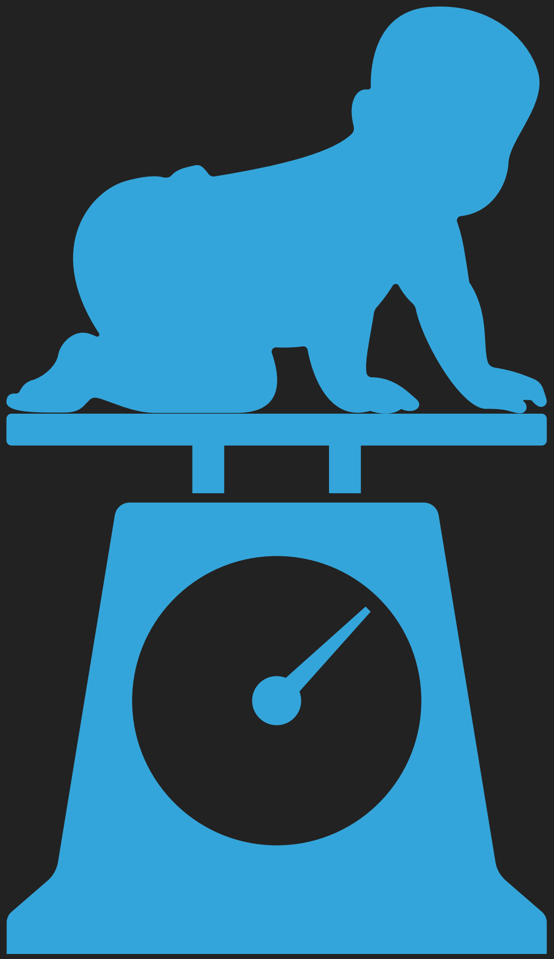
YOU SHOULD SEE A DR. ABOUT YOUR

SCALING

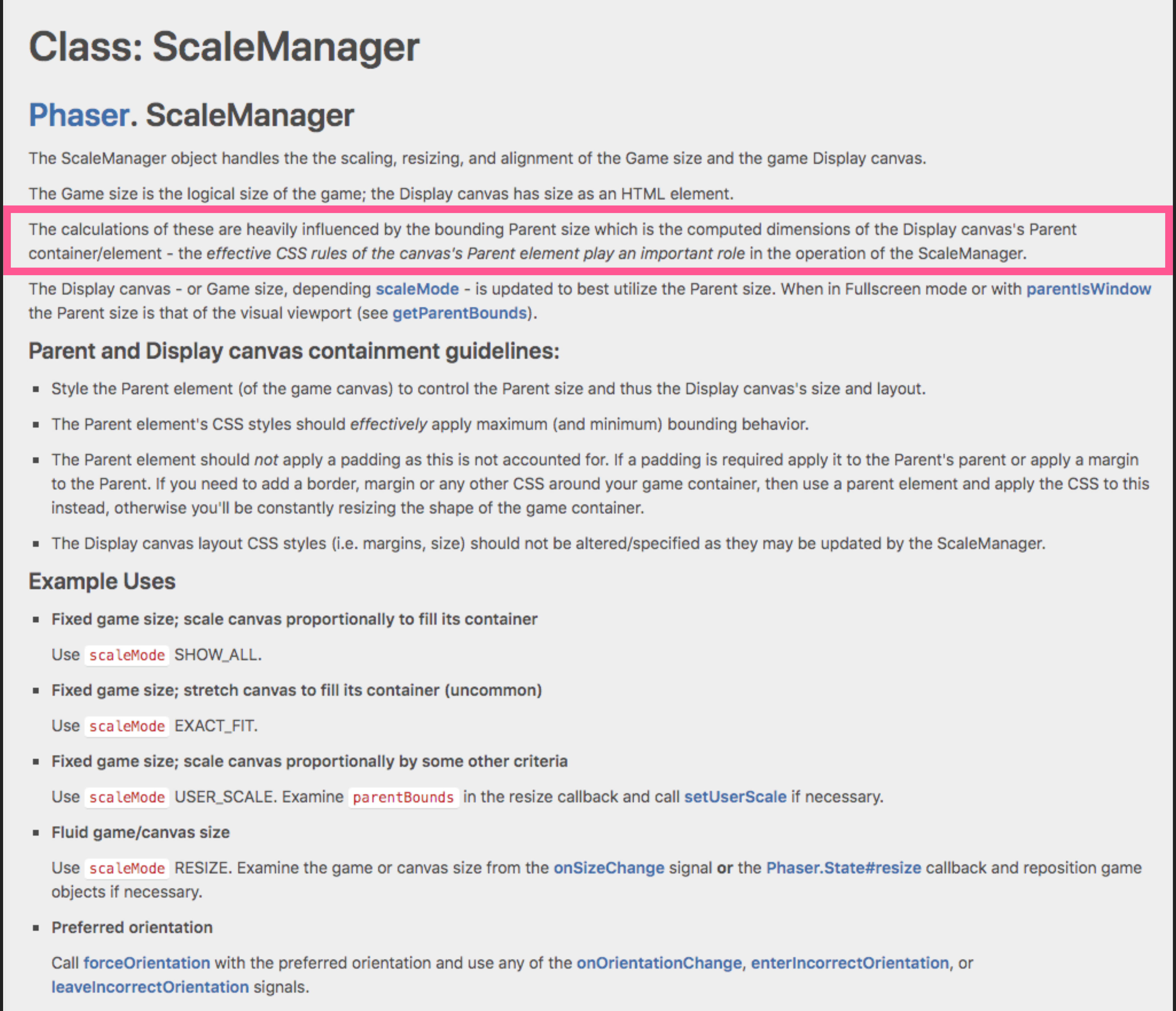


Phaser ScaleManager

"The ScaleManager object handles the the scaling, resizing, and alignment of the Game size and the game Display canvas."

Phaser CE ScaleManager Documentation

This opening description is worth reading through.



Class: ScaleManager

Phaser. ScaleManager

The ScaleManager object handles the the scaling, resizing, and alignment of the Game size and the game Display canvas.

The Game size is the logical size of the game; the Display canvas has size as an HTML element.

The calculations of these are heavily influenced by the bounding Parent size which is the computed dimensions of the Display canvas's Parent container/element - the *effective CSS rules of the canvas's Parent element play an important role* in the operation of the ScaleManager.

The Display canvas - or Game size, depending **scaleMode** - is updated to best utilize the Parent size. When in Fullscreen mode or with **parentIsWindow** the Parent size is that of the visual viewport (see **getParentBounds**).

