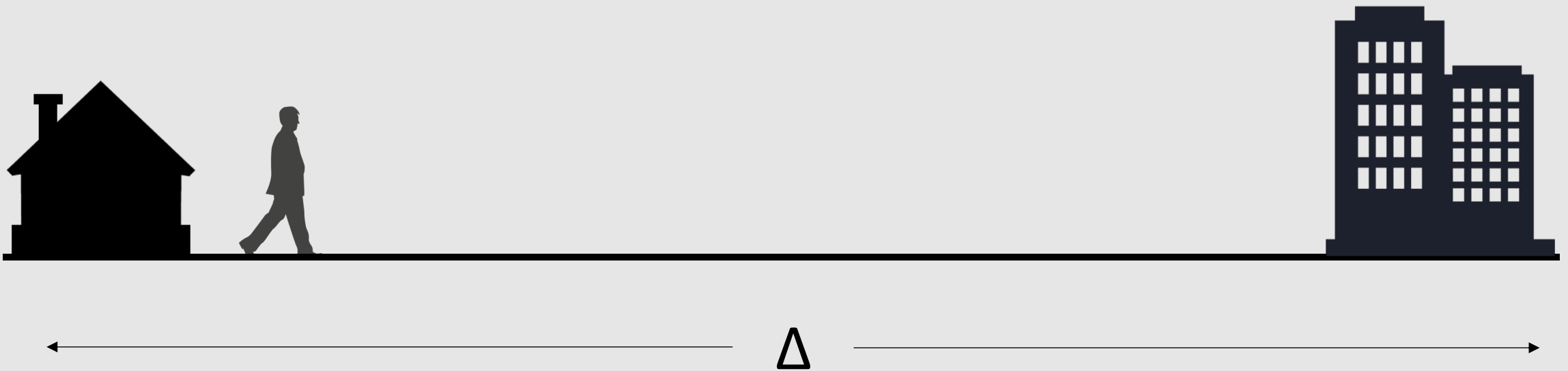


# Advanced Animations

- Animations: Recap
- Advanced Animations: A JQuery Approach
- Function Callbacks



### 3 Properties define an animation

- Change in property ( $\Delta$ )
- Time (t)
- Interpolation (I)

If two animation cycles share the same 3 properties, they are the same

# Transition Property

```
.css-transition {  
  width: 50%; }
```

```
.css-transition:hover {  
  width: 100%;  
  transition-property: width;  
  transition-duration: 2s;  
  transition-timing-function: linear;  
  transition-delay: 1s; }
```

Change in property ( $\Delta$ )	width += 50%
Time (t)	2 sec
Interpolation (I)	(50 / 2) (%/sec)

When we hover, the width increases linearly by 50% over 2 seconds with a 1 second delay.

When we unhover, the width decreases linearly by 50% over 2 seconds with a 1 second delay.

- ~~Animations: Recap~~

- Transitions in JQuery

- Function Callbacks

# Animation Events

---

Animations are queued by events in JQuery

The element moves down 200px in 5 seconds when clicked.

```
$(".cool-class").click( function() {  
    $(this).animate({top: '200px'}, 5000); } );
```

Moves down 200px for every click. Increments value of top rather than setting it to a constant.

```
$(".cool-class").click( function() {  
    $(this).animate({top: '+=200px'}, 5000); } );
```

# Animation Events

---

Can set a sequence of animations.

```
$(".mystery").click( function() {  
    $(this).animate({top: '200px'}, 5000);  
    $(this).animate({left: '200px'}, 5000);  
    $(this).animate({opacity: '0'}, 5000); } );
```

What does the above do?

Answer: A *mystery* object would move down 200px for 5s, then right 200px for 5s, and fade out in 5s, all sequentially.

# Animation Events

---

We can make animations occur simultaneously:

