Lesson 10

more jQuery

Housekeeping

Exit ticket question

Q: So, should we think in vanilla js and jquery js each time now? Should we try to do the homework both ways?

Exit ticket answer

A: In class for DOM manipulation we'll go back and forth between using the built-in DOM API, and using jQuery.

But primarily we'll use jQuery, so that you'll get a solid feeling for how to learn a library and use it to accomplish goals.

Don't worry too much about thinking both ways. Once you have the basics of JS the language itself, and of how the DOM tree is constructed, whether you access DOM elements with the browser's built-in DOM API functions, or through jQuery objects, is more of a detail. You can always look up the behavior of specific properties and functions when you need to.

Homework

Final projects

- For the final project, you'll be designing and building a single page web application

- This project will test your knowledge of JavaScript and ask you to apply everything you've learned in this course

- You could create anything from:

    - a blog users can comment on;

    - an app that allows users to search for social media posts; or even

    - an application that logs users' geolocations

- Work with your instructors to create project goals that are realistic given the scope and timing of the class

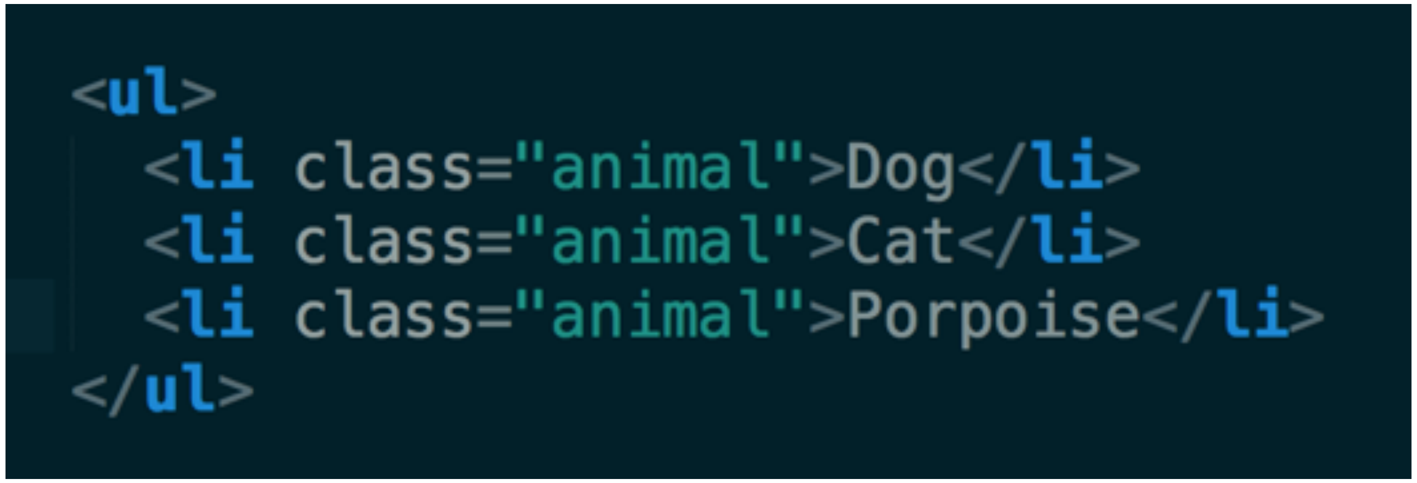
Final projects

- Link to final project info: http://bit.ly/jsdev2-final-project
- Some projects from last time:
  - tinder for pets: http://brianeng87.github.io/
  - web jukebox: http://mangonyc.github.io/
  - fitness tracker: https://cas-class-project.firebaseapp.com/
  - adventure time: http://ashleymcasey.github.io/
  - hot or not vinyl: http://christophercannon.github.io/
  - world weather: http://edswartz.github.io/

