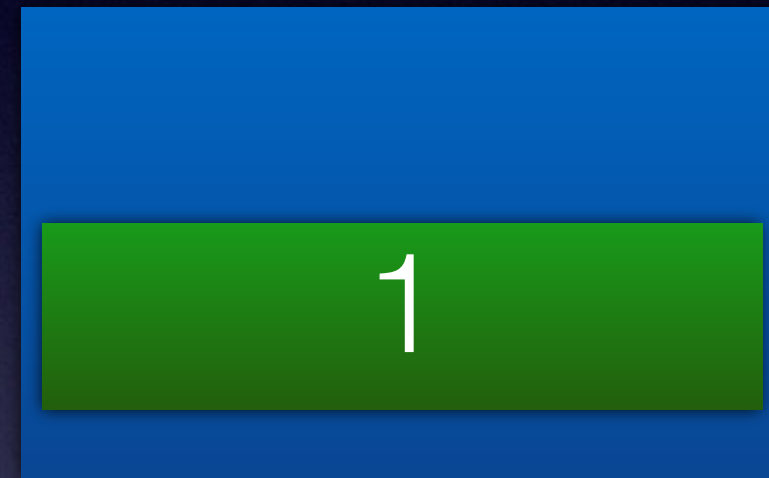


Technical Notes

Assignment 2 2015

Advanced Web Development

Positioning Doors on Animation Page



```
<li>
```

```
<div class = "door">
```

```
1
```