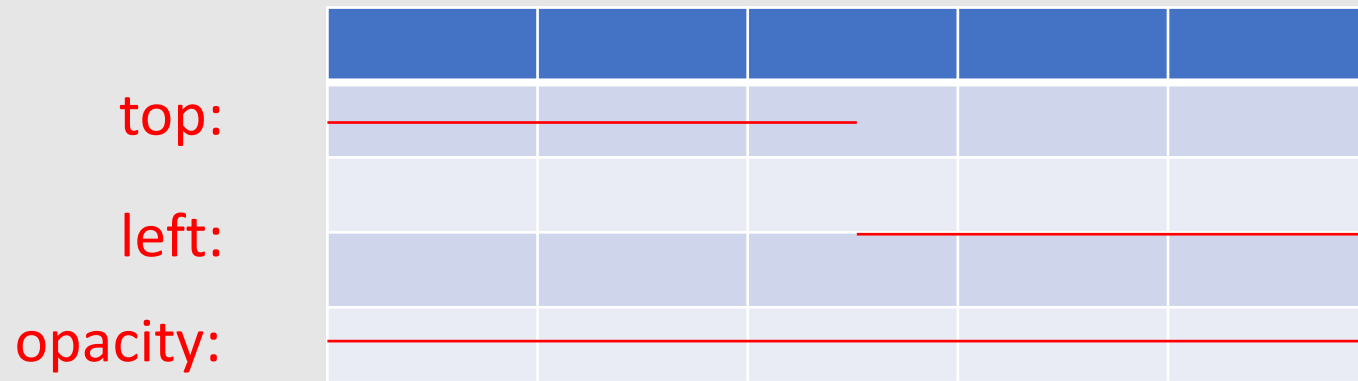
```
$(".mystery").click( function() {  
    $(this).animate({top: '200px',  
                      left: '200px',  
                      opacity: '0'}, 5000); } );
```



# Animation Events

top:				
left:				
opacity:				

jQuery is not very good at delaying animations for specific times.  
We usually default to CSS for these animations.



We can simulate concurrent animations in JQuery by using delays.

```
$(".mystery").click( function() {  
    $(this).css({'top': '200px',  
                'left': '200px',  
                'opacity': '0'});  
  
    $(this).css({'transition': 'top 2.5s linear 0s,  
                        left 2.5s linear 2.5s,  
                        opacity 1s linear 0s'}); } );
```

- ~~Animations: Recap~~
- ~~Transitions in JQuery~~
- **Function Callbacks**

# Function Callbacks

---

Animations can have callback functions that are called once the animation completes.

```
$(".element").click( function() {  
    $(this).animate( properties, time, callback ); } );
```

So yes, naturally, dumb things like these are allowed:

```
function infinite_recursion() {  
    $(".element").animate( {left: +=10px}, 1000 , infinite_recursion ); }
```

# Function Callbacks

---

Callbacks are useful for resetting properties of an animation.

```
$(".element").click( function() {  
    var left-val = $(this).css("left");  
    $(this).animate( {left: 100px}, 1s, function() {  
        $(this).animate( {left: left-val}, 1s ) } );  
    } );
```

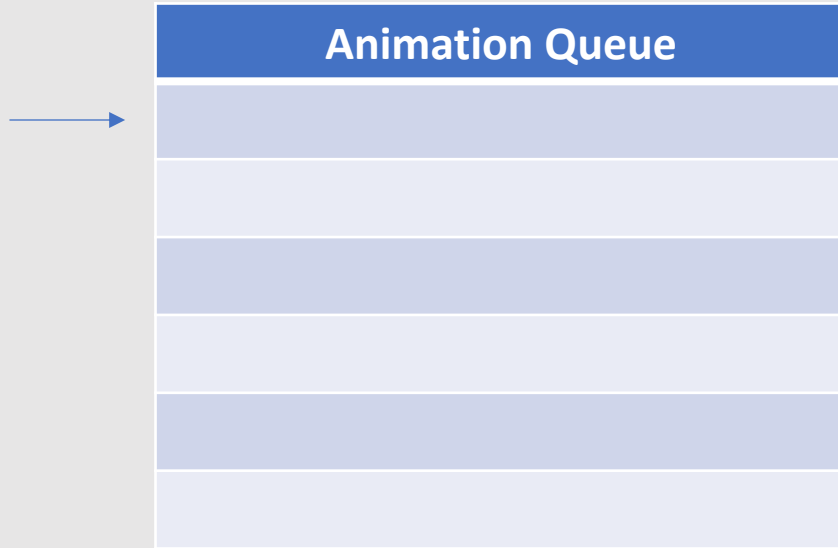
We do not need to know the initial value of the 'left' parameter to reset it.

