

# canvas HTML属性尺寸和CSS尺寸多个细节深入

这篇文章发布于 2018年07月8日, 星期日, 14:22, 归类于 [Canvas相关](#)。阅读 6883 次, 今日 7 次 [8 条评论](#)

by zhangxinxu from <http://www.zhangxinxu.com/wordpress/?p=7769>

本文可全文转载, 但需得到原作者书面许可, 同时保留原作者和出处, 摘要引流则随意。

## 一、 canvas width/height HTML属性值合法性

我们可以使用 `width` 和 `height` 这两个HTML属性控制 `<canvas>` 元素的尺寸。

从定义上讲, `<canvas>` 元素的 `width` 和 `height` 标准属性值只能是整数, 但是实际上, 你设置小数, 浏览器不会认为是不合法的。例如:

```
<canvas height="88.888" style="background:green;"></canvas>
```

实时效果如下:



经测试, 各个主流浏览器下都会渲染为 `88` 像素高度。

或者你自带 `px` 单位, 也是有效的, 例如:

```
<canvas height="88px" style="background:green;"></canvas>
```

实时效果如下:



经测试, 各个主流浏览器下都会渲染为 `88` 像素高度。

但是, 如果是其他单位, 例如 `em`, 则会被忽略, 或者说单位会自动忽略 (包括 `px`) :

```
<canvas height="88em" style="background:green;"></canvas>
```

实时效果如下:



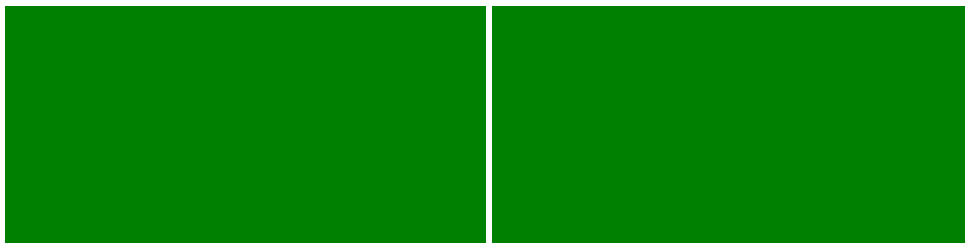
经测试, 各个主流浏览器下都会渲染为 `88` 像素高度。

但是，如果设置的是负数，或者直接不设置，对于高度，则会以 `150` 像素高度呈现；对于宽度，则会以 `300` 像素宽度呈现。

例如下面的 `height` 负值以及直接不设置 `width/height` 属性：

```
<canvas height="-150" style="background:green;"></canvas>
<canvas style="background:green;"></canvas>
```

实时效果如下：



然而，在IE浏览器下，负数尺寸值会被当做 `0` 解析，因此，看不到尺寸。这个和Chrome和Firefox浏览器是不一样的。

### 为什么默认尺寸是300\*150？

为什么 `<canvas>` 元素默认尺寸是 `300*150`，而不是其他尺寸组合呢？这个是CSS规范中定义的，作为替换元素，默认的尺寸是 `300*150`，`<canvas>` 元素就是替换元素，因此默认尺寸是 `300*150`。由于 `<svg>` 元素也是替换元素，因此，`<svg>` 默认的尺寸也是300\*150。

**canvas没有尺寸设置，我们可以通过HTML属性获取吗？**

例如，下面HTML，没有 `width` 和 `height` 设置，请问 `canvas.height` 的返回值是多少？

```
<canvas style="background:green;"></canvas>
```

```
console.log(document.querySelector('canvas').height);
```

结果返回值就是默认的 `150`，而不是 `undefined`。

```
> document.querySelector('canvas').height;
< 150
```

## 二、HTML属性尺寸和CSS尺寸关系

从视觉表现上而言，CSS的 `width`，`height` 属性权重要大于 `<canvas>` 元素的 `width`，`height` 属性权重，例如下面HTML代码，最终 `<canvas>` 元素高度是 `88` 像素。

```
<canvas height="150" style="background:green; height:88px;"></canvas>
```

