

Animaciones en JS (bases)

David Avellaneda
@davsket

animación

animación

ilusión de movimiento

animación

*secuencia de
cambios
en el tiempo*



primer intento

```
var direction = a < b ? 1 : -1,  
    delta = 2, c = a, temp; //10px per second  
while(c != b){  
    c = c + delta * direction;  
    if(c < b && a > b || c > b && a < b){  
        c = b;  
    }  
    elem.style[prop] = c + 'px';  
}
```

primer intento

- falla por falta de temporizadores
- mueve el elemento en deltas de unidades

segundo intento

```
var direction = a < b ? 1 : -1,  
    delta = 3, c = a, temp; //10px per second  
interval = setInterval(function(){  
    if(c != b){  
        c = c + delta * direction;  
        if(c < b && a > b || c > b && a < b){  
            c = b;  
        }  
        elem.style[prop] = c + 'px';  
    }  
    else{  
        clearInterval(interval);  
    }  
}, 10);
```

segundo intento

- si B queda muy lejos... la animación se vuelve leeeeeeeeeeeenta
- sigue restringida a la posición final

tercer intento
FÍSICA!

desplazamiento



A



B

desplazamiento



posición

(uniformemente acelerado)

$$X = X_0 + v \cdot t$$







$$X = X_0 + v \cdot \triangle t$$



$$X = X_0 + v \cdot \triangle t$$

$$X = A + \frac{B-A}{t} \cdot \triangle t$$



$$X = A + \frac{B-A}{t} \cdot \Delta t$$



$$X = A + \frac{B - A \cdot \Delta t}{t}$$

easing

$$X = A + B - A \cdot \frac{\triangle t}{t}$$

easing

es linear!

$$X = A + B - A \cdot \frac{\triangle t}{t}$$

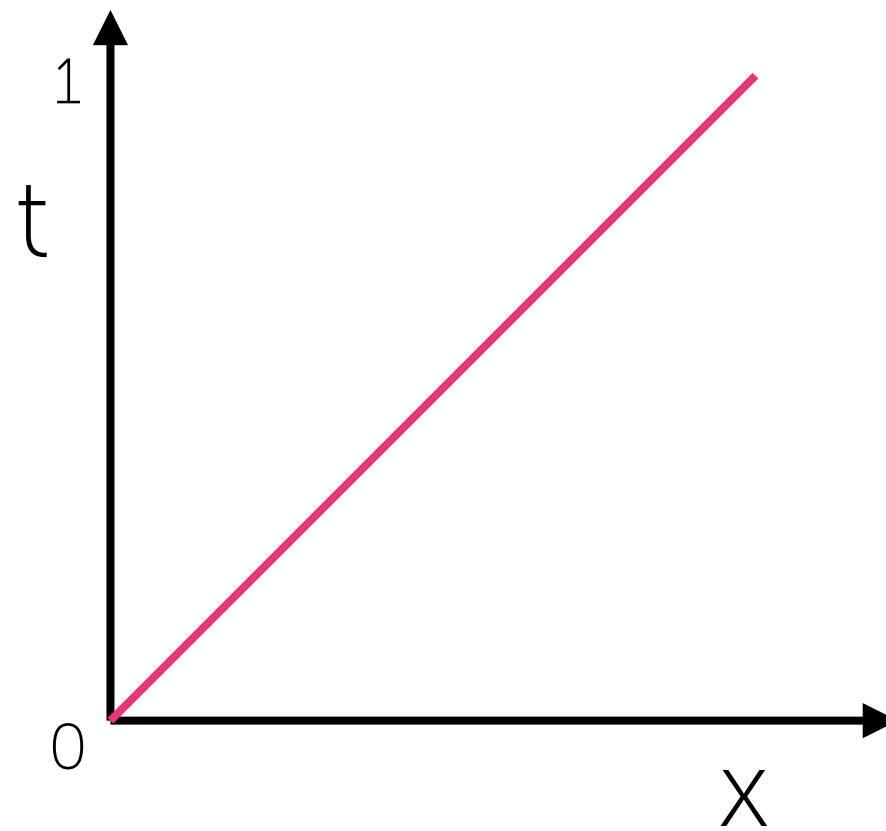
easing

es linear!

