

Animaciones

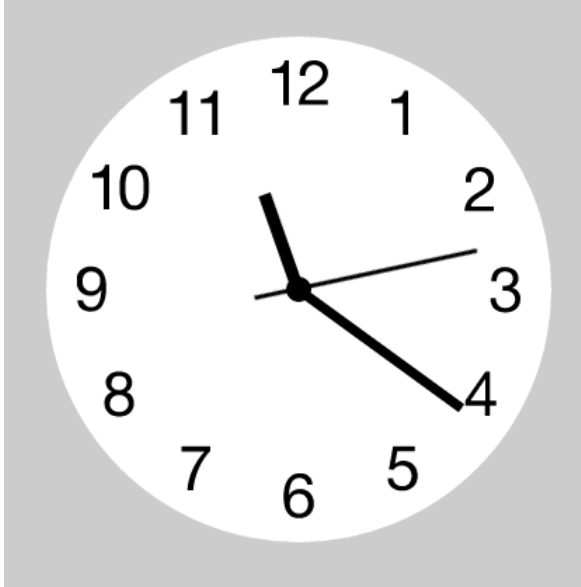
Animaciones

- **Procedurales**
- **Keyframe**
- **Basadas en física**

Animaciones Procedurales

La animación de cada **element** se realiza cambiando los valores de las propiedades mediante código

Reloj analógico



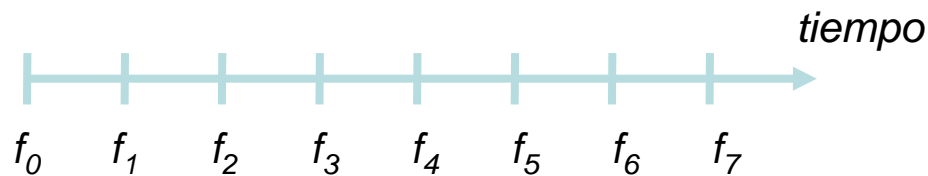
Rebote de pelota



Animaciones Keyframe

Secuencia de cuadros (frames) mostrados en secuencia para dar sensación de continuidad

Método clásico de animación (Dibujos animados)
Primera animación en 1872



Animaciones: Ejemplos

- Transición entre colores
- Movimiento de objetos
- Animación de un personaje 2D en un juego
- Animación complejas de fluido-dinámica

Animaciones: Ejemplos

Secuencia de imágenes:

<http://cssdeck.com/labs/google-doodle-in-css-follow-up>

Movimiento de objetos:

<http://neography.com/experiment/circles/solarsystem/>

Animación de personajes 2D (sprites):

<http://codepen.io/estelle/pen/LyItn>

Visualización compleja de fenómenos naturales:

<https://earth.nullschool.net/#current/wind/surface/level/orthographic=-54.91,-33.98,1489>

Animaciones vs Transiciones

Animaciones

- Pueden iniciar en cualquier momento
- No están atadas a un estado particular de un **element**
- Pueden ser “one shot” o “loopeables”
- Se puede definir la duración y dirección de reproducción
- Pueden modificar varias propiedades en varios pasos (steps)

Transiciones

- Se definen para modificar el **element** cuando pasa de un estado a otro
- Se debe tener cuidado al definir varias transiciones de un **element** a varios estados
- Pueden modificar varias propiedades en varios pasos (steps)

Ejemplo Transiciones

```
div {  
  width: 100px;  
  height: 100px;  
  background: red;  
  -webkit-transition: width 2s;  
  transition: width 2s;  
}
```

```
div:hover {  
  width: 300px;  
}
```

[Demo](#)

Easing

Se puede cambiar la forma en que se realiza la transición mediante las funciones de *easing*:

- ease
- ease-in
- ease-out
- ease-in-out
- linear
- cubic-Bezier
- o “Ninguna”