Parent and Display canvas containment guidelines:

- Style the Parent element (of the game canvas) to control the Parent size and thus the Display canvas's size and layout.
- The Parent element's CSS styles should *effectively* apply maximum (and minimum) bounding behavior.
- The Parent element should *not* apply a padding as this is not accounted for. If a padding is required apply it to the Parent's parent or apply a margin to the Parent. If you need to add a border, margin or any other CSS around your game container, then use a parent element and apply the CSS to this instead, otherwise you'll be constantly resizing the shape of the game container.
- The Display canvas layout CSS styles (i.e. margins, size) should not be altered/specified as they may be updated by the ScaleManager.

Example Uses

- **Fixed game size; scale canvas proportionally to fill its container**

  Use `scaleMode` SHOW_ALL.
- **Fixed game size; stretch canvas to fill its container (uncommon)**

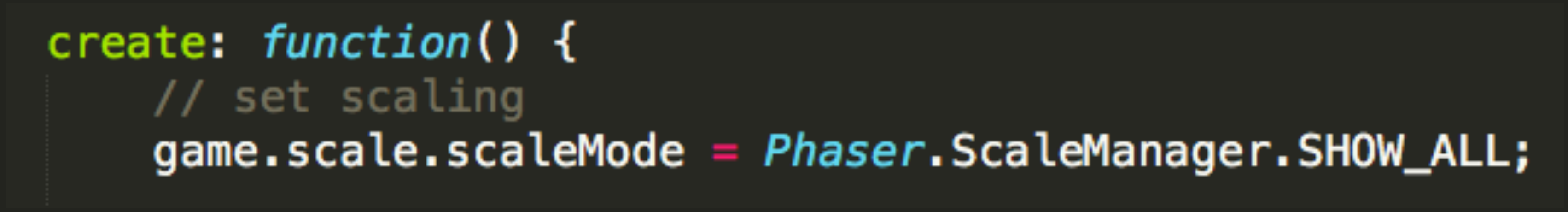
  Use `scaleMode` EXACT_FIT.
- **Fixed game size; scale canvas proportionally by some other criteria**

  Use `scaleMode` USER_SCALE. Examine `parentBounds` in the resize callback and call **setUserScale** if necessary.
- **Fluid game/canvas size**

  Use `scaleMode` RESIZE. Examine the game or canvas size from the **onSizeChange** signal **or** the **Phaser.State#resize** callback and reposition game objects if necessary.
- **Preferred orientation**

  Call **forceOrientation** with the preferred orientation and use any of the **onOrientationChange**, **enterIncorrectOrientation**, or **leaveIncorrectOrientation** signals.

SETTING THE SCALE MODE

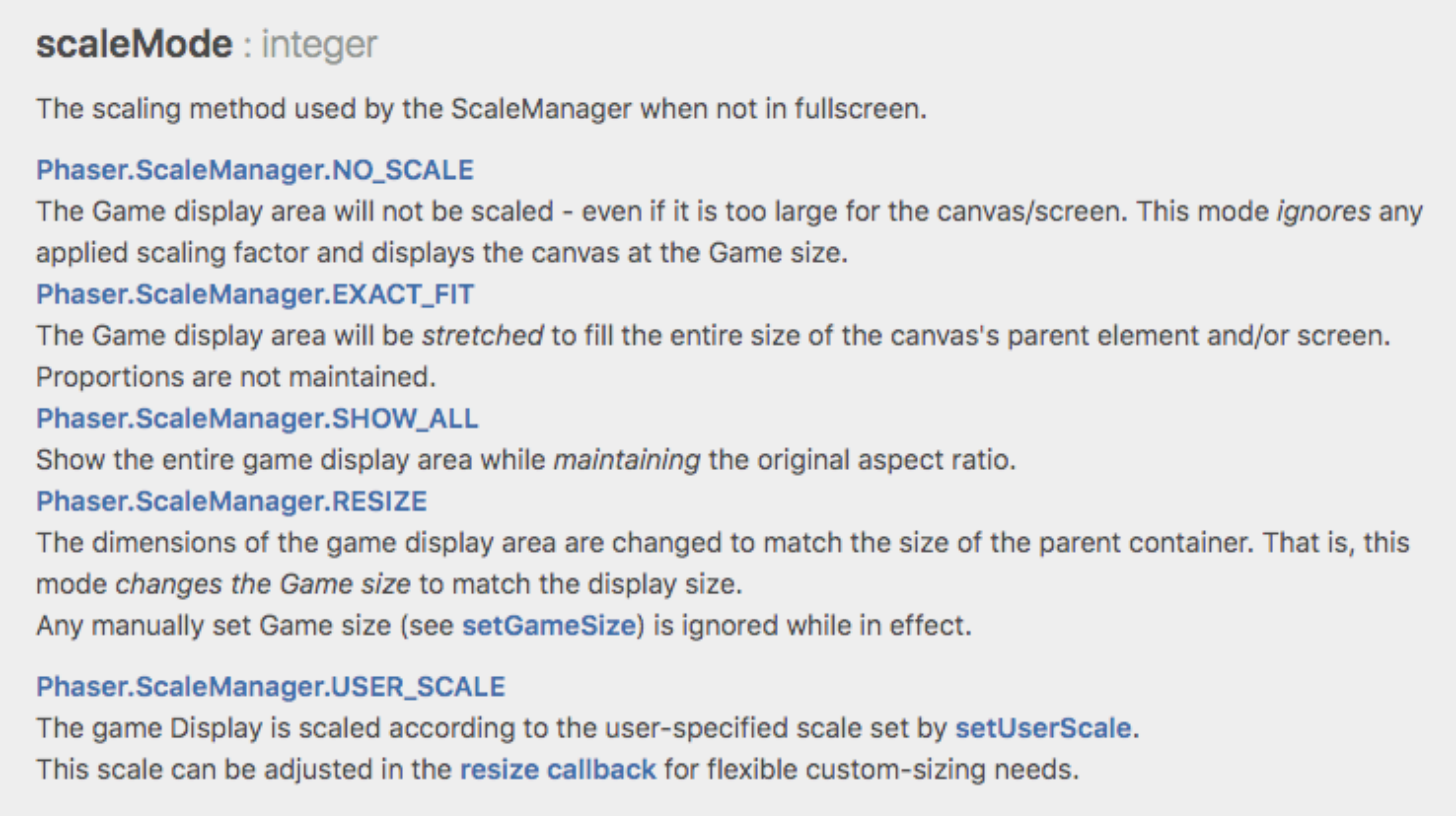


```
create: function() {
    // set scaling
    game.scale.scaleMode = Phaser.ScaleManager.SHOW_ALL;
```

scaleMode property

a Phaser constant
describing what type
of scaling we want

Only one line of code gets you proportional scaling!