# Function Callbacks

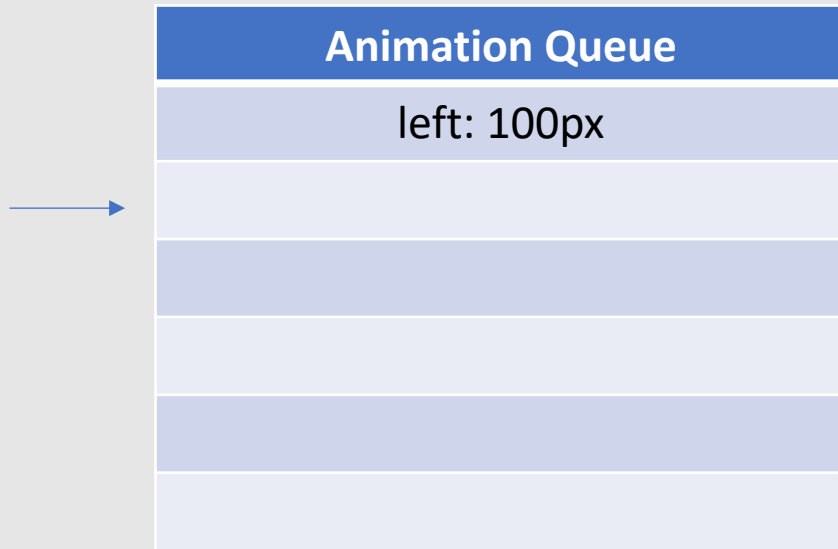
---

How will this execute?

```
function friendly_helper( class, reset ) {  
    $(class).animate( {left: reset}, 1s );  
}  
  
$(".element").click( function() {  
    var left-val = $(this).css("left");  
    $(this).animate( {left: 100px}, 1s, friendly_helper( this, left-val );  
    $(this).animate( {top: 100px}, 1s); } );
```



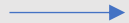
```
function friendly_helper( class, reset ) {  
    $(class).animate( {left: reset}, 1s );  
}  
  
$(".element").click( function() {  
    var left-val = $(this).css("left");  
    $(this).animate( {left: 100px}, 1s, friendly_helper( this, left-val );  
    $(this).animate( {top: 100px}, 1s); } );
```



```
function friendly_helper( class, reset ) {  
    $(class).animate( {left: reset}, 1s );  
}
```

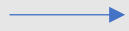
```
$(".element").click( function() {  
    var left-val = $(this).css("left");  
    $(this).animate( {left: 100px}, 1s, friendly_helper( this, left-val );  
    $(this).animate( {top: 100px}, 1s); } );
```





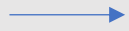
Animation Queue
left: 100px
top: 100px

```
function friendly_helper( class, reset ) {  
    $(class).animate( {left: reset}, 1s );  
}  
  
$(".element").click( function() {  
    var left-val = $(this).css("left");  
    $(this).animate( {left: 100px}, 1s, friendly_helper( this, left-val );  
    $(this).animate( {top: 100px}, 1s); } );
```



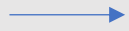
Animation Queue
<del>left: 100px</del>
top: 100px
left: 0px

```
function friendly_helper( class, reset ) {  
    $(class).animate( {left: reset}, 1s );  
}  
  
$( ".element" ).click( function() {  
    var left-val = $(this).css("left");  
    $(this).animate( {left: 100px}, 1s, friendly_helper( this, left-val );  
    $(this).animate( {top: 100px}, 1s); } );
```



Animation Queue
<del>left: 100px</del>
<del>top: 100px</del>
left: 0px

```
function friendly_helper( class, reset ) {  
    $(class).animate( {left: reset}, 1s );  
}  
  
$(".element").click( function() {  
    var left-val = $(this).css("left");  
    $(this).animate( {left: 100px}, 1s, friendly_helper( this, left-val );  
    $(this).animate( {top: 100px}, 1s); } );
```



Animation Queue
<del>left: 100px</del>
<del>top: 100px</del>
<del>left: 0px</del>

```
function friendly_helper( class, reset ) {  
    $(class).animate( {left: reset}, 1s );  
}  
  
$( ".element" ).click( function() {  
    var left-val = $(this).css("left");  
    $(this).animate( {left: 100px}, 1s, friendly_helper( this, left-val );  
    $(this).animate( {top: 100px}, 1s); } );
```

## When To Use JQuery Animate Tag?

- More flexible for programming-based animations.
- Can store variables and perform operations with JavaScript
- Easier to store Callback functions
  - Sometimes can lead to unexpected execution order if not used properly

## When To Use CSS Tag?

- With multiple independent animations.
  - If two animations share the same time and interpolation, they can be in the same animation tag.
  - Else, they must be in different animation tags independently.

# Homework Ideas

---

- ❑ Convert old CSS animations to JQuery animations.
- ❑ Use the JQuery queueing feature to queue animations and play them sequentially.
- ❑ Provide function callbacks resetting JQuery animations to their original state.

# Live Demo