

Use Stack Overflow for Teams at work to find answers in a private and secure environment. Get your first 10 users free. [Sign up.](#)

## Dynamically updating css in Angular 2

Asked 3 years, 6 months ago   Active 10 months ago   Viewed 142k times



Let's say I have an Angular2 Component

88

*//home.component.ts*



```
import { Component } from 'angular2/core';
```



21

```
@Component({
  selector: "home",
  templateUrl: "app/components/templates/home.component.html",
  styleUrls: ["app/components/styles/home.component.css"]
})
export class HomeComponent {
  public width: Number;
  public height: Number;
}
```

The template html file for this component

*//home.component.html*

```
<div class="home-component">Some stuff in this div</div>
```

And finally the css file for this component

*//home.component.css*

```
.home-component{
  background-color: red;
  width: 50px;
  height: 50px;
}
```

As you can see there are 2 properties in the class, `width` and `height`. I would like the css styles for width and height to match the values of the width and height properties and when the properties update, the width and height of the div update. What is the proper way to accomplish this?

css



asked Mar 9 '16 at 4:05



tt9

2,094

1

26

48

## 10 Answers



Try this

215



```
<div class="home-component"
[style.width.px]="width"
[style.height.px]="height">Some stuff in this div</div>
```



[Updated]: To set in % use

```
[style.height.%]="height">Some stuff in this div</div>
```

edited Nov 12 '17 at 7:44

answered Mar 9 '16 at 8:32



Gaurav Mukherjee

4,200

1

18

28

14 How to add width in %? – [sudheer KB](#) Jan 18 '17 at 9:20

6 See Helen solution below – [Sebastien DErrico](#) May 16 '17 at 19:06

how about if I want to set with image background? `[style.background-image]` ? – [Nere](#) Jan 11 '18 at 8:00



To use % instead of px or em with [@Gaurav's](#) answer, it's just

45

```
<div class="home-component" [style.width.%]="80" [style.height.%]="95">  
Some stuff in this div</div>
```

answered Feb 13 '17 at 1:44



Helen

502 4 11

---

or..... [style.width]='50+%"' incase you switch between px and % base. – Zze Jul 11 '17 at 1:47

---

▲ This should do it:

16

```
<div class="home-component"  
  [style.width]="width + 'px'"  
  [style.height]="height + 'px'">Some stuff in this div</div>
```

answered Mar 9 '16 at 4:09



Sasxa

30.2k 11 76 95

▲ You can also use hostbinding:

9

```
import { HostBinding } from '@angular/core';  
  
export class HomeComponent {  
  @HostBinding('style.width') width: Number;  
  @HostBinding('style.height') height: Number;  
}
```

Now when you change the width or height property from within the HomeComponent, this should affect the style attributes.

edited Dec 13 '17 at 14:12

answered Nov 25 '16 at 15:46



11mb

1,088 2 14 30

[Check working Demo here](#)

6

```
import {Component, bind} from 'angular2/core';
import {bootstrap} from 'angular2/platform/browser';
import {FORM_DIRECTIVES} from 'angular2/form';

import {Directive, ElementRef, Renderer, Input, ViewChild, AfterViewInit} from
'angular2/core';

@Component({
  selector: 'my-app',
  template: `
    <style>
      .myStyle{
        width:200px;
        height:100px;
        border:1px solid;
        margin-top:20px;
        background:gray;
        text-align:center;
      }
    </style>

    <div [class.myStyle]="my" [style.background-color]="randomColor"
[style.width]="width+'px'" [style.height]="height+'px'"> my width={{width}} & height=
{{height}}</div>
  `,
  directives: []
})

export class AppComponent {
  my:boolean=true;
  width:number=200px;
  height:number=100px;
  randomColor;
  randomNumber;
  intervalId;
  textArray = [
    'blue',
    'green',
    'yellow',
    'orange',
    'pink'
  ];
};
```

```

constructor()
{
  this.start();
}

start()
{
  this.randomNumber = Math.floor(Math.random()*this.textArray.length);
  this.randomColor=this.textArray[this.randomNumber];
  console.log('start' + this.randomNumber);
  this.intervalId = setInterval(()=>{
    this.width=this.width+20;
    this.height=this.height+10;
    console.log(this.width + " " + this.height)
    if(this.width==300)
    {
      this.stop();
    }

  }, 1000);
}
stop()
{
  console.log('stop');
  clearInterval(this.intervalId);
  this.width=200;
  this.height=100;
  this.start();
}
}

bootstrap(AppComponent, []);

```

answered Mar 9 '16 at 5:23

[micronyks](#)

39.8k

12

82

117



The accepted answer is not incorrect.