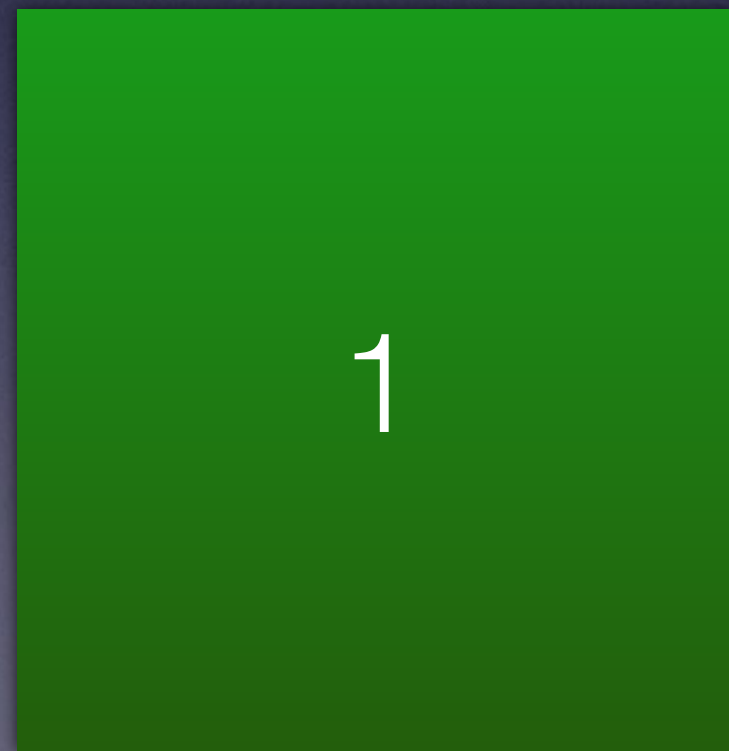
```
</div>
```

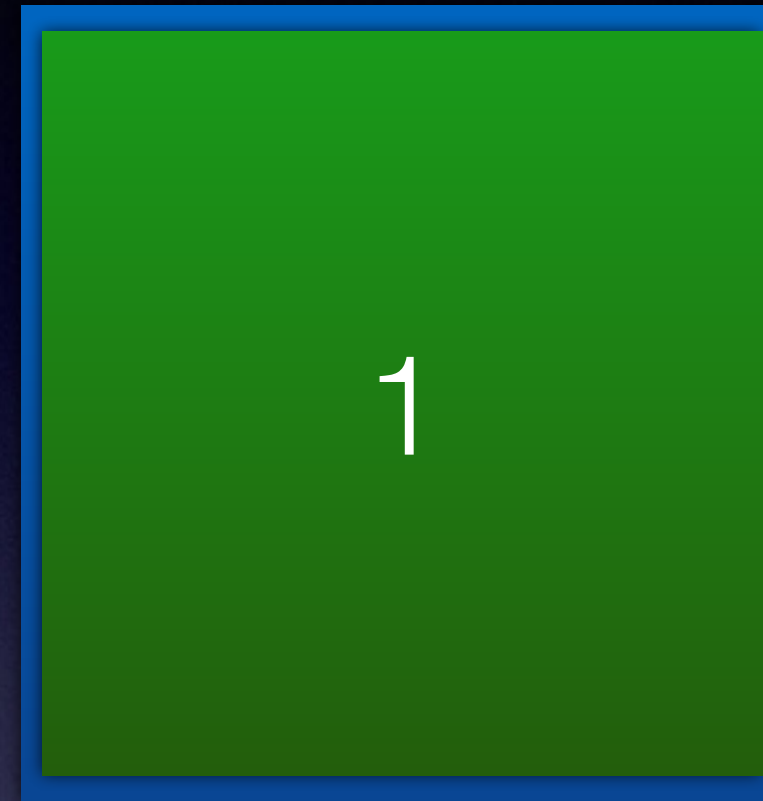
```
</li>
```

The door element (containing the date) is ***inside*** the `` element.



Use CSS to make
both elements the
same size.



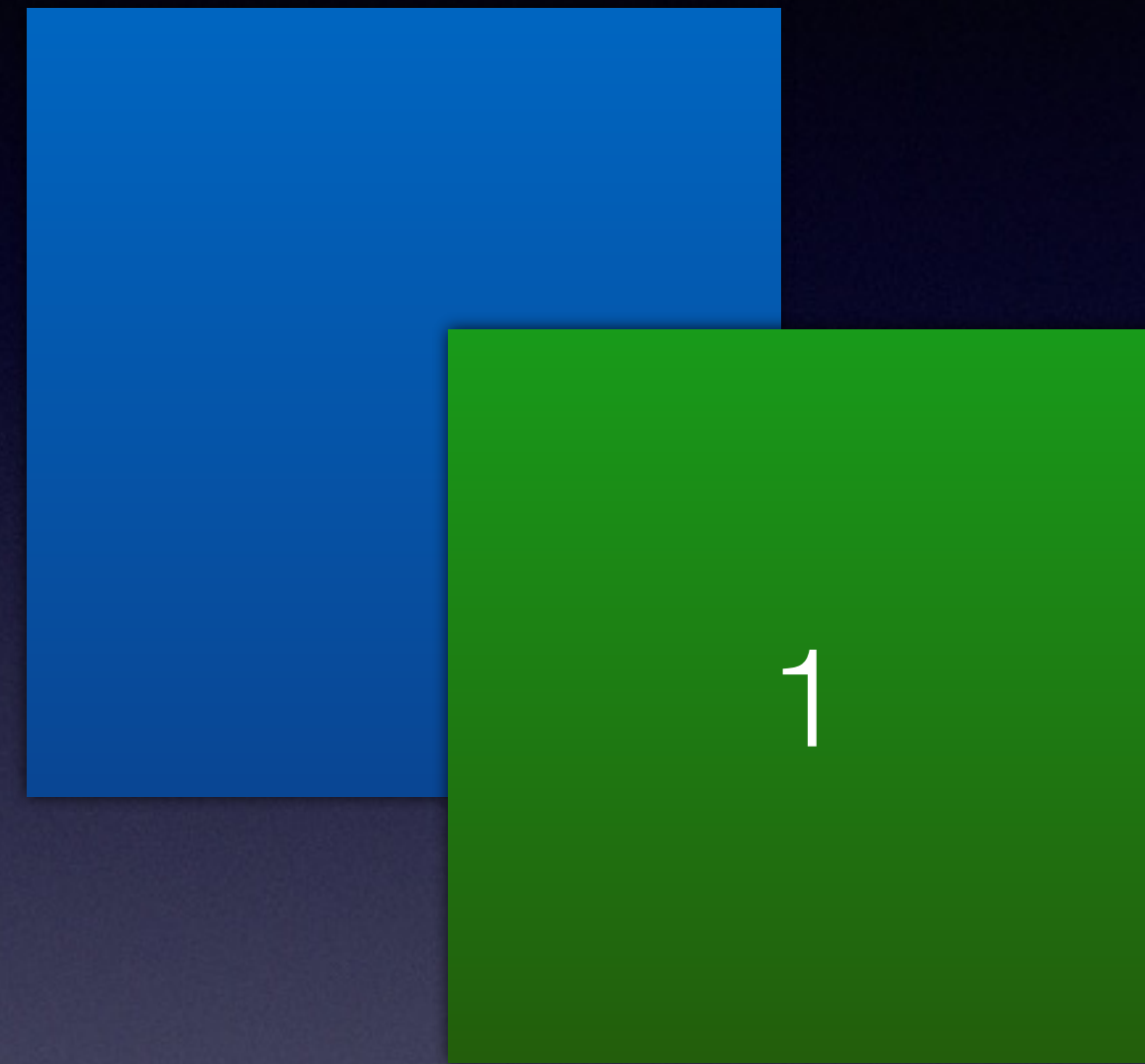


Use CSS to place the door `<div>` over the day ``.