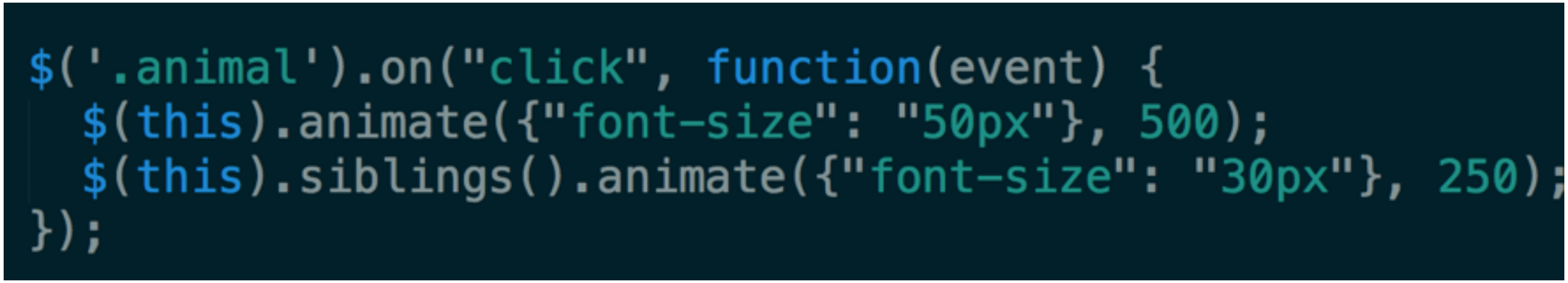
More fun with jQuery & events

- jQuery: **.siblings()** and **.animate()**

- Note: inside event handlers, the **this** keyword means the DOM element where the event was triggered.



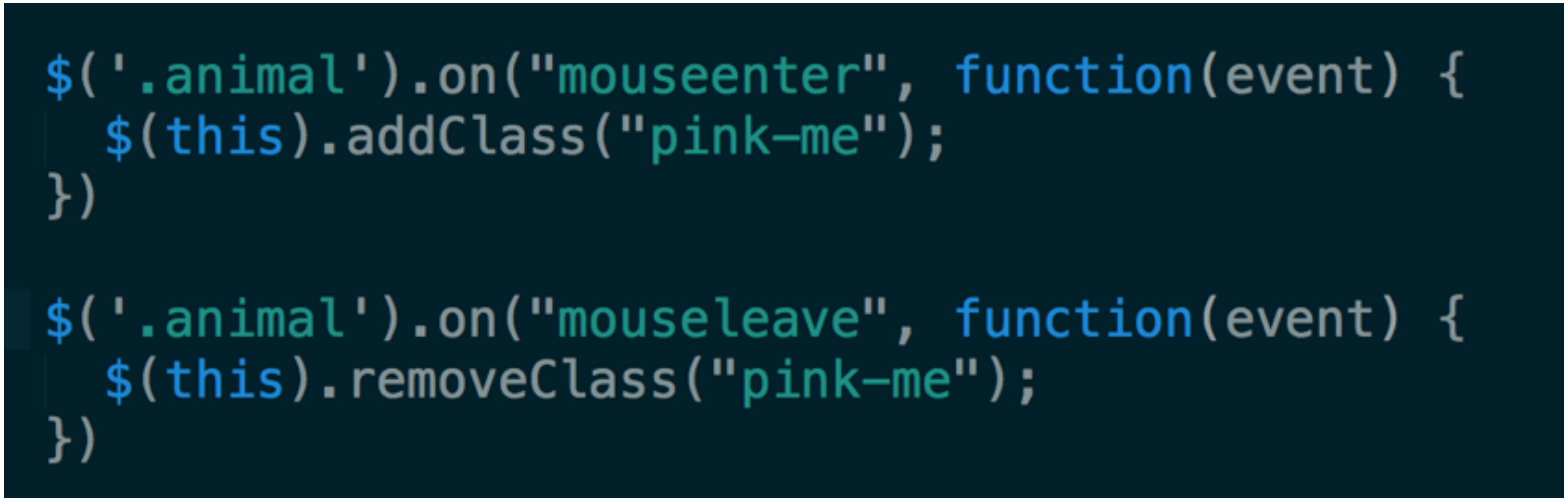
```
<ul>
  <li class="animal">Dog</li>
  <li class="animal">Cat</li>
  <li class="animal">Porpoise</li>
</ul>
```



```
$('.animal').on("click", function(event) {
  $(this).animate({"font-size": "50px"}, 500);
  $(this).siblings().animate({"font-size": "30px"}, 250);
});
```

More fun with jQuery & events

- jQuery: **.addClass()** and **.removeClass()**
- events: **mouseenter** and **mouseleave**