实时效果如下：



`<canvas>` 元素本质上就是一个图片，其很多样式表现和 `<img>` 元素是一致的，例如这里，CSS控制图片尺寸时候，如果高度或宽度缺省，`<canvas>` 元素依然保持原始的高宽比例。具体描述为：`<canvas>` 元素这里原始尺寸是 `300*150`，CSS设置为 `88px`，最终的宽度表现不是 `300` 而是等比例缩放的 `176px`。

但是，深究下来，和 `<img>` 还是有一点区别，就是 `<canvas>` 的等比例特性是强制的，会忽略HTML属性的设置，但 `<img>` 不会这样。对比下面的测试HTML：

```

<canvas width="300" height="150" style="background:green; height:88px;"></canvas>
```



可以看到 `<img>` 宽度依然是 `300px`，并没有保持原始比例，这就是和 `<img>` 的区别。

### Canvas绘制与CSS尺寸无关

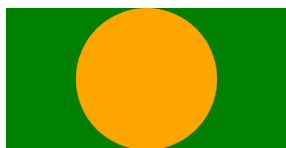
当我们使用各个Canvas API进行图形图像绘制的时候，其坐标位置，尺寸大小等都是相对于 `<canvas>` 元素的HTML属性 `height`，与CSS `height` 没有任何关系。例如：

```
<canvas height="150" style="background:green; height:88px;"></canvas>
```

我们想以 `<canvas>` 元素中心点为圆心画一个圆，则垂直坐标应该是 `75`，是HTML `height` 属性值的一半，而非最终视觉展示高度的一半。

```
var canvas = document.querySelector('canvas');
var context = canvas.getContext('2d');
context.fillStyle = 'orange';
context.arc(150, 75, 75, 0, 2 * Math.PI);
context.fill();
```

实时效果如下：



`<canvas>` 元素本质上是位图，因此，在retina高密度屏幕下，如果绘制图像，则如果按照视觉尺寸来绘制，可能会模糊，我们可以将 `<canvas>` 元素的尺寸用HTML高宽属性设置为2倍尺寸，然后使用CSS高宽属性设置视觉尺寸为布局需要的尺寸大小。

## 三、结束语

就放个笑脸吧~



(本篇完) // 想要打赏? 点击[这里](#)。有话要说? 点击[这里](#)。



« [小tips: touch-action简介与treated as passive错误解决](#)

[实力科普：为什么浮层或弹框一定要有叉叉关闭按钮？](#) »

#### 猜你喜欢

- CSSOM视图模式(CSSOM View Module)相关整理
- HTML textarea cols,rows属性和宽度高度关系研究
- 鲜为人知的一个解决兼容性问题的利器——小数
- CSS实现圆角六色渐变自适应按钮详解
- CSS/CSS3长度、时间、频率、角度单位大全
- CSS流体(自适应)布局下宽度分离原则
- 翻译：CSS中的槽粕
- 理解CSS3 max/min-content及fit-content等width值
- 深入理解CSS的width:auto
- css行高line-height的一些深入理解及应用
- 贝塞尔曲线与CSS3动画、SVG和canvas的基情

分享到： 1

标签： canvas, height, px, width

#### 发表评论（目前8条评论）

<input type="text"/>	名称 (必须)
<input type="text"/>	邮件地址(不会被公开) (必须)
<input type="text"/>	网站
<div></div>	
<input type="button" value="提交评论"/>	

1. 扎肾了老铁说道：  
2018年07月30日 15:12

张大，有个问题我困扰已久。请问你对各个版本浏览器，甚至ie各种低版本的浏览器的测试是怎么完成的?

[回复](#)



2.

LovLong悸动说道：  
2018年07月27日 09:37  
  
哎，知道的越多，越发现自己越无知？  
  
[回复](#)


3.

Arther说道：  
2018年07月13日 09:34  
  
学习了  
  
[回复](#)


4.

广建说道：  
2018年07月10日 15:59  
  
理清楚最重要  
  
[回复](#)


5.