You can use

```
position: absolute
```

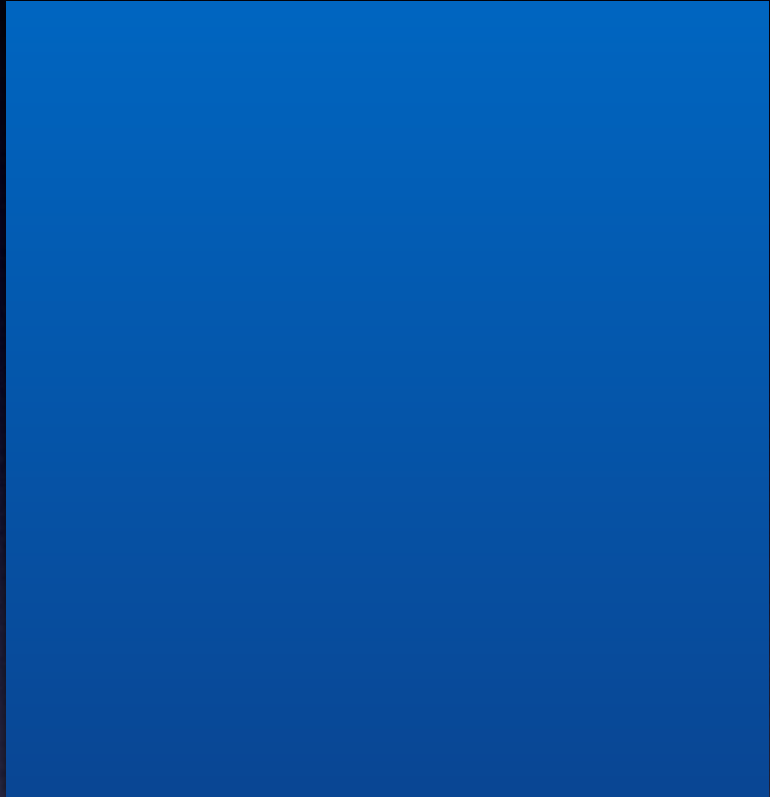
to achieve this.



Using absolute positioning means we can put it anywhere we like on the page over (or under) the other content.

1

You can use **top**, **bottom**, **left** and/or **right** as offsets to position the element when you use absolute positioning.



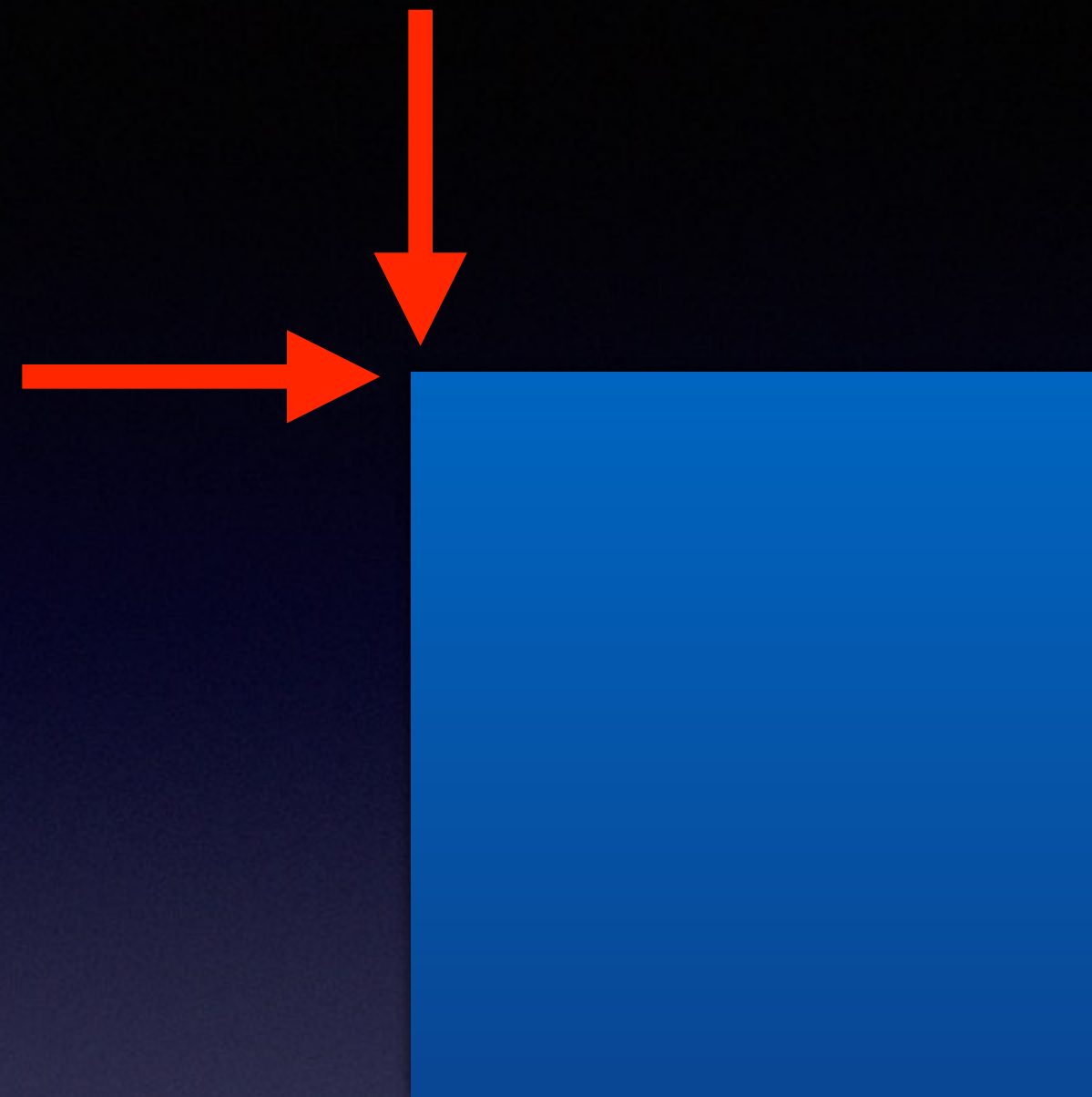
```
.door {  
position: absolute;  
top: 0px; left 0px;  
}
```