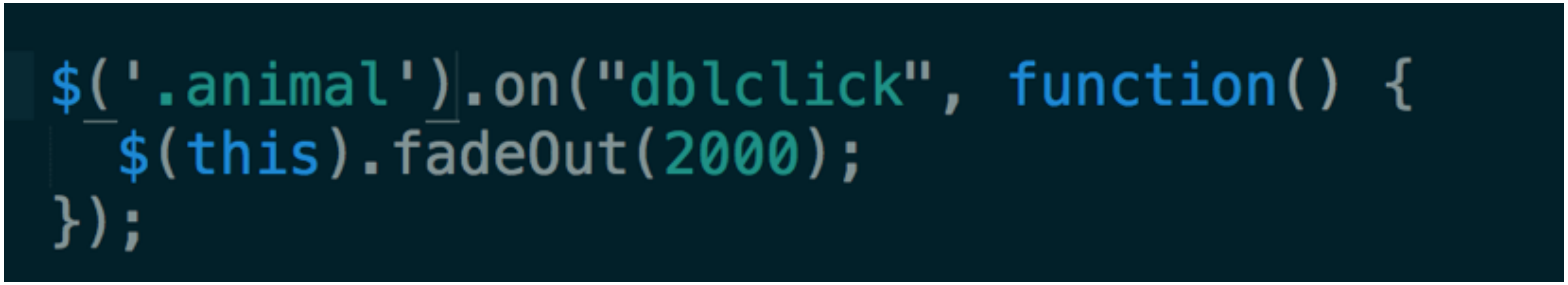


```
$('.animal').on("mouseenter", function(event) {
  $(this).addClass("pink-me");
})

$('.animal').on("mouseleave", function(event) {
  $(this).removeClass("pink-me");
})
```

More fun with jQuery & events

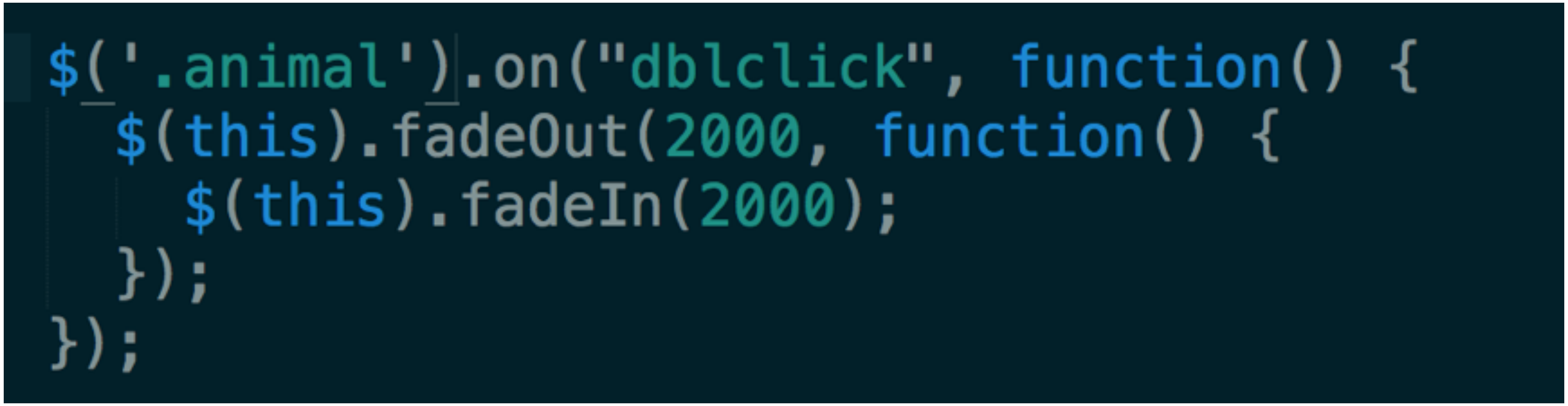
- jQuery: **.fadeOut()**
- events: **dblclick**



```
$('.animal').on("dblclick", function() {
  $(this).fadeOut(2000);
});
```

