
```

slides

Returns all slides available as a jquery object.

```
var slides = $('#player1').p2h('slides');
```

```
window.console.log(slides.length);
```

getCurrentIndex

Returns the index of the current slide displayed.

```
var index = $('#player1').p2h('getCurrentIndex');
```

```
window.console.log(index);
```

togglePlayback

Toggles the playback on video/audio child elements. If the current state is playing then it will be paused and vice versa. Accepts two parameters. First is the HTML DOM element that might contain child audio/video elements. The second is the state of the playback. A value of true will play it and false will pause it.

```
$('#player1').p2h('togglePlayback', elem, true);
```

Public Events

It is possible to subscribe to event handlers that will fire when a determined operation is executed. Note that the event parameters are case sensitive. All events need to be subscribed during instantiation hence the code sample is provided after the event handler description below.

onLoaded

This event fires when the player is loaded for the first time.

onFirst

This event fires when the first slide is viewed, either through the use of the API manually or when reaching the first slide by clicking the next/prev buttons in the toolbar, or simply on first load, in which case the first slide is displayed. The index of the current slide is passed in through the

parameter.

onNext

This event fires when sliding to the next slide, either through the use of the API manually or by clicking the next button in the toolbar. The index of the current slide is passed in through the parameter.

onPrev

This event fires when sliding to the previous slide, either through the use of the API manually or by clicking the previous button in the toolbar. The index of the current slide is passed in through the parameter.

onLast

This event fires when the last slide is viewed, either through the use of the API manually or when reaching the last slide by clicking the next/prev buttons in the toolbar. The index of the current slide is passed in through the parameter.

onSwipeLeft

Triggered when a swipe event occurs moving in the left direction. A jquery type event object is passed as an argument.

onSwipeRight

Triggered when a swipe event occurs moving in the right direction. A jquery type event object is passed as an argument.

onClickTap

Triggered after a quick, complete touch event on touch devices. This is also triggered during a click event when clicking with the mouse. A jquery type event object is passed as an argument.

onBeforeAnimTo

Triggered before an animation is applied when transitioning in.

onAfterAnimTo

Triggered after an animation is applied when transitioning in.

onBeforeAnimFrom

Triggered before an animation is applied when transitioning out.

Events code sample below. Note that you don't need to subscribe to all events, but rather to just the ones you want to listen for.

```
$('#player1').p2h({
```

```
'onLoaded': function(){
    window.console.log('loaded');
}
, 'onFirst': function(i){
    window.console.log('On first fired for slide #' + i);
}
, 'onPrev': function(i){
    window.console.log('On prev fired for slide #' + i);
}
, 'onNext': function(i){
    window.console.log('On next fired for slide #' + i);
}
, 'onLast': function(i){
    window.console.log('On last fired for slide #' + i);
}
, 'onSwipeLeft': function(e){
    window.console.log(e);
}
, 'onSwipeRight': function(e){
    window.console.log(e);
}
, 'onClickTap': function(e){
    window.console.log(e);
}
, 'onBeforeAnimTo': function(elem, animationName){
    window.console.log('onBeforeAnimTo: ' + animationName);
}
, 'onAfterAnimTo': function(elem, animationName){
    window.console.log('onAfterAnimTo: ' + animationName);
}
, 'onBeforeAnimFrom': function(elem, animationName){
    window.console.log('onBeforeAnimFrom: ' + animationName);
}
});
```

CSS3 Animations

Css3 animations are applied on elements that contain the anim attribute with a valid animation name i. e. anim="1,20,0,fadein".

The animation name is extracted from the anim attribute always located in position 3 after the string split.

```
<div anim="1,20,0,fadein" id="1000" style="z-index:-1; ...">
  
  <pre style="...">PowerPoint </pre>
  <pre style="...">Presentations</pre>
</div>
```

The above example has a fadein animation expressed declaratively in the html. The css definition can be written as follows:

```
.p2h-player div.fadein{
  /*the default state*/
  -webkit-transition-duration: 1000ms;
  -moz-transition-duration: 1000ms;
  -o-transition-duration: 1000ms;
  -ms-transition-duration: 1000ms;
  transition-duration: 1000ms;

  -webkit-transition-property: opacity;
  -moz-transition-property: opacity;
  -o-transition-property: opacity;
  -ms-transition-property: opacity;
  transition-property: opacity;
  opacity: 0;
}
```

