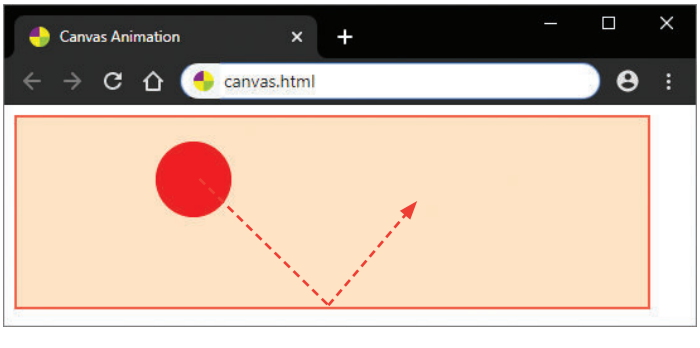
3. Now, insert statements to calculate new X and Y coordinates and call a function to paint the ball onto the canvas every 25 milliseconds

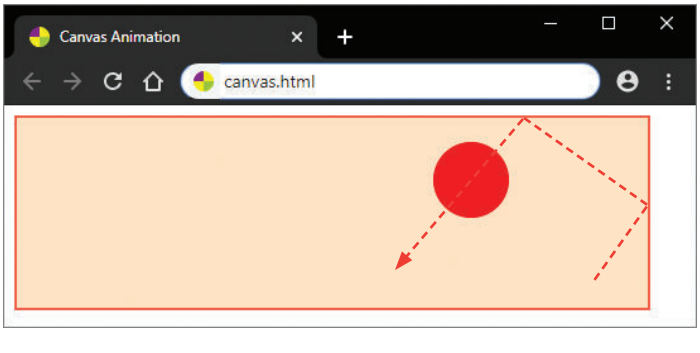
```
setInterval( function( ) {
  if ( ( x + dx > cw-30 ) || ( x + dx < 10 ) ) dx = -dx
  if ( ( y + dy > ch-30 ) || ( y + dy < 10 ) ) dy = -dy
  x += dx
  y += dy
  paint( context, cw, ch, x, y )  } , 25 )
```

4. Finally, add the function to actually paint the ball onto the canvas

```
function paint( context, cw, ch, x, y ) {
  context.clearRect( 0, 0, cw, ch )
  context.beginPath( )
  context.arc( x, y, 30, 0, ( Math.PI * 2 ), true )
  context.fill( )
}
```

5. Save the HTML document and script, then open the web page in your browser to see the animated ball magically bounce around the canvas





Notice how the polarity of the direction step gets reversed when the ball collides with a perimeter – so it doesn't bounce right off the canvas.

Create static background and borders as styles – so they need not be repeatedly painted.

Discover more about the Canvas2D API online at **html.spec.whatwg.org/multipage/canvas.html**