scaleMode : integer

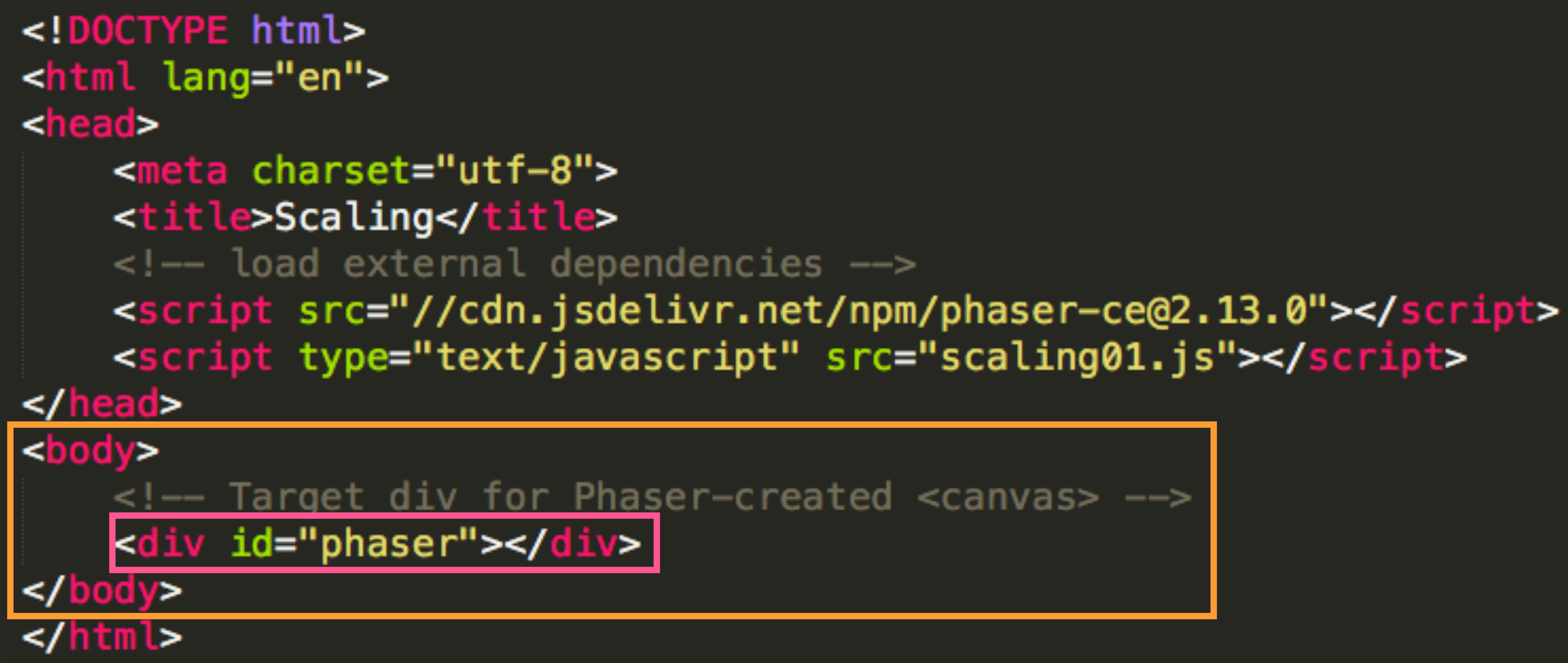
The scaling method used by the ScaleManager when not in fullscreen.

Phaser.ScaleManager.NO_SCALE
The Game display area will not be scaled - even if it is too large for the canvas/screen. This mode *ignores* any applied scaling factor and displays the canvas at the Game size.
Phaser.ScaleManager.EXACT_FIT
The Game display area will be *stretched* to fill the entire size of the canvas's parent element and/or screen. Proportions are not maintained.
Phaser.ScaleManager.SHOW_ALL
Show the entire game display area while *maintaining* the original aspect ratio.
Phaser.ScaleManager.RESIZE
The dimensions of the game display area are changed to match the size of the parent container. That is, this mode *changes the Game size* to match the display size.
Any manually set Game size (see **setGameSize**) is ignored while in effect.

Phaser.ScaleManager.USER_SCALE
The game Display is scaled according to the user-specified scale set by **setUserScale**.
This scale can be adjusted in the **resize callback** for flexible custom-sizing needs.

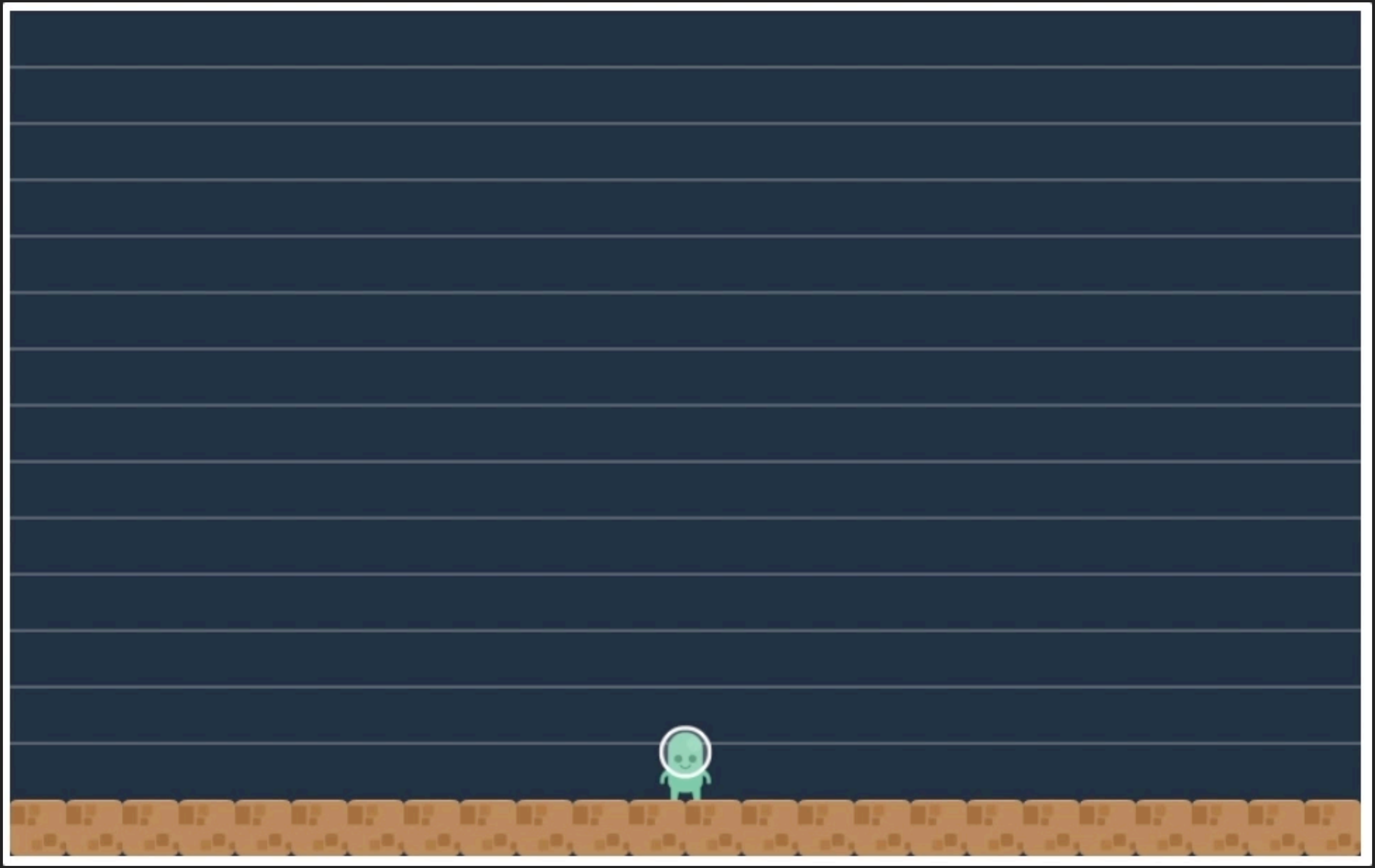
Phaser gives us FIVE options for scaling, of varying usefulness.