Kouleason说道：  
2018年07月9日 11:22  
  
长知识, 一直以来都感觉canvas的width 和height就很坑,今天总算明了  
  
[回复](#)


6.

lalala说道：  
2018年07月9日 08:32  
  
我想知道有些非常复杂的canvas都是自己一点一点写出来的吗？还是有什么生成公式....  
  
[回复](#)

严文彬说道：  
2018年07月9日 09:50  
  
想多了，就像很多人认为HTML也是根据设计图自动生成一样  
  
[回复](#)

DeathGhost说道：  
2018年07月9日 13:50  
  
很少吧  
  
[回复](#)







## 最新文章

- » [常见的CSS图形绘制合集](#)
  - » [粉丝群第1期CSS小测点评与答疑](#)
  - » [分享三个纯CSS实现26个英文字母的案例](#)
  - » [小tips: 纯CSS实现打字动画效果](#)
  - » [CSS/CSS3 box-decoration-break属性简介](#)
  - » [CSS :placeholder-shown伪类实现Material Design占位符交互效果](#)
  - » [从天猫某活动视频不必要的3次请求说起](#)
  - » [CSS vector-effect与SVG stroke描边缩放](#)
  - » [CSS ::backdrop伪元素是干嘛用的？](#)
  - » [周知：CSS -webkit-伪元素选择器不再导致整行无效](#)

## 今日热门

- » [常见的CSS图形绘制合集](#) (179)

- » [粉丝群第1期CSS小测点评与答疑](#) <sup>(113)</sup>
- » [未来必热：SVG Sprite技术介绍](#) <sup>(111)</sup>
- » [HTML5终极备忘大全（图片版+文字版）](#) <sup>(85)</sup>
- » [让所有浏览器支持HTML5 video视频标签](#) <sup>(83)</sup>
- » [Selectivizr-让IE6~8支持CSS3伪类和属性选择器](#) <sup>(80)</sup>
- » [CSS3下的147个颜色名称及对应颜色值](#) <sup>(78)</sup>
- » [小tips: 纯CSS实现打字动画效果](#) <sup>(73)</sup>
- » [写给自己看的display: flex布局教程](#) <sup>(70)</sup>
- » [分享三个纯CSS实现26个英文字母的案例](#) <sup>(70)</sup>

#### 今年热议

- » [《CSS世界》女主角诚寻靠谱一起奋斗之人](#) <sup>(76)</sup>
- » [不借助Echarts等图形框架原生JS快速实现折线图效果](#) <sup>(64)</sup>
- » [看，for..in和for..of在那里吵架！](#) <sup>(60)</sup>
- » [是时候好好安利下LuLu UI框架了！](#) <sup>(47)</sup>
- » [原来浏览器原生支持JS Base64编码解码](#) <sup>(35)</sup>
- » [妙法攻略：渐变虚框及边框滚动动画的纯CSS实现](#) <sup>(33)</sup>
- » [炫酷H5中序列图片视频化播放的高性能实现](#) <sup>(31)</sup>
- » [CSS scroll-behavior和JS scrollToView让页面滚动平滑](#) <sup>(30)</sup>
- » [windows系统下批量删除OS X系统.DS\\_Store文件](#) <sup>(26)</sup>
- » [写给自己看的display: flex布局教程](#) <sup>(26)</sup>

#### 猜你喜欢

- [CSSOM视图模式\(CSSOM View Module\)相关整理](#)
- [HTML textarea cols,rows属性和宽度高度关系研究](#)
- [鲜为人知的一个解决兼容性问题的利器——小数](#)
- [CSS实现圆角六色渐变自适应按钮详解](#)
- [CSS/CSS3长度、时间、频率、角度单位大全](#)
- [CSS流体\(自适应\)布局下宽度分离原则](#)
- [翻译：CSS中的糟粕](#)
- [理解CSS3 max/min-content及fit-content等width值](#)
- [深入理解CSS的width:auto](#)
- [css行高line-height的一些深入理解及应用](#)
- [贝塞尔曲线与CSS3动画、SVG和canvas的基情](#)