6

For grouped styles one can also use the ngStyle directive.

`<some-element [ngStyle]='{'font-style': styleExpression, 'font-weight': 12}'>...</some-element>`

The official docs are [here](#)

answered Apr 13 '18 at 7:58



JoshuaTree

536 5 17

---

And how does that solve OPs problem exactly? – Grmpfhmb1 Sep 6 '18 at 15:40

---

You can dynamically change the style(width and height) of div by attaching dynamic value to inline [style.width] and [style.hiegh] property of div.

5

In your case you can bind width and height property of HomeComponent class with the div's inline style width and height property like this... **As directed by Sasxa**

```
<div class="home-component"
  [style.width]="width + 'px'"
  [style.height]="height + 'px'">Some stuff in this div
</div>
```

For the working demo take a look at this plunker(<http://plnkr.co/edit/cUbbo2?p=preview>)

```
//our root app component
import {Component} from 'angular2/core';
import {FORM_DIRECTIVES, FormBuilder, AbstractControl, ControlGroup,} from
"angular2/common";

@Component({
  selector: 'home',
  providers: [],
  template: `
    <div class="home-component" [style.width]="width+'px'"
[style.height]="height+'px'">Some this div</div>
    <br/>
    <form [ngFormModel]="testForm">
      width:<input type="number" [ngFormControl]="txtWidth"/> <br>
      Height:<input type="number"[ngFormControl]="txtHeight" />
```

```

    </form>
  `,
  styles:[`
    .home-component{
      background-color: red;
      width: 50px;
      height: 50px;
    }
  `],
  directives: [FORM_DIRECTIVES]
})
export class App {
  testForm:ControlGroup;
  public width: Number;
  public height: Number;
  public txtWidth:AbstractControl;
  public txtHeight:AbstractControl;

  constructor(private _fb:FormBuilder) {
    this.testForm=_fb.group({
      'txtWidth':['50'],
      'txtHeight':['50']
    });

    this.txtWidth=this.testForm.controls['txtWidth'];
    this.txtHeight=this.testForm.controls['txtHeight'];

    this.txtWidth.valueChanges.subscribe(val=>this.width=val);
    this.txtHeight.valueChanges.subscribe(val=>this.height =val);
  }
}

```

answered Mar 9 '16 at 6:11



Jorin

496 1 8 16

▲ All the above answers are great. But if you were trying to find a solution that won't change the html files below is helpful

4

```

ngAfterViewChecked(){
  this.renderer.setStyle(targetItem.nativeElement, 'height', textHeight+"px");
}

```

You can import `renderer` from `import {Renderer} from '@angular/core';`

answered Feb 23 '17 at 11:05



WenhaoWu

70 1 10

1 Since this answer was posted, `Renderer` was deprecated. Use `Renderer2` instead, and replace `setElementStyle()` by `setStyle()` – Chris Feb 11 at 0:37



2



I liked the look of **WenhaoWu**'s idea above, but I needed to identify the div with **class** `.ui-tree` in the **PrimeNG tree component** to set the height dynamically. All the answers I could find required the div to be named (ie `#treediv`) to enable the use of `@ViewChild()`, `@ViewChildren()`, `@ContentChild()`, `@ContentChildren()` etc. This was messy with a third party component.

I finally found a snippet from [Günter Zöchbauer](#) :

```
ngAfterViewInit() {
  this.elRef.nativeElement.querySelector('.myClass');
}
```

This made it easy:

```
@Input() height: number;
treeheight: number = 400; //default value

constructor(private renderer: Renderer2, private elRef: ElementRef) { }

ngOnInit() {
  this.loading = true;
  if (this.height != null) {
    this.treeheight = this.height;
  }
}

ngAfterViewInit() {
  this.renderer.setStyle(this.elRef.nativeElement.querySelector('.ui-tree'), 'height',
    this.treeheight + "px");
}
```

edited Oct 15 '18 at 15:16  
Juan Castillo

answered Sep 22 '17 at 1:57  
davaus





3,726 3 30 38



703 9 14



If you want to set width dynamically with variable than use [] braces instead {{}}:

2

```
<div [style.width.px]="[widthVal]" [style.height.px]="[heightVal]"></div>
```



```
<div [style.width.%]="[widthVal]" [style.height.%]="[heightVal]"></div>
```

answered Nov 1 '18 at 5:54

**Govind Samrow**  
5,772 10 33 64