BE AWARE OF “PARENT ELEMENTS” 👨‍👨‍👧‍👧

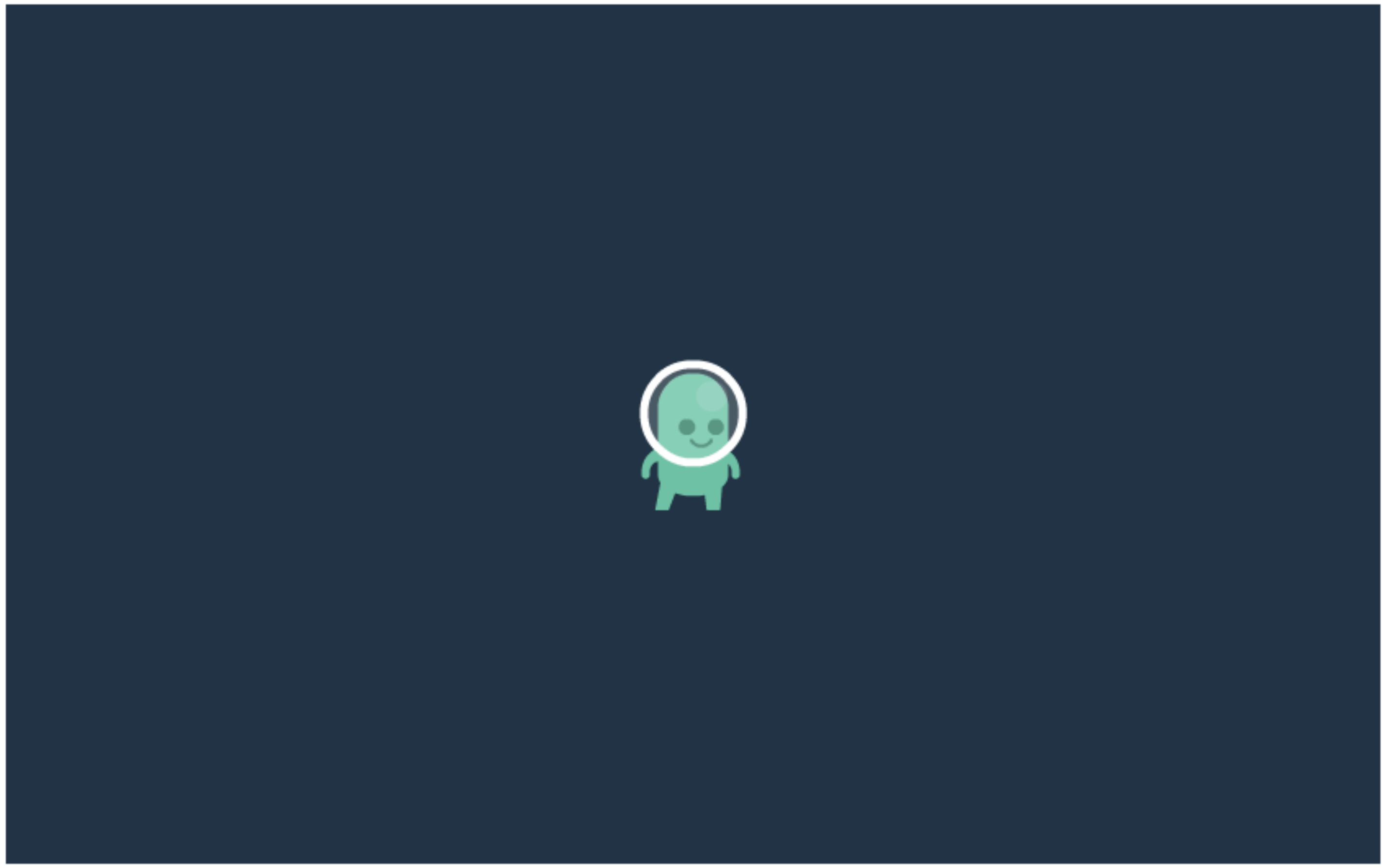


```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>Scaling</title>
    <!-- load external dependencies -->
    <script src="//cdn.jsdelivr.net/npm/phaser-ce@2.13.0"></script>
    <script type="text/javascript" src="scaling01.js"></script>
</head>
<body>
    <!-- Target div for Phaser-created <canvas> -->
    <div id="phaser"></div>
</body>
</html>
```

In this case, <body> is the “phaser” <div>’s parent element.