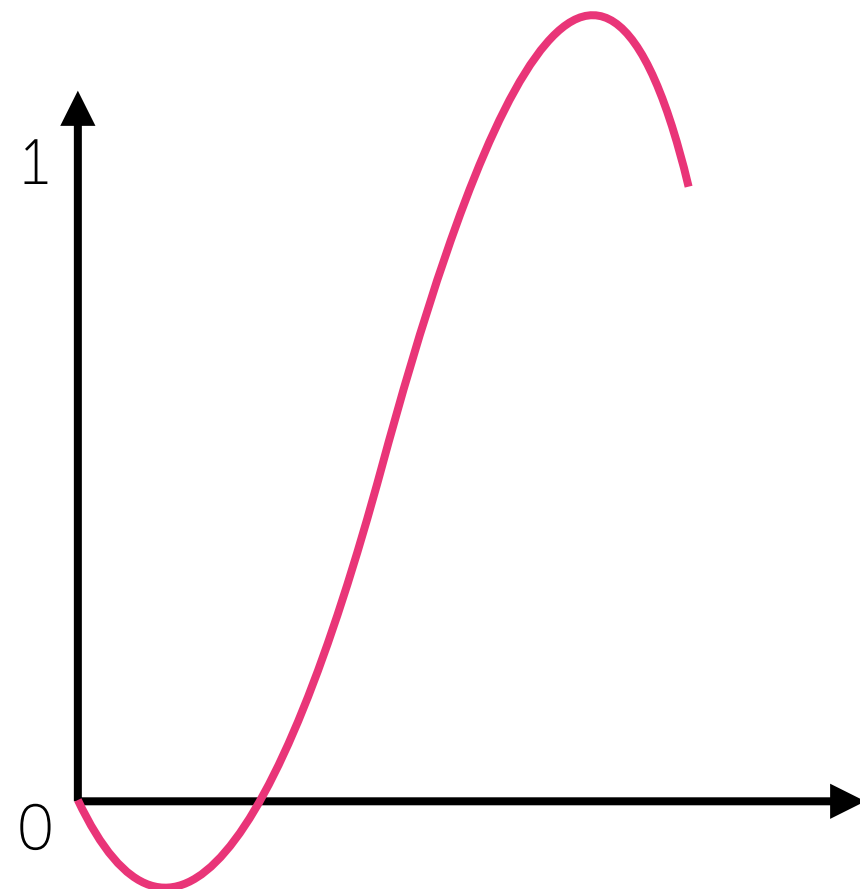
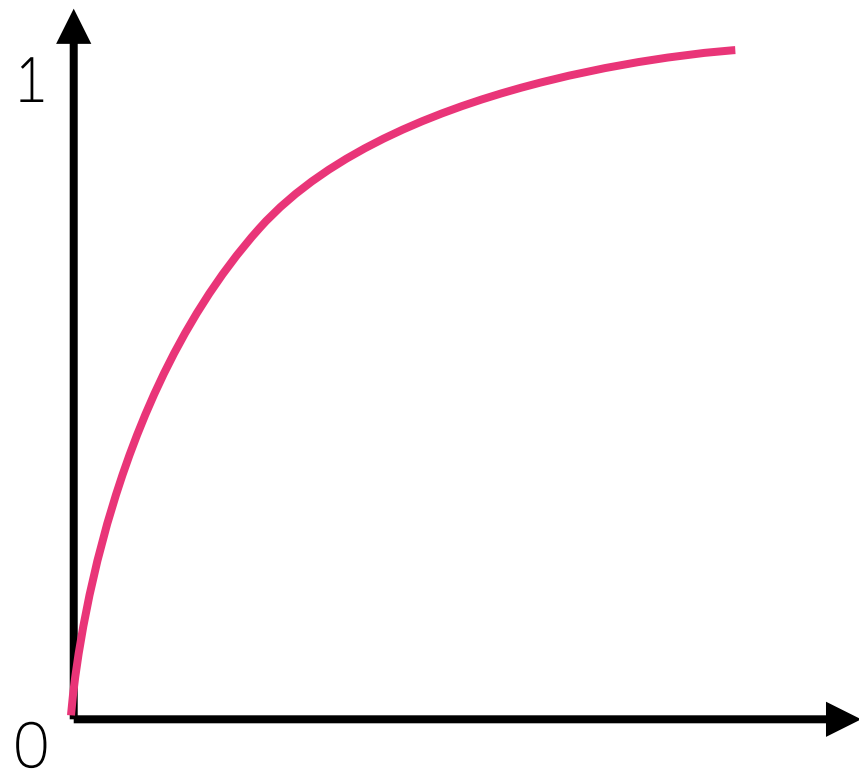
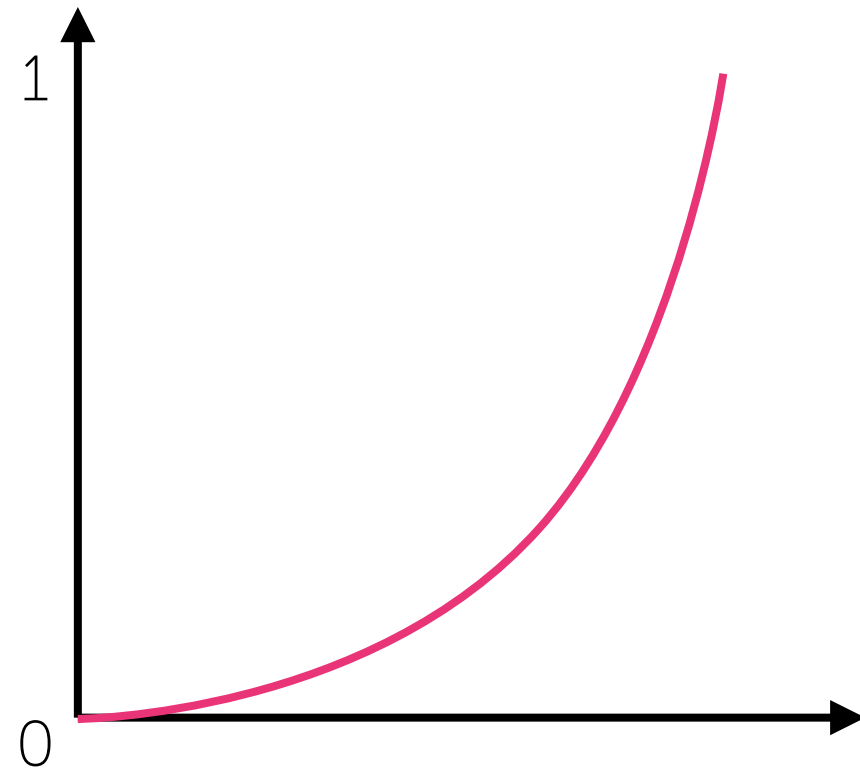
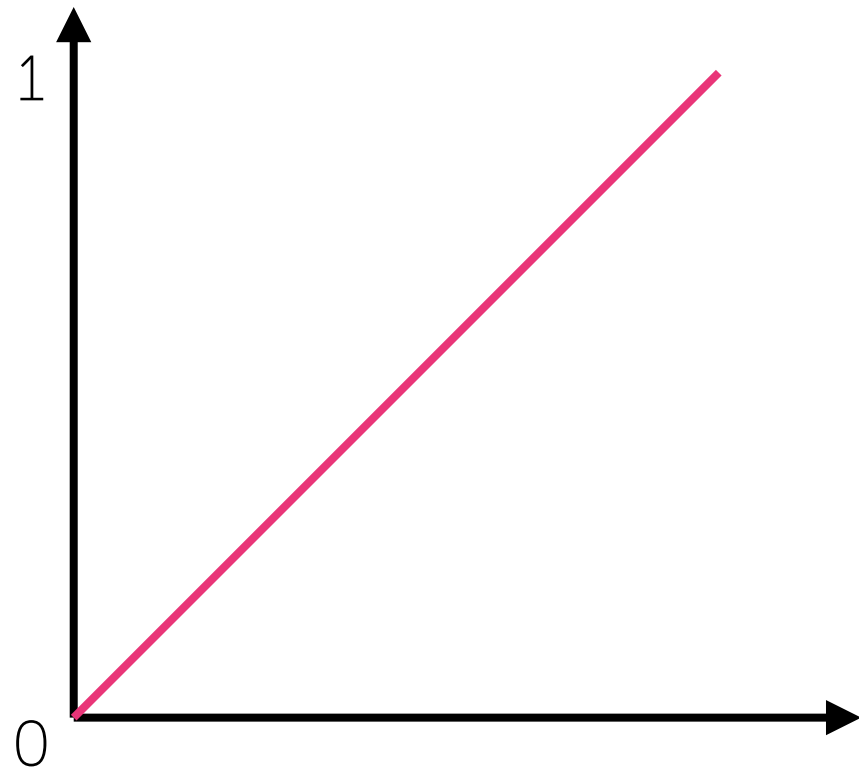
$$X = A + B - A \cdot \frac{\triangle t}{t}$$

easing

linear



easing



en javascript

```
function delta(start, end, duration, time){  
    return parseInt(start + (end-start) * time/duration);  
}
```


en javascript

easing

```
function delta(start, end, duration, time, easing)
{
    easing = easing || function(a){ return a };
    return parseInt(start + (end-start) *
        easing(time/duration));
}
```

en javascript

```
function animate(element, property, end, duration){
    var beginning = (new Date()).getTime(),
        start = parseInt(getStyle(element, property) || 0),
        interval = setInterval(morph, fps);

    function morph(){
        var now = (new Date()).getTime(),
            deltaAttr = end;
        if(duration > now - beginning){
            deltaAttr = delta(start, end, duration, now -
beginning);
        }else{
            clearInterval(interval);
        }
        element.style[property] = deltaAttr + 'px';
    }
}
```

subiendo el nivel

- animaciones de colores
- animaciones de propiedades complejas como:
 - bordes
 - outlines
 - box-shadow
 - ...

emile.js

- presentación: <http://script.aculo.us/downloads/emile.pdf>
- video: <http://fronteers.nl/congres/2009/sessions/roll-your-own-effects-framework>