```
.p2h-player div.fadein.to{
  /*apply effect, from default state of 0 to full opacity 1*/
  opacity: 1;
}
```

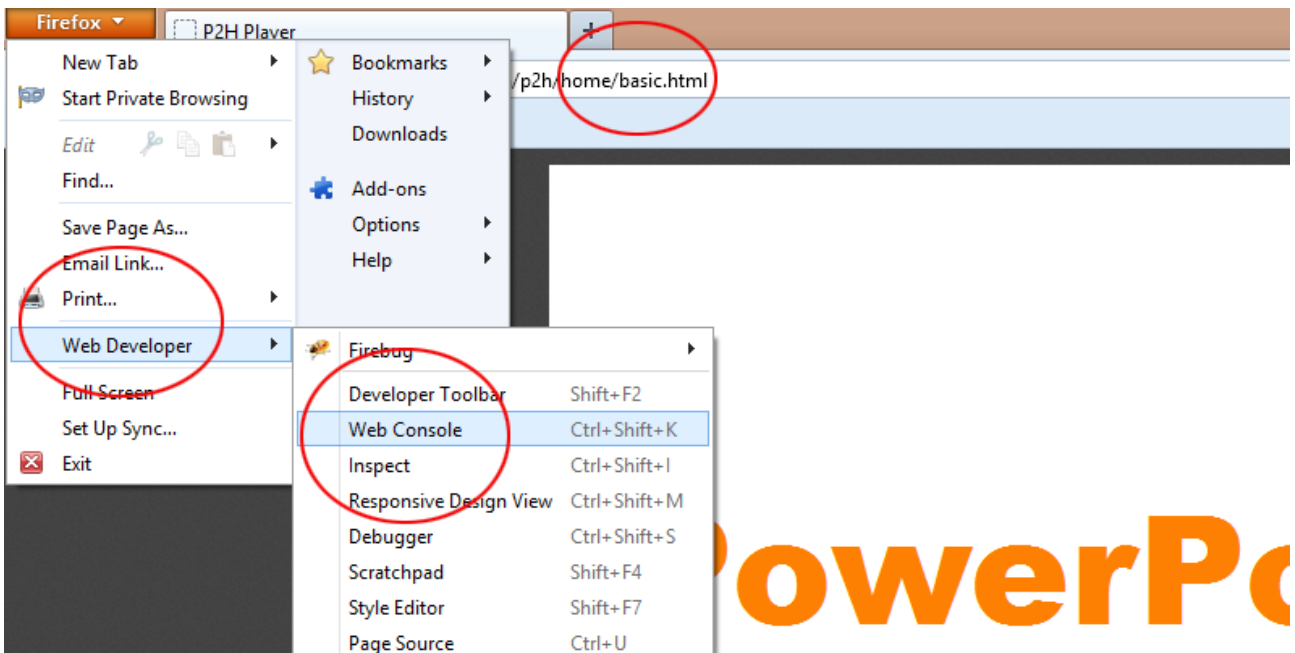
The player will add the css classes to play the animations. First it will apply the fadein class, this is the default state of the animation, following our example above, the default state is opacity: 0, it

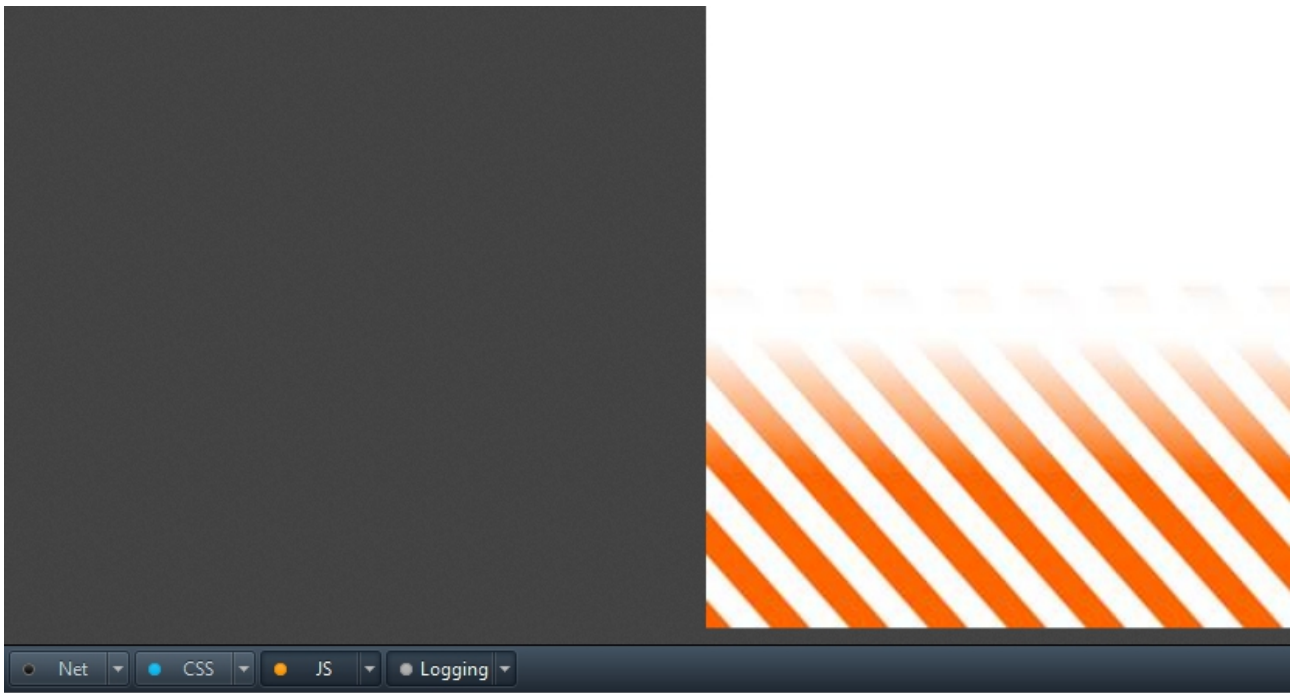
will then apply a “to” class. This is the state of the animation to which the element will transition to. In our example above, opacity is transitioned from 0 to 1 (when the “to” class is applied).

Testing the API

To test the API, a basic sample page is provided that does not include toolbars and navigation. Simply open the code in your browser of choice and execute the API calls. Reference the screen shot below for how this can be done on Firefox using the web console.

Every browser provides a similar set of dev tools you can use. Simply load basic.html file and launch the console as follows:





```
> $('#player1').p2h('next')
```

Simply hit enter and it will execute the code.