
Aspect Ratio Tools Documentation

Release

Colin McNeil

May 11, 2017

Contents

1	Installation	3
2	Quickstart	5
3	API	7
3.1	Creating and Modifying the Object	7
3.2	Getting Object Information	8
4	Features	9
5	License	11

Aspect Ratio Tools is an extensive toolset for working with images and displays for aspect ratio.

CHAPTER 1

Installation

This is a node package, remember to install node and NPM.

Install Aspect Ratio Tools by running:

```
npm install --save aspect-ratio-tools
```


CHAPTER 2

Quickstart

```
const {AR} = require('aspect-ratio-tools');  
var myAR = new AR(1920,1080);  
console.log(myAR.toString())
```


Creating and Modifying the Object

class **AR** (*width, height*)

The main class that defines an abstract rectangle with a set width and height. All the other parameters, such as aspect ratio are generated from that.

Arguments

- **width** (*number*) – Width of the virtual resolution.
- **height** (*number*) – Height of the virtual resolution.

AR.setDimensions (*width, height*)

Resets the dimensions of the object.

Arguments

- **width** (*number*) – New width of the virtual resolution.
- **height** (*number*) – New height of the virtual resolution.

Returns *void*

AR.scaleWidth (*newHeight*)

Scales the width based on a desired height whilst maintaining aspect ratio.

Arguments

- **newHeight** (*number*) – New height of the virtual resolution.

Returns *void*

AR.scaleHeight (*newWidth*)

Scales the height based on a desired width whilst maintaining aspect ratio.

Arguments

- **newWidth** (*number*) – New width of the virtual resolution.

Returns *void*

AR.scaleDimensions (*scaleRatio*)

Scales the height and width based on a ratio. Example: 100x50 scaled by 2 = 200x100 (effectively 4x area/pixels)

Arguments

- **scaleRatio** (*number*) – Ratio by which to scale. 0.5=50%,1 = 100%,2=200%...

Returns *void*

AR.scaleArea (*scaleRatio*)

Powerful function to scale the dimensions based on area. Example 100x50 scaled by 2 = (141.421356237 x 70.7106781187). The area (total units/pixels) goes from 5000 to 10000.

Arguments

- **scaleRatio** (*number*) – Ratio by which to scale. 0.5=50%,1 = 100%,2=200%...

Returns *void*

Getting Object Information

AR.getAR()

Gets the specific aspect ratio (width/height) of the object.

Returns (*float*) The aspect ratio.

AR.getARString()

Gets the readable aspect ratio ie:(16:9) of the object.

Returns (*string*) The readable aspect ratio.

AR.getWidth()

Gets the width of the object.

Returns (*number*) The width.

AR.getHeight()

Gets height of the object.

Returns (*number*) The height.

AR.toString()

Returns the object in readable form

Returns (*string*) The object's string representation.

Example:

```
Aspect Ratio Object:
Width: 1920
Height: 1080
Aspect Ratio: 1.7777777777777777 (16:9)
```

CHAPTER 4

Features

- Object Oriented nature
- Scaling by area & dimensions
- Getting exact aspect ratio and readable string 1.777777 vs (16:9)
- Resize height or width while maintaining aspect ratio.
- Source Code: github.com/colinmcneil/aspect-ratio-tools

CHAPTER 5

License

The project is licensed under the ISC license.

A

- AR() (class), [7](#)
- AR.getAR() (AR method), [8](#)
- AR.getARString() (AR method), [8](#)
- AR.getHeight() (AR method), [8](#)
- AR.getWidth() (AR method), [8](#)
- AR.scaleArea() (AR method), [8](#)
- AR.scaleDimensions() (AR method), [8](#)
- AR.scaleHeight() (AR method), [7](#)
- AR.scaleWidth() (AR method), [7](#)
- AR.setDimensions() (AR method), [7](#)
- AR.toString() (AR method), [8](#)