1

This may not work the way you want it to initially. In this case **top: 0px** and **left: 0px;** puts the **.door** <div> 0 pixels from the top and 0 pixels from the left ...

... but from the top and left of what?

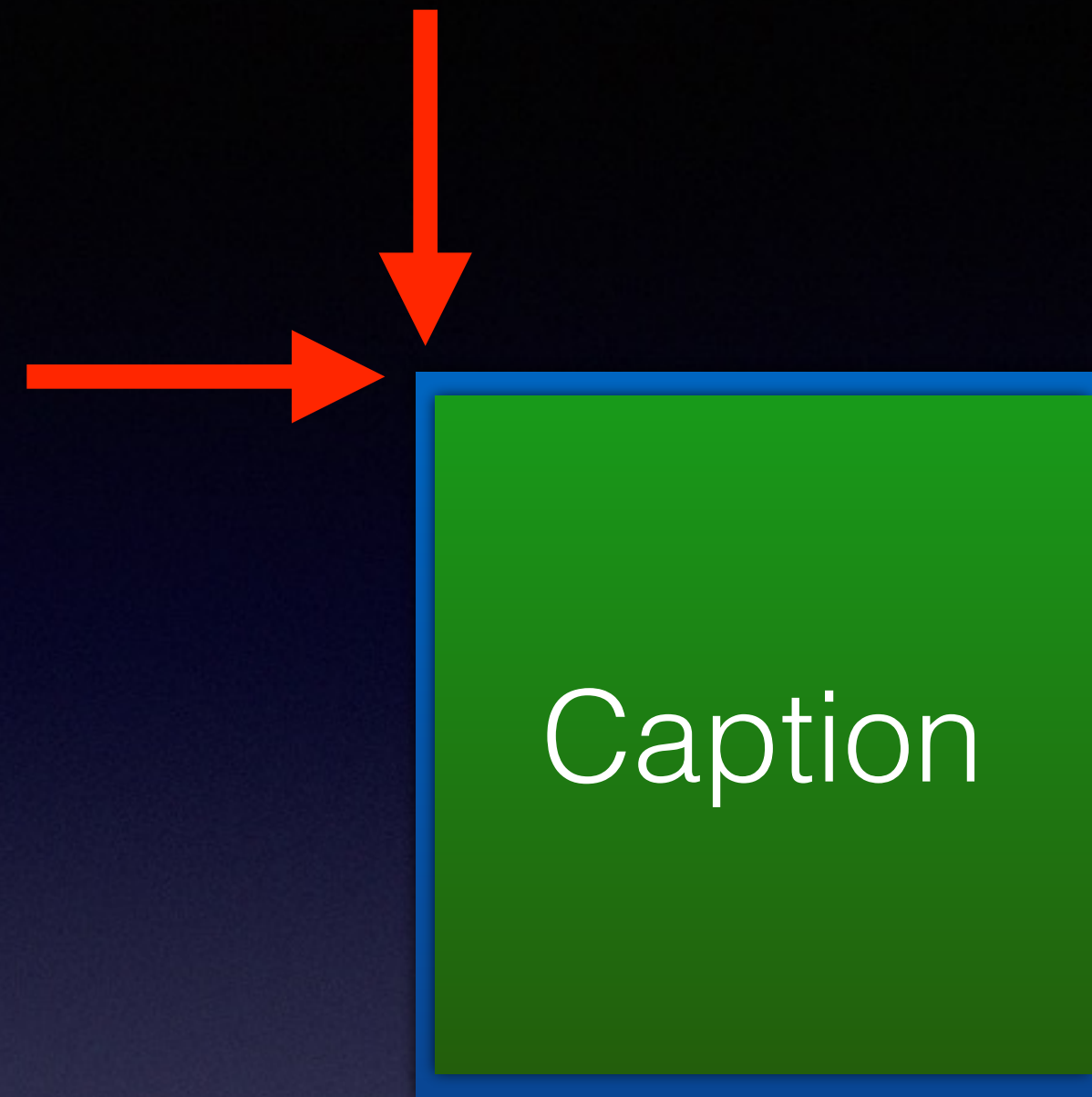
```
.door {  
  position: absolute;  
  top: 0px; left 0px;  
}
```

