

Animation Frames

In this exercise we will use `requestAnimationFrame()` to animate a ball/pizza moving around the canvas.

Exercises

1. In `ball.html`, put a border around the canvas so that we can see its edges.

```
border: 1px solid rgb(100, 100, 100);
```

2. Stop the ball moving when it hits the bottom of the canvas. You might use an `if` statement for this purpose.

```
// If the ball hits the bottom of the canvas.  
if (...) {  
  // Stop the ball.  
  ...  
}
```

3. Clear the canvas at each step of the animation, so that only one copy of the ball is visible at a time. You can use the `clearRect` method for this.

```
ctx.clearRect(left, top, width, height);
```

4. Represent the ball as an object, rather than by a global `y` value and hard-coded `x` and `r` values.

```
var ball = {  
  ...  
}
```

5. Change the code so that the ball starts moving up the way once it reaches the bottom of the screen.

```
// If the ball hits the bottom of the canvas.  
if (...) {  
  // Change ball's direction.  
  ...  
}
```

6. Change the code so that the ball bounces from the bottom to the top of the screen, and back again, repeatedly.

```
// If the ball hits the bottom of the canvas.  
if (...) {  
  // Reverse the ball.  
  ...  
}
```

```
}  
// Otherwise, if the ball hits the top.  
else if (...) {  
    // Reverse the ball the other way.  
}
```

7. Give the ball a horizontal velocity, as well as a vertical one, and have it bounce off the left and right sides of the canvas also.
8. Change the code so that the pizza from lab 2 rather than the ball bounces

Advanced exercises

1. Give the ball/pizza a downwards acceleration, so that it gets faster as it falls, and slower as it ascends.

Notes

- See [here](#) for Mozilla's docs on `Window.requestAnimationFrame()`.
- Mozilla Developer Network: [WindowTimers.setInterval\(\)](#)
- Mozilla Developer Network: [WindowTimers.setTimeout\(\)](#)