<http://easings.net/es>

<http://www.bufo.es/css3-transitions/>

Animaciones Keyframing CSS3

```
/* código de la animación */  
@keyframes example {  
    from {background-color: red;}  
    to {background-color: yellow;}  
}
```

```
/* El element al cual se aplica la animación */  
div {  
    width: 100px;  
    height: 100px;  
    background-color: red;  
    animation-name: example;  
    animation-duration: 4s;  
}
```

Keyframing usando CSS3 (2)

```
/* código de la animación */
```

```
@keyframes example {  
    0%    {background-color: red;}  
    25%   {background-color: yellow;}  
    50%   {background-color: blue;}  
    100%  {background-color: green;}  
}
```

```
/* El element al cual se aplica la animación */
```

```
div {  
    width: 100px;  
    height: 100px;  
    background-color: red;  
    animation-name: example;  
    animation-duration: 4s;  
}
```

Keyframing usando CSS3 (3)

```
/* código de la animación */
@keyframes example {
    0%    {background-color: red; left:0px; top:0px;}
    25%   {background-color: yellow; left:200px; top:0px;}
    50%   {background-color: blue; left:200px; top:200px;}
    75%   {background-color: green; left:0px; top:200px;}
    100%  {background-color: red; left:0px; top:0px;}
}
```

```
/* El element al cual se aplica la animación */
div {
    width: 100px;
    height: 100px;
    position: relative;
    background-color: red;
    animation-name: example;
    animation-duration: 4s;
}
```

Keyframing usando CSS3 (4)

Propiedades de animación:

- animation-direction
- animation-delay
- animation-timing-function
- animation-iteration-count
- ...

Shorthand:

```
div {  
    animation: example 5s linear 2s infinite alternate;  
}
```

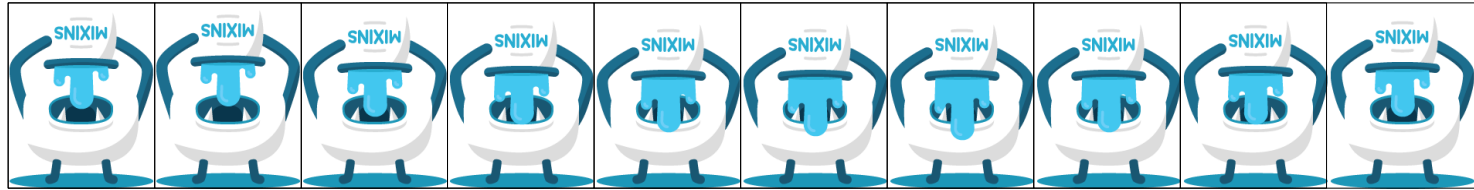
**Let's
work**

Animaciones Keyframe usando sprite sheet

Sprite Sheet



Sprite Sheet (2)



f_0

f_1

f_2

f_n

Sprite Sheet (3)



f_0

f_1

f_2

f_n

```
.monster {  
  width: 190px;  
  height: 240px;  
  background: url('monster-sprite.png') left center;  
  animation: play 0.8s  
}
```

```
@keyframes play {  
  100% { background-position: -1900px; }  
}
```

Sprite Sheet (4)



f_0

f_1

f_2

f_n

```
.monster {  
  width: 190px;  
  height: 240px;  
  background: url('monster-sprite.png') left center;  
  animation: play 0.8s ✘  
}
```

```
@keyframes play {  
  100% { background-position: -1900px; }  
}
```

Sprite Sheet (5)



f_0

f_1

f_2

f_n

```
.monster {  
  width: 190px;  
  height: 240px;  
  background: url('monster-sprite.png') left center;  
  animation: play 0.8s steps(10) infinite;  
}
```

```
@keyframes play {  
  100% { background-position: -1900px; }  
}
```

Sprite sheet: Resultado

Resultado final

<http://codepen.io/Guilh/pen/yldGp>

Post referencia:

<http://blog.teamtreehouse.com/css-sprite-sheet-animations-steps>

Sprite sheets de ejemplo para descargar:

<http://opengameart.org/content/2d-complete-characters>

DEMO