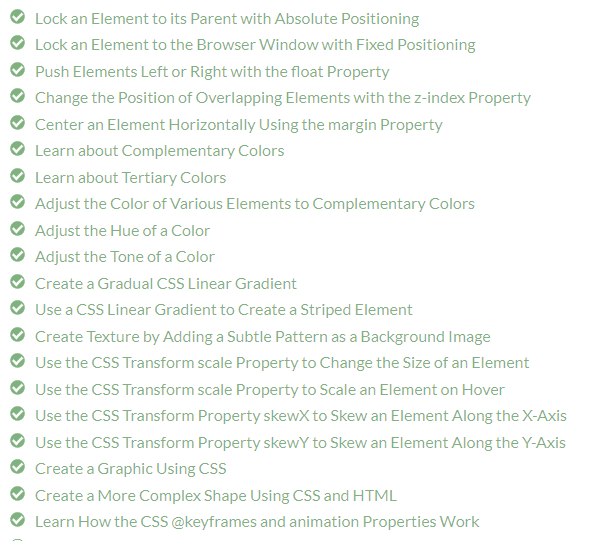
**2018/9/3 Day11**



**11，absolute position 相对于relative position的位置**

Applied Visual Design: Lock an Element to its Parent 
with Absolute Positioning 
The next option for the CSS position property is absolute , which locks the element in place 
relative to its parent container. Unlike the relative position, this removes the element from the 
normal flow of the document, so surrounding items ignore it. The CSS offset properties (top or 
bottom and left or right) are used to adjust the position. 
One nuance with absolute positioning is that it will be locked relative to its closest positioned 
ancestor. If you forget to add a position rule to the parent item, (this is typically done using 
position: relative; ), the browser will keep looking up the chain and ultimately default to the 
body tag. 

parent container 是 section部分， section部分就是<section>与</section>之间的所有段落之间的内容，可以把它看成一个整块

2 
*searchbar { 
/*position: absolute; Y/ 
top: Sapx; 
right: Sepx; 
section { 
position: relative; 
>We1 come ! 
Welcome! 
Search: 
<section> 
<form id 
<label 
<input 
<input 
section> 
/ bodyfA 
" search" >Search 
name=" search" > 

enable *position: absolute；* 后, <form></form>的位置就是相对于section的位置了

2 
11 
12 
21 
*searchbar { 
position: absolute; 
top: Sapx; 
right: Sepx; 
section { 
position: relative; 
<body> 
<hl >We1 come ! 
Welcome! 
Search: 
Go! 
<section> 
<form id 
<label 
<input 
<input 
section> 
< / body > 
for= " search" >Search 
name=" search" > 
name="submit" " > 

*注意理解right:50px 是指的absolute相对于relative右侧距离，是50px*

**11，float ???**

**12, z-index 调节重叠部分哪一个在上**

Applied Visual Design: Change the Position of 
Overlapping Elements with the z-index Property 
When elements are positioned to overlap, the element coming later in the HTML markup will, by 
default, appear on the top of the other elements. However, the z- index property can specify the 
order of how elements are stacked on top of one another. It must be an integer (i.e. a whole number 
and not a decimal), and higher values for the z -index property of an element move it higher in the 
stack than those with lower values. 

本来是蓝色在上，因为先是有蓝色的HTML代码

28 
22 
23 
<style> 
div { 
t,'idth: 68%, 
height: 2eapx; 
margin-top: 28px; 
.first { 
background-color: red; 
position: absolute; 
z-index: 2; 
. second { 
background-color: blue; 
position: absolute; 
left: 48px; 
top: Sapx; 
z-index: 1, 
<div 
<div class=" 

**13, using hsl() to adjust hue, saturation, and lightness**

<style> 
body { 
background -color : 
. green { 
background -color : 
. cyan { 
background -color : 
.blue { 
background -color : 
div { 
hs1(12ß, 
hs1(18ß, 
hs1(24ß, 
188%, 
188%, 
188%, 
sex) 
sex) 
sex) 
<div 
<div 
<div 
display: inline-block; 
height: leapx; 
t,'idth: leepx; 
class=" 
class=" cyan" 

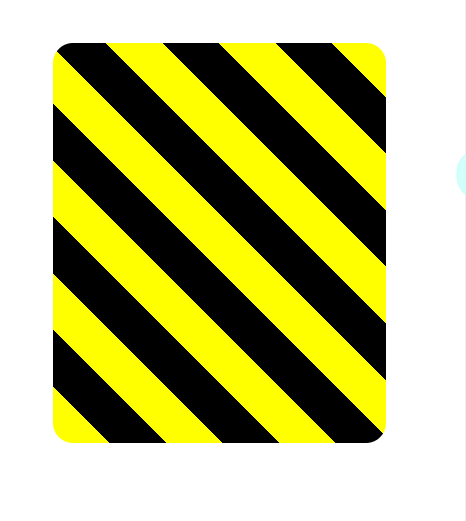
**14, linear-gradient (deg, color1, color2, ….)**

*来自 <*[*https://learn.freecodecamp.org/responsive-web-design/applied-visual-design/create-a-gradual-css-linear-gradient*](https://learn.freecodecamp.org/responsive-web-design/applied-visual-design/create-a-gradual-css-linear-gradient)*>*