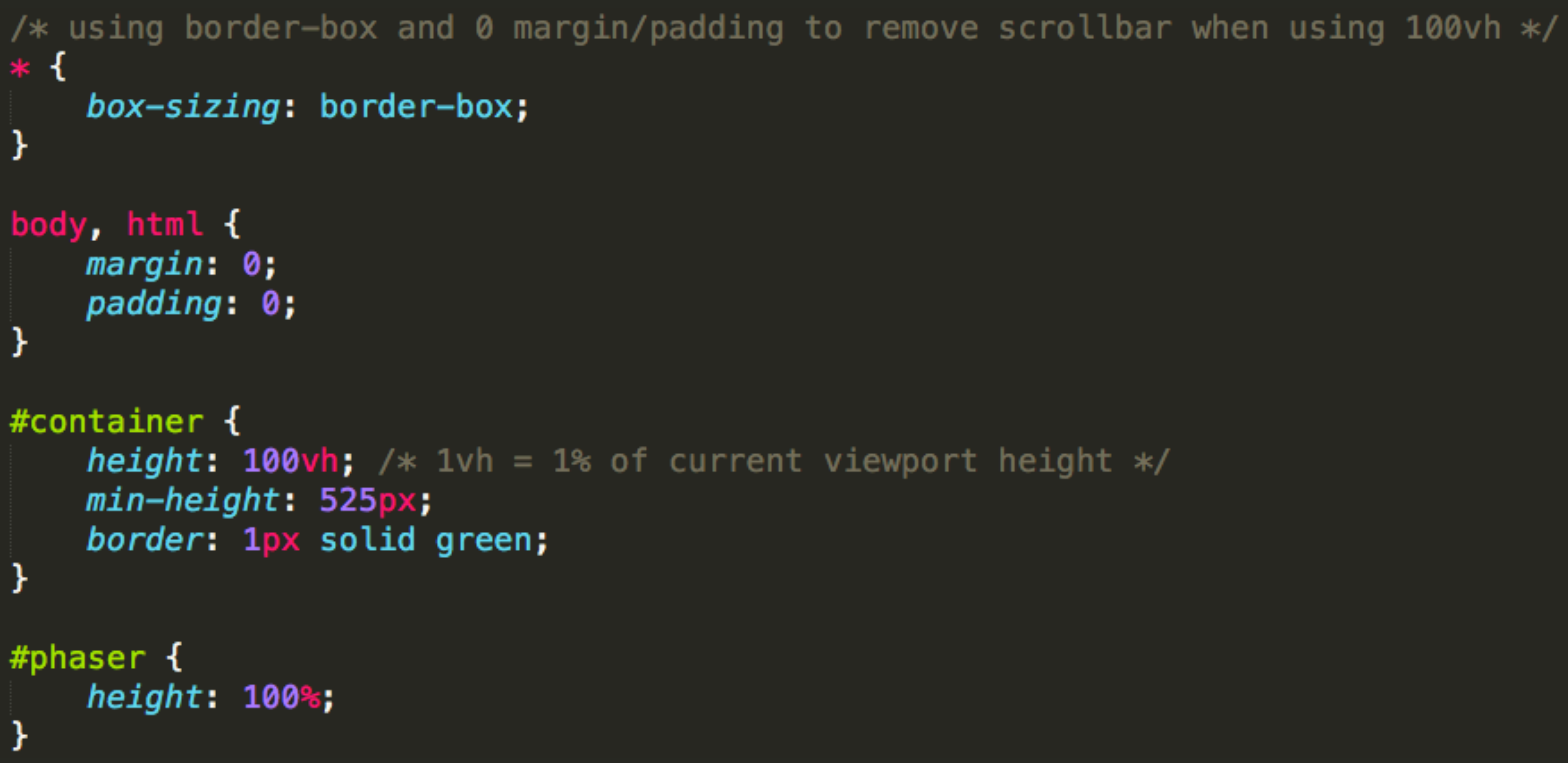
[Let’s check our scaling options in index.html and scaling01.js]



My God, Nathan, can you do horizontal *and* vertical centering in Phaser?

Why YES…

...BUT YOU'LL NEED SOME C$$

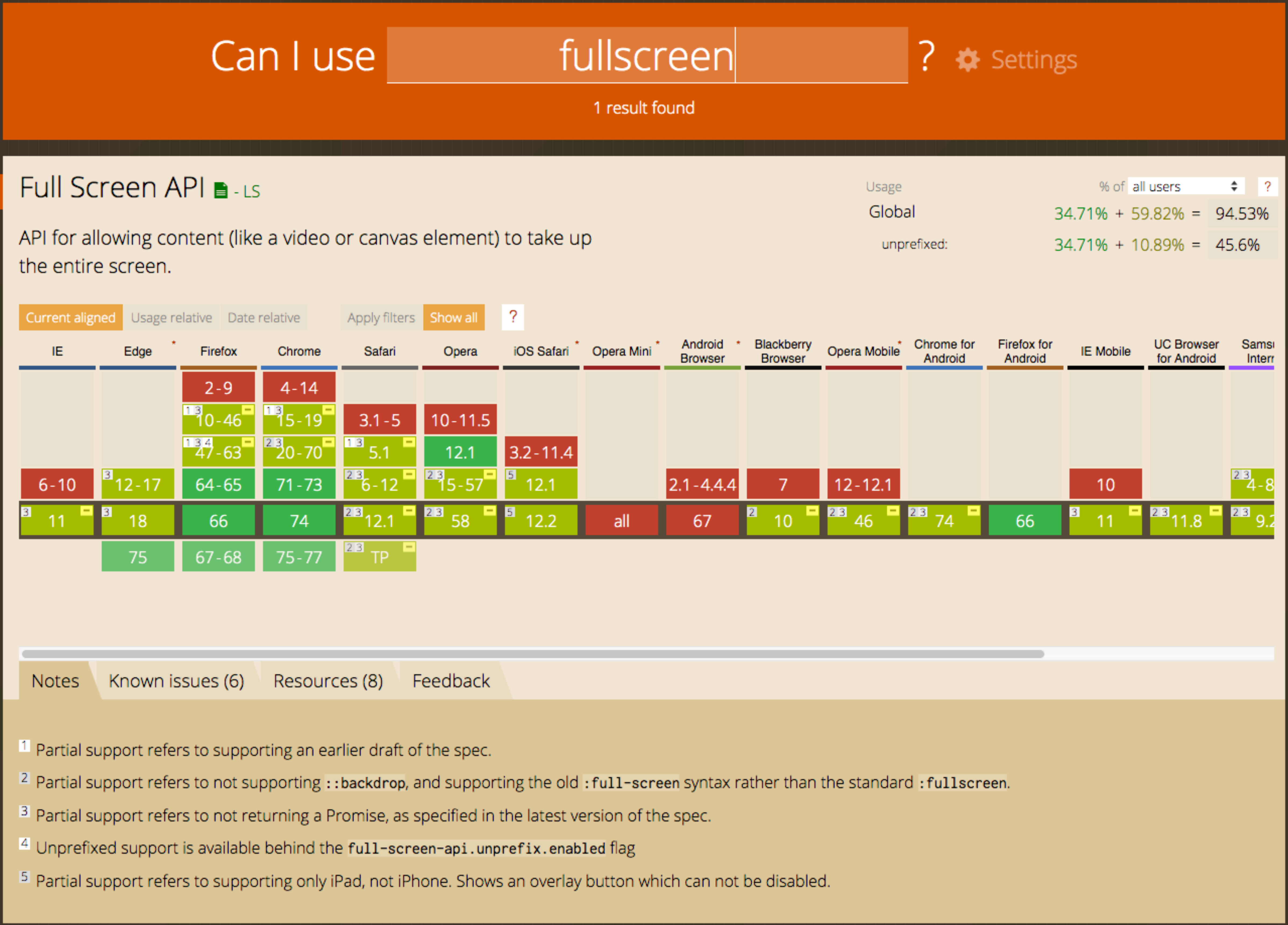


```css
/* using border-box and 0 margin/padding to remove scrollbar when using 100vh */
* {
    box-sizing: border-box;
}

body, html {
    margin: 0;
    padding: 0;
}

#container {
    height: 100vh; /* 1vh = 1% of current viewport height */
    min-height: 525px;
    border: 1px solid green;
}

#phaser {
    height: 100%;
}
```