More fun with jQuery & events

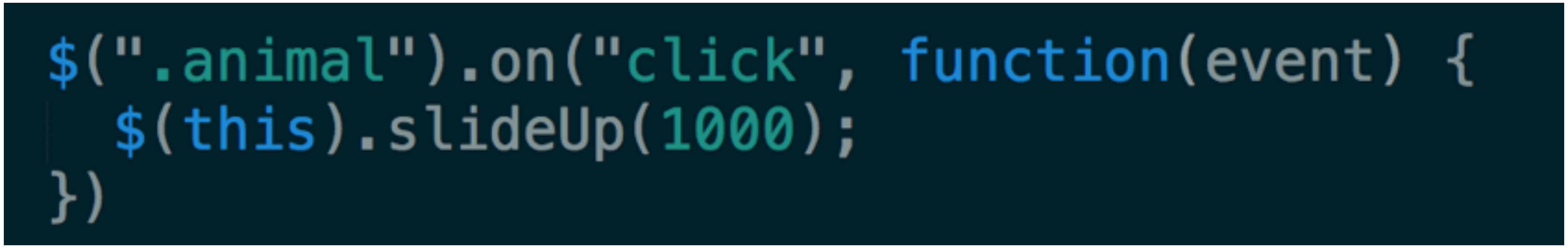
- jQuery: **.fadeIn()**
- If your head is exploding because of all the functions, don’t worry! We can put it (your head) back together later.



```
$('.animal').on("dblclick", function() {
  $(this).fadeOut(2000, function() {
    $(this).fadeIn(2000);
  });
});
```

More fun with jQuery & events

- jQuery: **.slideUp()**



```
$(".animal").on("click", function(event) {
  $(this).slideUp(1000);
})
```

Codealong

- Go to ~/GA-JS.
- Create a folder called “lesson10”
- cd into lesson10
- Create a folder called jquery-events
- create an html file and a js file
- subl .
- The link for jQuery: https://code.jquery.com/jquery-2.2.4.js

Practice on your own

- Create a folder called jquery-events-practice
- Make an html file and a js file
- jQuery link: https://code.jquery.com/jquery-2.2.4.js
- Make a list (**ul**) with three things (**li**s), all with the same CSS class.
- Play around with adding click listeners (or other event listeners) to the items to do all the effects we just talked about.
- **Bonus:** Look at the jQuery docs, https://api.jquery.com/category/effects/, and use one of the other effects you find there,
  like **.fadeTo()** for instance.

Moar events

- Many different events are available, divided into different categories: mouse, keyboard, form, document/window (and others).