With no other option it will use the `<body>` tag (i.e. the entire document).

However, we want it to be calculated from the top lefthand corner of our ``.

```
.door {  
  position: absolute;  
  top: 0px; left 0px;  
}
```

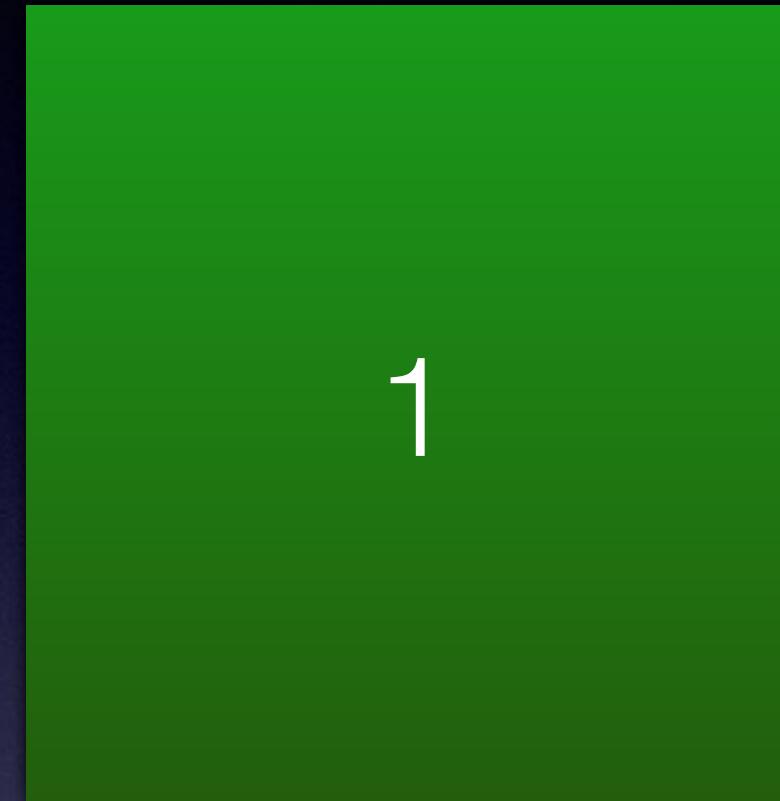



If we set the position of one of the elements that *contain* the **door** <div> to something other than the default (i.e. *static*) then it will use that element to calculate the offsets instead of the <body> tag.