[Check out index02.html, scaling02.js, and scale02.css]

YOU'RE ON A ROLL, NATHAN…HOW ABOUT FULLSCREEN…?!?!

First we need to check for fullscreen browser compatibility…



Can I use fullscreen ? Settings

1 result found

Full Screen API - LS

API for allowing content (like a video or canvas element) to take up the entire screen.

Usage		% of all users ?
Global	34.71% + 59.82% =	94.53%
unprefixed:	34.71% + 10.89% =	45.6%

Current aligned | Usage relative | Date relative | Apply filters | Show all | ?

IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Blackberry Browser	Opera Mobile *	Chrome for Android	Firefox for Android	IE Mobile	UC Browser for Android	Samsu Interr
		2-9	4-14												
		[1][3] 10-46	[1][3] 15-19	3.1-5	10-11.5										
		[1][3][4] 47-63	[2][3] 20-70	[1][3] 5.1	12.1	3.2-11.4									
6-10	[3] 12-17	64-65	71-73	[2][3] 6-12	[2][3] 15-57	[5] 12.1		2.1-4.4.4	7	12-12.1			10		[2][3] 4-8
[3] 11	[3] 18	66	74	[2][3] 12.1	[2][3] 58	[5] 12.2	all	67	[2] 10	[2][3] 46	[2][3] 74	66	[3] 11	[2][3] 11.8	[2][3] 9.2
	75	67-68	75-77	[2][3] TP											

Notes | Known issues (6) | Resources (8) | Feedback

[1] Partial support refers to supporting an earlier draft of the spec.

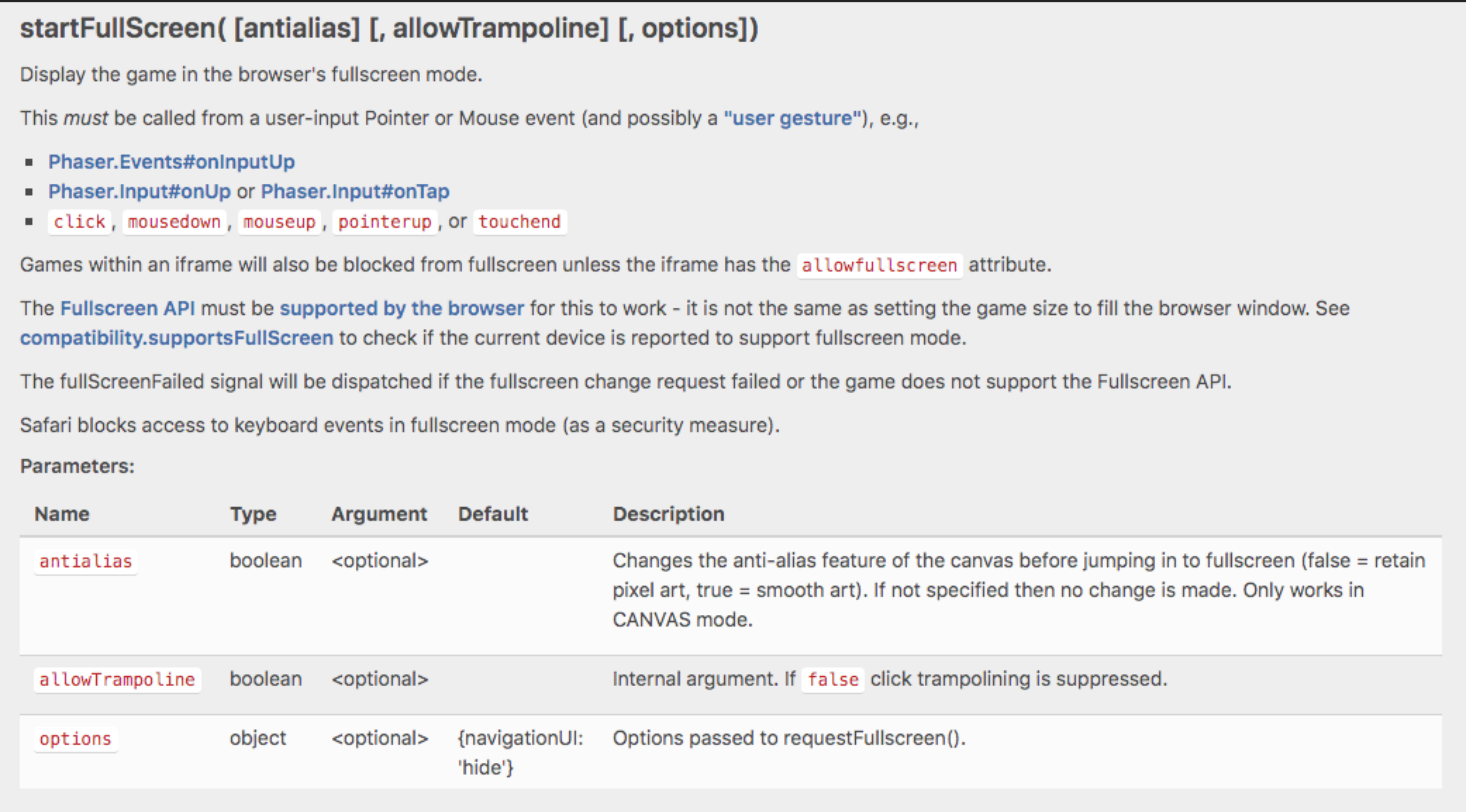
[2] Partial support refers to not supporting `::backdrop`, and supporting the old `:full-screen` syntax rather than the standard `:fullscreen`.