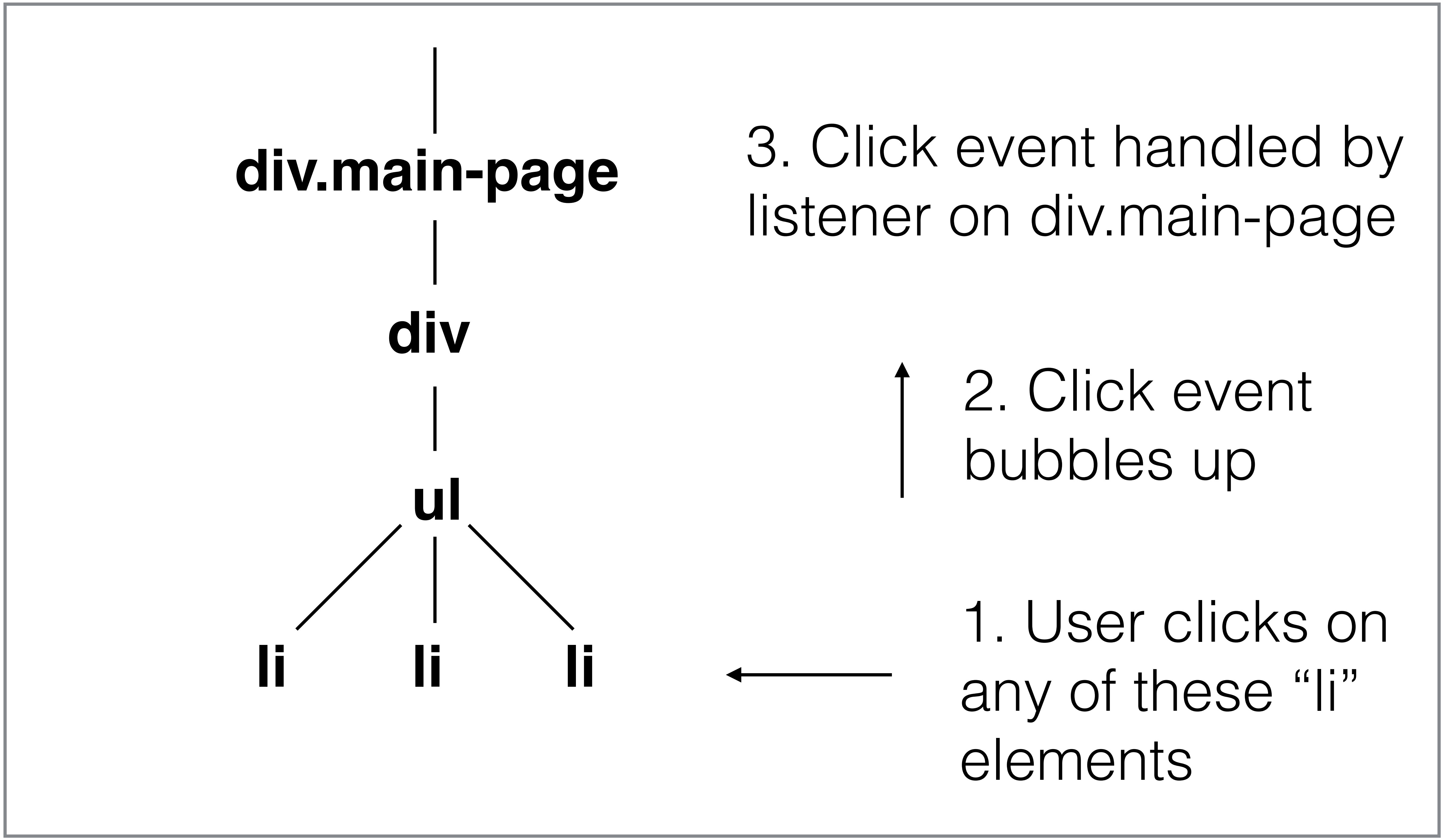
- A few common ones:

Mouse	click, dblclick, mouseenter, mouseleave
Keyboard	keypress, keydown, keyup
Form	submit, change, focus, blur
Document/ Window	load, resize, scroll, unload

Event bubbling

- Unless stopped, events **bubble** up the DOM tree and then hit any listeners they find.

Event Bubbling



3. Click event handled by listener on div.main-page

2. Click event bubbles up

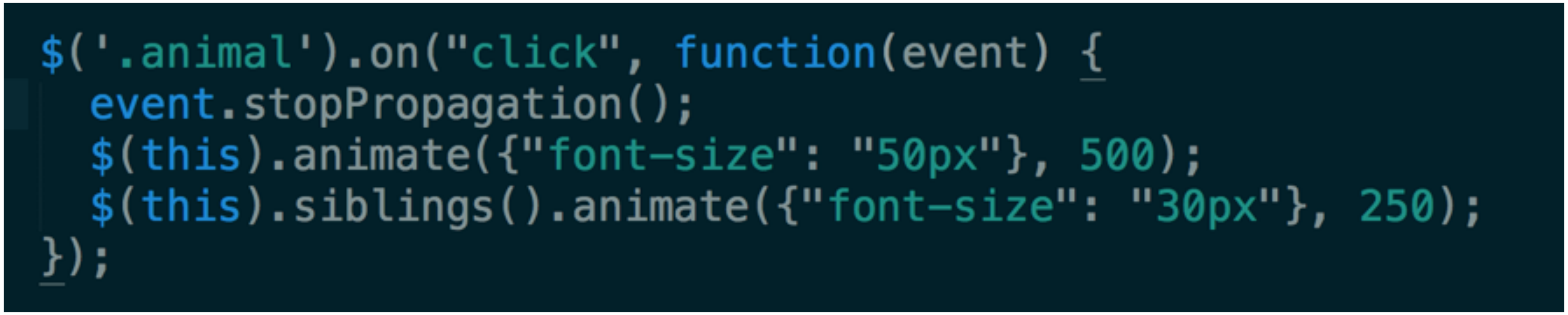
1. User clicks on any of these “li” elements

Event bubbling

- What if you already handled your event down below, and you don’t want any click listeners up the tree handling your event again?