<https://developer.mozilla.org/zh-CN/docs/Web/CSS/linear-gradient>

**15, striped color 色条**

freeCodeCamp (b) 
Applied Visual Design: Use a CSS Linear Gradient to 
Create a Striped Element 
The repeating-linear-gradient() function is very similar to linear-gradient() with the 
major difference that it repeats the specified gradient pattern. repeating-linear-gradient( ) 
accepts a variety of values, but for simplicity, you'll work with an angle value and color stop 
values in this challenge. 
The angle value is the direction of the gradient. Color stops are like width values that mark where 
a transition takes place, and are given with a percentage or a number of pixels. 
In the example demonstrated in the code editor, the gradient starts with the color yellow at O 
pixels which blends into the second color blue at 40 pixels away from the start. Since the next 
color stop is also at 40 pixels, the gradient immediately changes to the third color green , which 
itself blends into the fourth color value red as that is 80 pixels away from the beginning of the 
gradient. 
For this example, it helps to think about the color stops as pairs where every two colors blend 
together. 
ßpx [yellow 
bl end 
blue] 4Øpx [green 
blend 
red] 8Øpx 
If every two color stop values are the same color, the blending isn't noticeable because it's 
between the same color, followed by a hard transition to the next color, so you end up with 
stripes. 
Make stripes by changing the repeating- linear-gradient() to use a gradient angle of 45deg , 
then set the first two color stops to yellow , and finally the second two color stops to black . 
div{ 
border-radius: 28px; 
t,'idth : 
height: 
4eapx; 
58 auto; 
margin: 
background: repeating-linear-gradient( 
45deg, 
yellow epx, 
yellow 48px, 
black 4epx, 
black 8epx 
17 
19 
// running test 
// tests completed 



**16, 加背景图片 background: url(** *link address* **)**

body { 
background : 
running test 
tests completed 
url( https: //i . imgur.com/MJAkxbh.ug); 

**17, change the size of an element tansform:scale( *times* )**

.ball { 
t,'idth: 4epx; 
height: 4epx; 
margin: Se auto; 
position: fixed; 
background: linear-gradient( 
14 
17 
26 
35deg, 
#ffcccc 
border-radius : 
#ba111 { 
left: 28%; 
#ba112 { 
left: 65%; 
€0 
an sform : 
sax, 
<div class= 
<div class= 
"ball" 
"ball" 

**18， change the size of an element when on hover**

Curriculum 
div { 
t,'idth: 78%, 
height: leapx; 
margin: Sapx auto; 
background: linear-gradient( 
S3deg, 
#ccfffc, 
#ffcccf 
1 v : hover 
transform: scale(l.l) 
Forum 
17 

**19, change the skew of an element**

X-Axis

freeCodeCamp (b) 
Q 
Seerch 8.000+ lessons. articles. Encl videos 
Applied Visual Design: Use the CSS Transform 
Property skewX to Skew an Element Along the X-Axis 
The next function of the transform property is skewX() , which skews the selected element along 
its X (horizontal) axis by a given degree. 
The following code skews the paragraph element by -32 degrees along the X-axis. 
transform: skewX(-32deg) 
Skew the element with the id of bottom by 24 degrees along the X-axis by using the tran sform 
property. 
Run the Tests 
div { 
t,'idth : 
height: 
margin: 
*top { 
leapx; 
58px auto; 
14 
16 
17 
background -color: red; 
*bottom { 
background-color: blue; 
transform: skewX(24deg) 
<div 
<div >k/divA 

div { 
t,'idth: 78%, 
height: leapx; 
margin: Sapx auto; 
*top { 
background -color: red; 
*bottom { 
background-color: blue; 
transform: skewX(24deg) 
14 
16 <div 
17 <div >k/divA 

Y-Axis

freeCodeCamp (b) 
Applied Visual Design: Use the CSS Transform 
Property skewY to Skew an Element Along the Y-Axis 
Given that the skewX() function skews the selected element along the X-axis by a given degree, it is 
no surprise that the skewY() property skews an element along the Y (vertical) axis. 
Skew the element with the id of top -10 degrees along the Y-axis by using the transform property. 
Run the Tests 
Reset All Code 
Get a hint 
Ask for help 
The element with id top should be skewed by -10 degrees along its Y-axis. 

11 
IS 
17 
18 
div { 
t,'idth: 78%, 
height: leapx; 
margin: Sepx auto; 
*top { 
background-color: red; 
transform: skewYi( ledeg)l 
*bottom { 
background -color: blue; 
transform: skewX(24deg) ; 
<div 
<div 

**20, 在原来的对象旁边加上两个矩形（radius可以让它变圆）**

2 . heart { 
position: absolute; 
4 
13 
14 
24 
33 
34 
margin: auto; 
top: e; 
right: e; 
bottom: a, 
left: 8; 
background-color: 
height: 58px; 
width: Sapx; 
transform: 
. heart: : after 
background-color: 
content . 
pi nk; 
1 ue; 
border-radius: 25 
position: absolute; 
nidth: Sapx; 
height: 5øp)gi 
top: apx; 
left: SSpx 
. heart: : before 
content: " 
background-color: black 
border-radius: 58%; 
position: absolut • 
nidth: Sapx; 
;j 
height: 5ßpx 
top: 55px; 
left: 
<div class 

**21， @Keyframes**

animation-name sets the name of the animation, which is later used by@keyframes to tell CSS 
which rules go with which animations. 
animation - duration sets the length of time for the animation. 
@keyframes is how to specify exactly what happens within the animation over the duration. This 
is done by giving CSS properties for specific "frames" during the animation, with percentages 
ranging from 0% to 100%. If you compare this to a movie, the CSS properties for 0% is how the 
element displays in the opening scene. The CSS properties for 100% is how the element appears 
at the end, right before the credits roll. Then CSS applies the magic to transition the element over 
the given duration to act out the scene. Here's an example to illustrate the usage of@keyframes 
and the animation properties: 
#anim 
animati on-name: col orful 
animation-duration: as; 
Okeyframes col orful 
background-col or : 
background-col or : 
blue, 
yellow; 

开始 0% ：背景变蓝色，

结束 100% ：背景变黄色