[3] Partial support refers to not returning a Promise, as specified in the latest version of the spec.

[4] Unprefixed support is available behind the `full-screen-api.unprefix.enabled` flag

[5] Partial support refers to supporting only iPad, not iPhone. Shows an overlay button which can not be disabled.

Next, we need to be aware of Phaser’s rules for using the Fullscreen API.



startFullScreen([antialias] [, allowTrampoline] [, options])

Display the game in the browser's fullscreen mode.

This *must* be called from a user-input Pointer or Mouse event (and possibly a **"user gesture"**), e.g.,

- **Phaser.Events#onInputUp**
- **Phaser.Input#onUp** or **Phaser.Input#onTap**
- `click`, `mousedown`, `mouseup`, `pointerup`, or `touchend`

Games within an iframe will also be blocked from fullscreen unless the iframe has the `allowfullscreen` attribute.

The **Fullscreen API** must be **supported by the browser** for this to work - it is not the same as setting the game size to fill the browser window. See **compatibility.supportsFullScreen** to check if the current device is reported to support fullscreen mode.

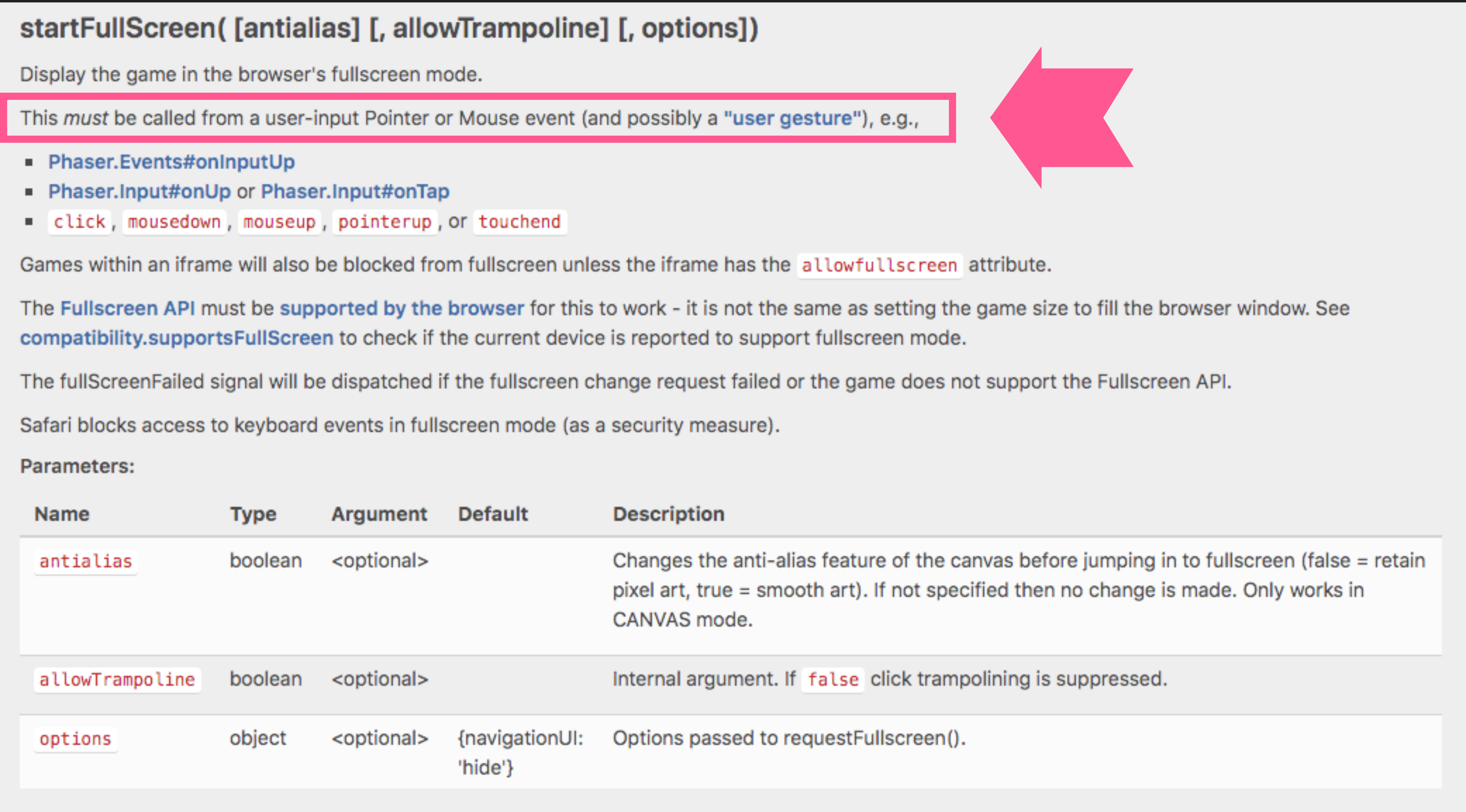
The fullScreenFailed signal will be dispatched if the fullscreen change request failed or the game does not support the Fullscreen API.

Safari blocks access to keyboard events in fullscreen mode (as a security measure).

Parameters:

Name	Type	Argument	Default	Description
`antialias`	boolean	<optional>		Changes the anti-alias feature of the canvas before jumping in to fullscreen (false = retain pixel art, true = smooth art). If not specified then no change is made. Only works in CANVAS mode.
`allowTrampoline`	boolean	<optional>		Internal argument. If `false` click trampolining is suppressed.
`options`	object	<optional>	{navigationUI: 'hide'}	Options passed to requestFullscreen().

This, for instance, is pretty important.



startFullScreen([antialias] [, allowTrampoline] [, options])

Display the game in the browser's fullscreen mode.

This *must* be called from a user-input Pointer or Mouse event (and possibly a **"user gesture"**), e.g.,

- **Phaser.Events#onInputUp**
- **Phaser.Input#onUp** or **Phaser.Input#onTap**
- `click`, `mousedown`, `mouseup`, `pointerup`, or `touchend`

Games within an iframe will also be blocked from fullscreen unless the iframe has the `allowfullscreen` attribute.

The **Fullscreen API** must be **supported by the browser** for this to work - it is not the same as setting the game size to fill the browser window. See **compatibility.supportsFullScreen** to check if the current device is reported to support fullscreen mode.