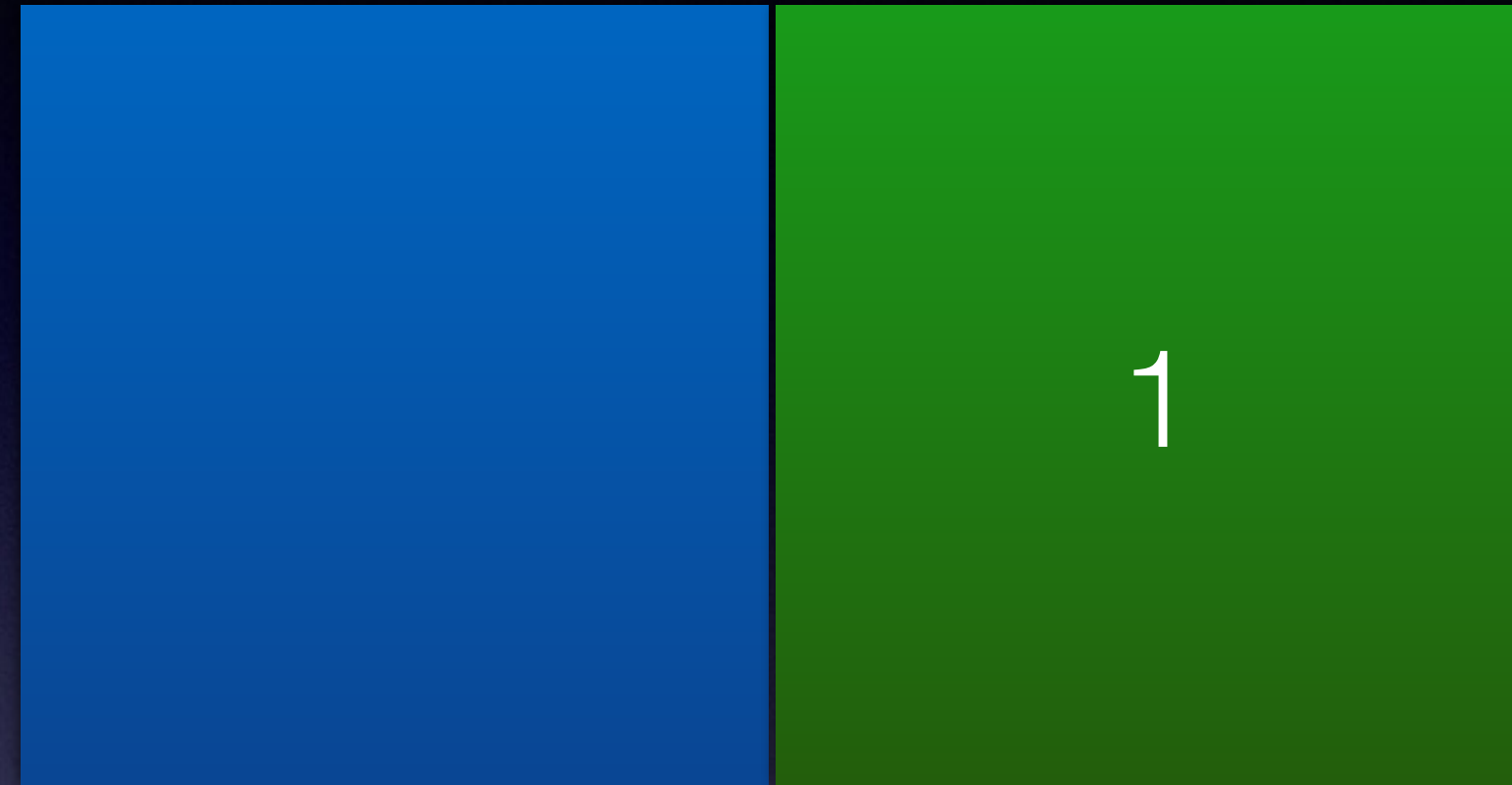
```
#calendar li {position: relative;}
```

```
.door {  
  position: absolute;  
  top: 0px; left 0px;  
}
```


Rollover



Once you position it correctly you can then effectively remove the **.door** <div> from the page by moving it off the



E.g.

`left: 75px;`



If you set overflow to hidden in the `` then any of its contents that don't appear in the space it takes on screen will not be visible.

```
overflow: hidden;
```




You only need to make it move when the mouse is over the container ``.

```
#calendar li:hover .door { left: 75px; }
```


This detects when you hover over the ``

... but we apply the style the **.door** `<div>` that is inside the **``**

This makes the **.door** `<div>` move.



```
#calendar li:hover .door { left: 75px; }
```


This change in the **left** property can then be animated with CSS3 transitions.