- How do you stop it?

- Answer: **event.stopPropagation()**



```
$('.animal').on("click", function(event) {
  event.stopPropagation();
  $(this).animate({"font-size": "50px"}, 500);
  $(this).siblings().animate({"font-size": "30px"}, 250);
});
```

Event delegation

- We can take advantage of event bubbling to set listeners high up which are the first listeners to catch events bubbling up from below.

- This is called **event delegation**

- Two advantages:

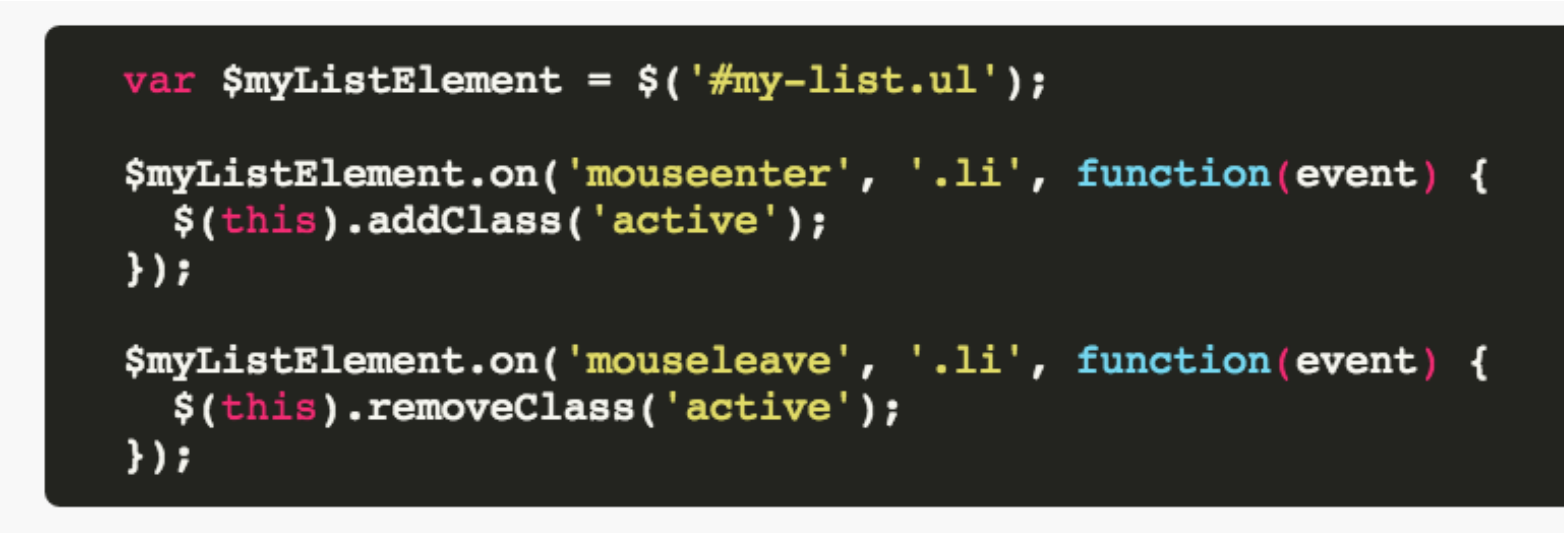
    - The browser doesn’t have to keep track of listening for very many events, and

    - We can catch events from elements that didn’t even exist when we set the event listener on their parent.

Event Delegation

- How do you get access to the element where the event actually occurred?

  - In jQuery, it’s all taken care of automagically by the **.on** method — you just add an extra argument:



```
var $myListElement = $('#my-list.ul');

$myListElement.on('mouseenter', '.li', function(event) {
  $(this).addClass('active');
});

$myListElement.on('mouseleave', '.li', function(event) {
  $(this).removeClass('active');
});
```

Codealong

- Make a new folder in your lesson10 folder: “delegation”
- Create a “delegation.html” and “delegation.js” file
- Open the folder in Sublime.

jQuery lab: event delegation

- (github being flaky, I’ll slack it)
- assigned as homework, due Monday, June 6