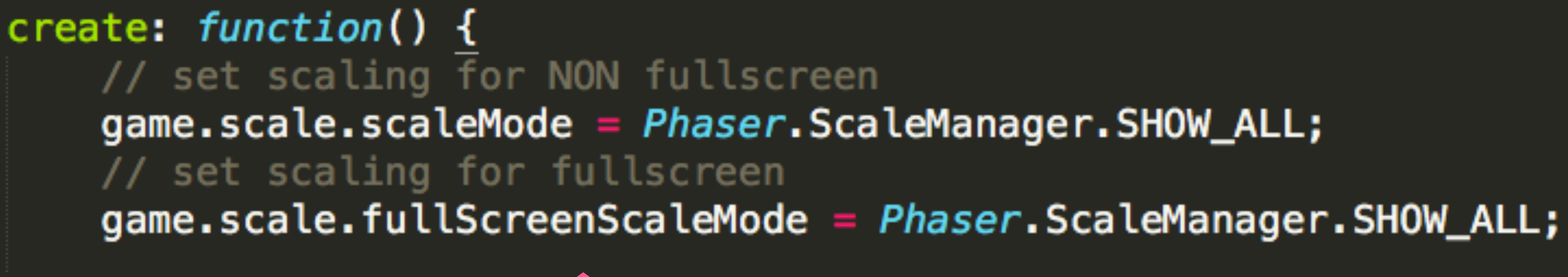
The fullScreenFailed signal will be dispatched if the fullscreen change request failed or the game does not support the Fullscreen API.

Safari blocks access to keyboard events in fullscreen mode (as a security measure).

Parameters:

Name	Type	Argument	Default	Description
`antialias`	boolean	<optional>		Changes the anti-alias feature of the canvas before jumping in to fullscreen (false = retain pixel art, true = smooth art). If not specified then no change is made. Only works in CANVAS mode.
`allowTrampoline`	boolean	<optional>		Internal argument. If `false` click trampolining is suppressed.
`options`	object	<optional>	{navigationUI: 'hide'}	Options passed to requestFullscreen().

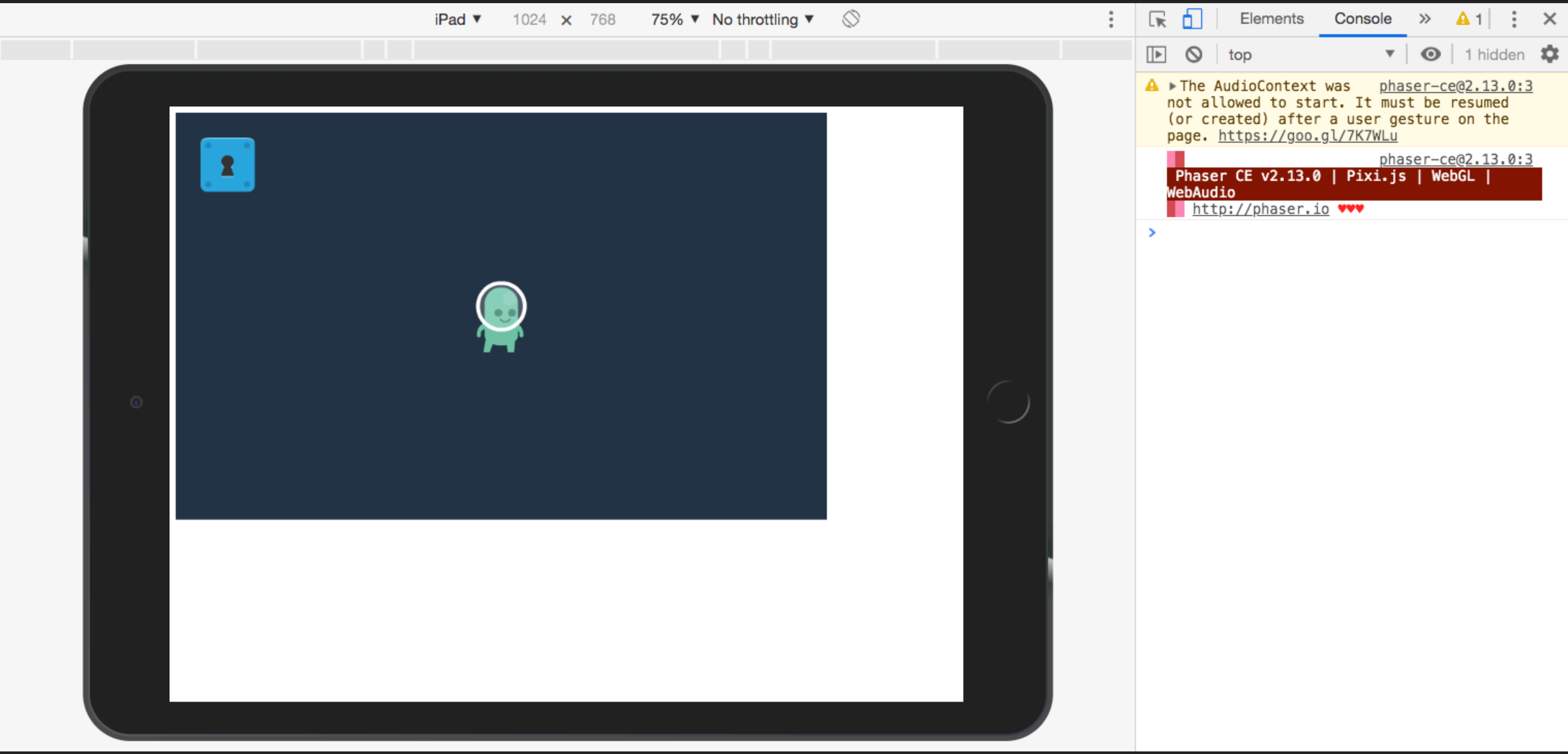
HEY, WATCH OUT, BUDDY



```
create: function() {
    // set scaling for NON fullscreen
    game.scale.scaleMode = Phaser.ScaleManager.SHOW_ALL;
    // set scaling for fullscreen
    game.scale.fullScreenScaleMode = Phaser.ScaleManager.SHOW_ALL;
```

Fullscreen gets its own scaling mode! Phaser, you tricksy hobbits!

Be sure to test on different browsers and devices.



```
⚠ ▸The AudioContext was not allowed to start. It must be resumed (or created) after a user gesture on the page. https://goo.gl/7K7WLu    phaser-ce@2.13.0:3

phaser-ce@2.13.0:3
 Phaser CE v2.13.0 | Pixi.js | WebGL | WebAudio
 http://phaser.io ♥♥♥
>
```

[Hop back to index.html with scaling03.js]