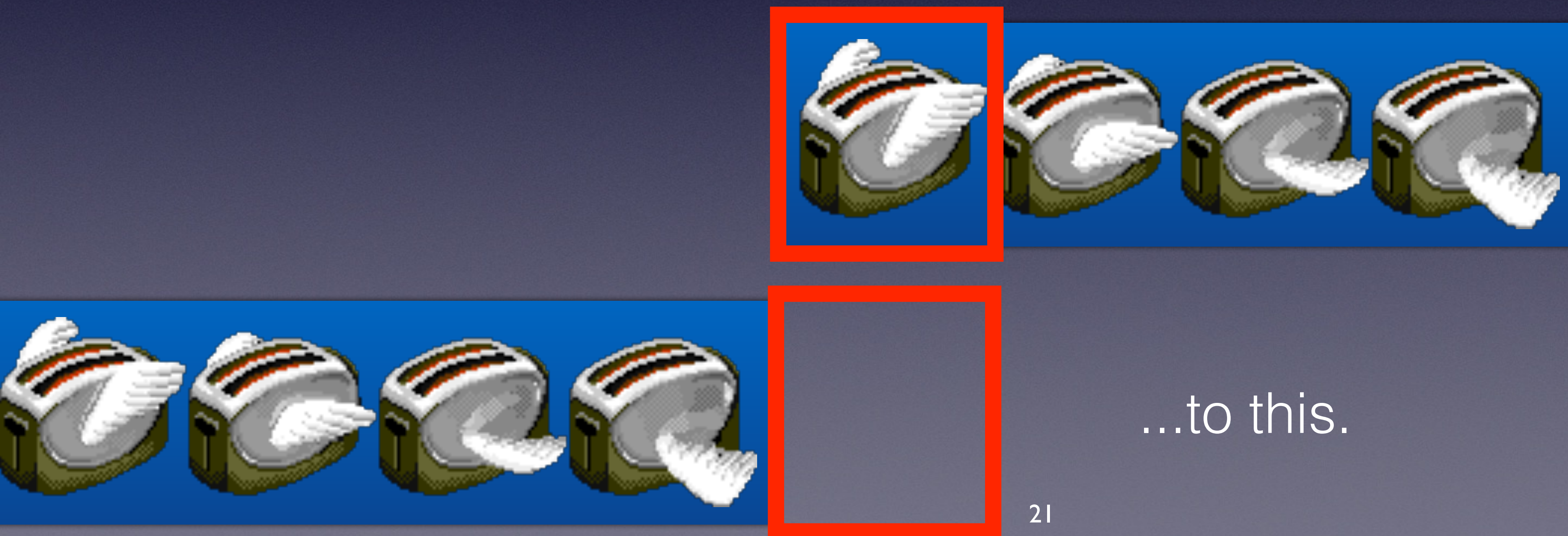
Animation Tips

For the scrolling landscape effect you need only animate a change in the position of a background image.

Note: Animating the position of an element (e.g. the opening of the door) can be more efficient if you animate the **translate transform** instead of the **top, left** properties.

For the animated sprite effects you can use keyframe animation. You need to change the position of a background image. The image will contain several frames of animation.

I.e. you need to go from this...



You are provided with three sprites.



sprite3.png 75 X 2376
32 Frames





toaster-sprite.png 75 X 300
4 Frames



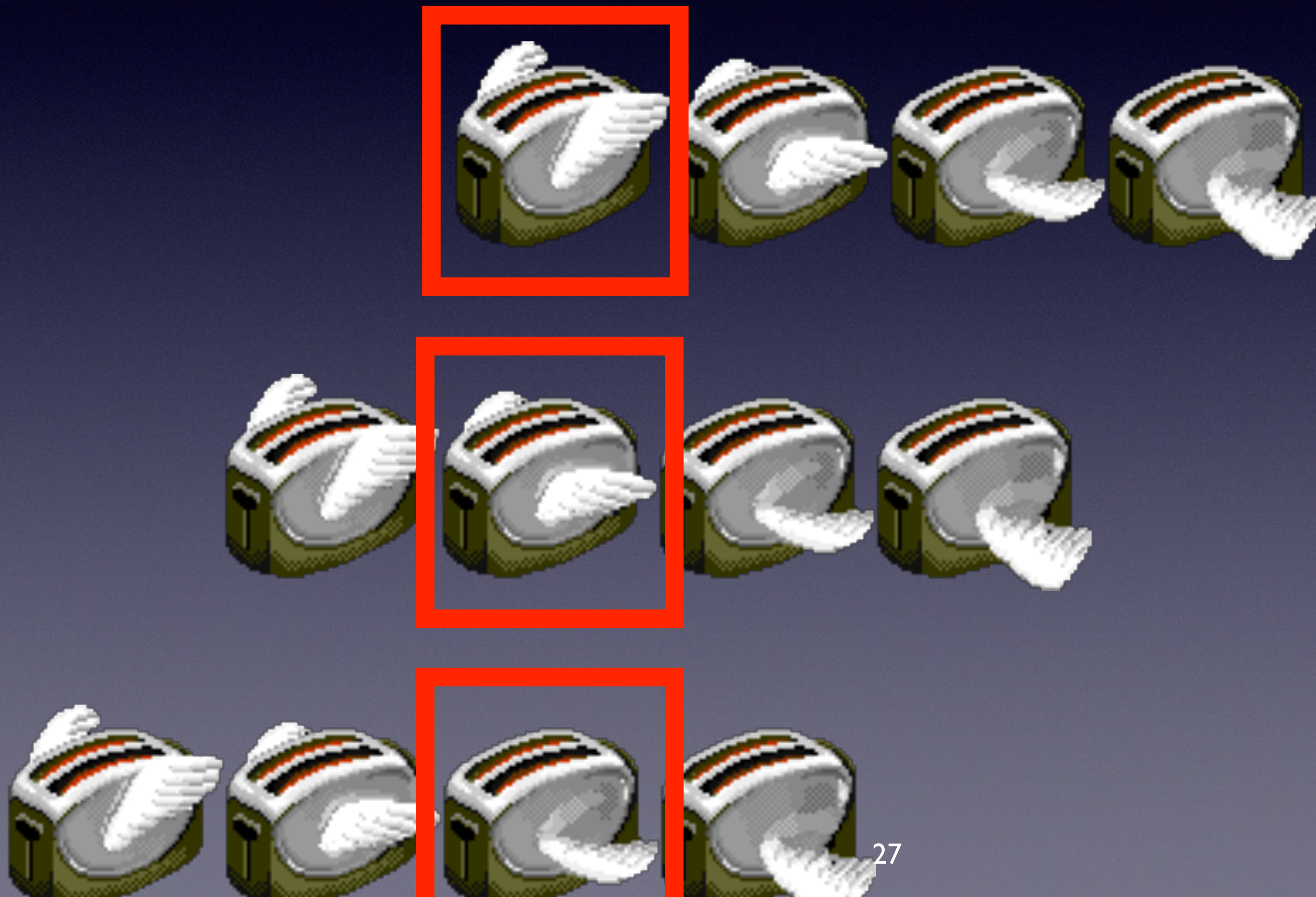
walk-left.png 75 X 450
6 Frames

Note: A *door* size of 75 X 75 will work best with these graphics.

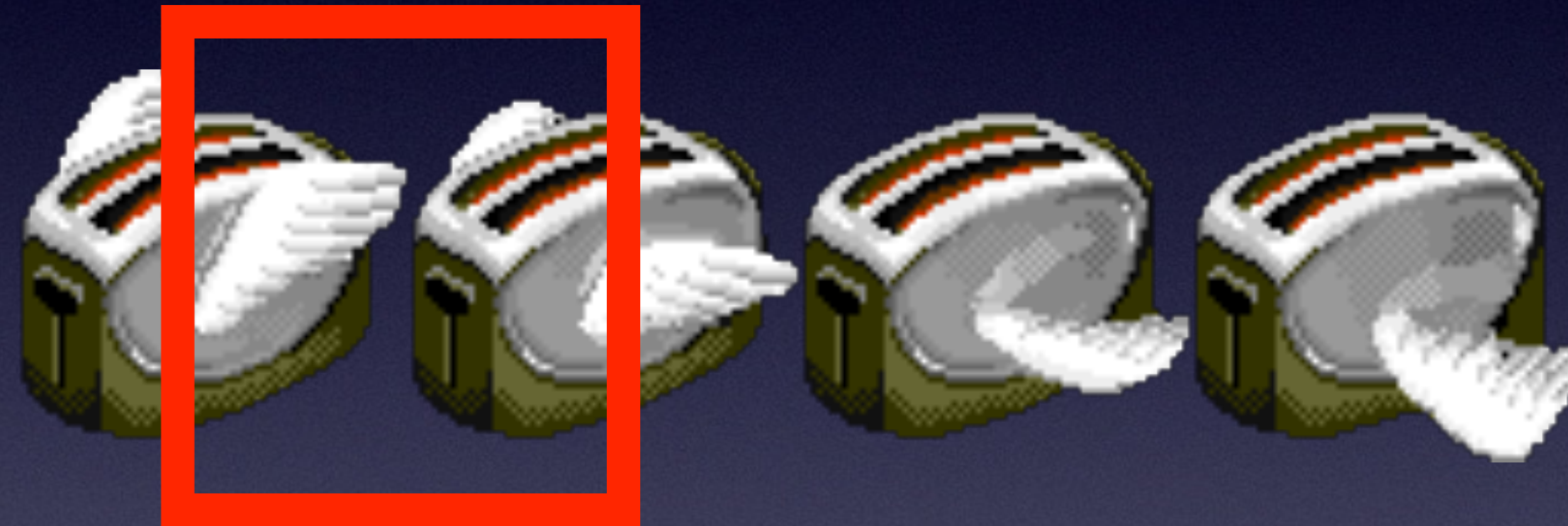


The sprite above is provided for you but you can also use find/create your own.

You can animate sprites by using the background position property.



To avoid the intermediate positions as shown below you can specify the number of steps the animation should take.



If the background position is moved through the entire length of the image, and the number of steps is the same as images in the sprite, then the background will only show complete sprite images.

```
animation: sprite-animation .4s steps(4)
```

Step 1



Step 2



Step 3



Step 4

