Web Development

COMP 431 / COMP 531

Lecture 7: Design

Instructor: Mack Joyner

Department of Computer Science, Rice University

mjoyner@rice.edu

http://www.clear.rice.edu/comp431

Recap

- HTML and HTML5, Storage, Canvas
- JavaScript and Scope
- Forms
- CSS
  https://www.clear.rice.edu/comp431/pdfs/lec_css.pdf
- Events

Homework Assignment 3
(JS Game)
Due Thursday 9/28

Q: Can we use external JavaScript libraries?
A: Only those explicitly discussed in class.

Design Decisions

- Many times there are multiple ways to solve a problem
  - First solution that comes to mind might not be a good solution
- Key Question: What's the best (a good) design for the solution?
  - Design reviews are helpful
- Code smells bad: Not a good design choice
  - Get exposed when code size grows
- Design is important (web dev apps)

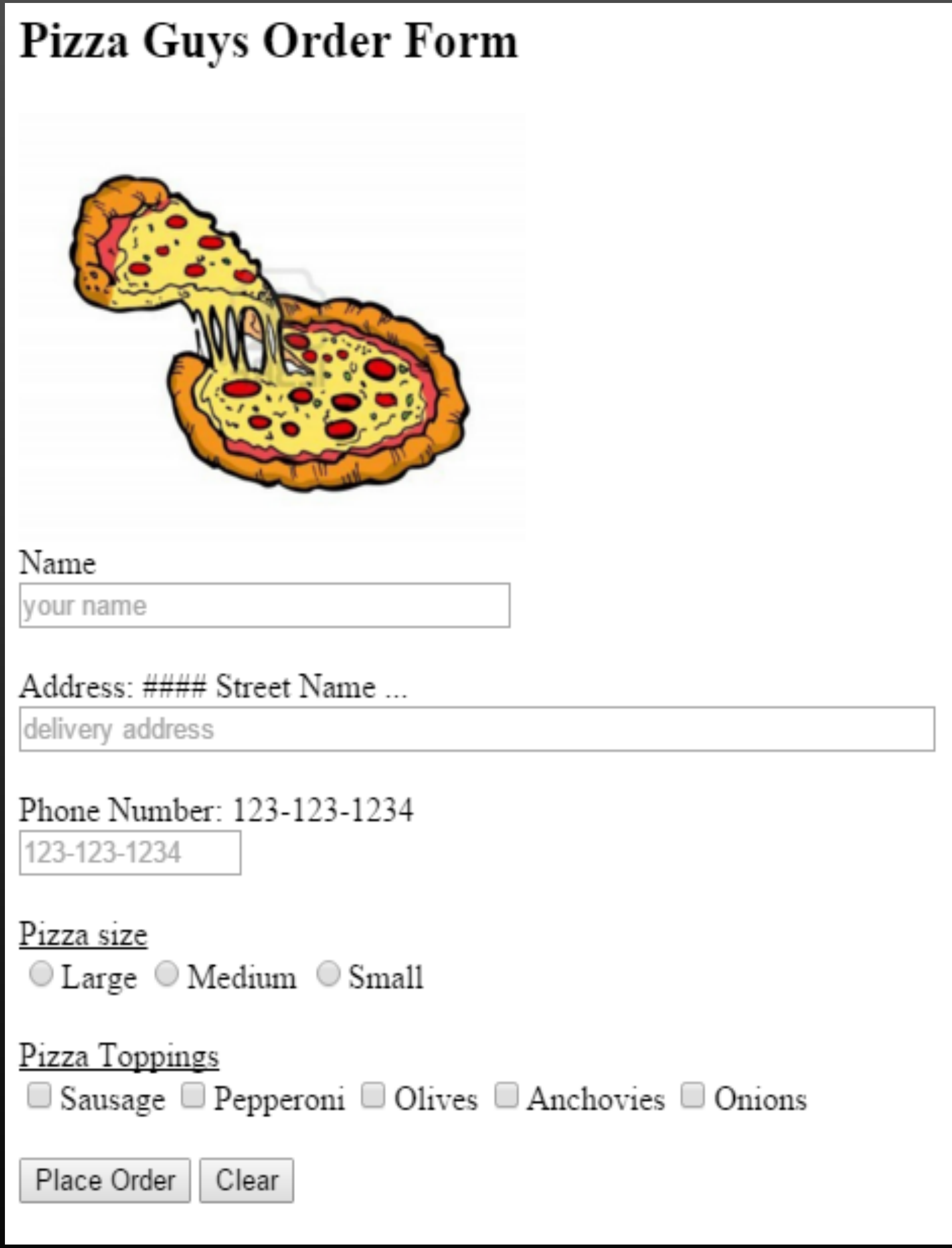
In-Class Exercise 2: Pizza Order Form

Make a validated Pizza Order Form

1. Text fields for customer name and street address
2. Phone number (placeholder)
3. User selects size of pizza (radio button group)
4. User selects toppings (check boxes)
5. Button to Place Order (does validation)
6. Button to Clear form (clears the form)
7. Validate required name, address, valid phone number, size of pizza has been selected. Can be all HTML5!

Could use JavaScript to ensure pizza size is checked.

Or, just set pizza size (large) by default



Pizza Guys Order Form

Name

your name

Address: #### Street Name ...

delivery address

Phone Number: 123-123-1234

123-123-1234

Pizza size

○ Large ○ Medium ○ Small

Pizza Toppings

☐ Sausage ☐ Pepperoni ☐ Olives ☐ Anchovies ☐ Onions

Place Order | Clear

In-Class Exercise 6: Skyline Game

Improve the skyline game

1. Not all lights are on in each building
2. Mouse click on a building makes that building grow taller
3. Add the Sun and have it move across the sky

   Hint: Erase and redraw the image
4. Add a car that drives along the ground

https://www.clear.rice.edu/comp431/sample/skyline.html
https://www.clear.rice.edu/comp431/sample/skyline.js

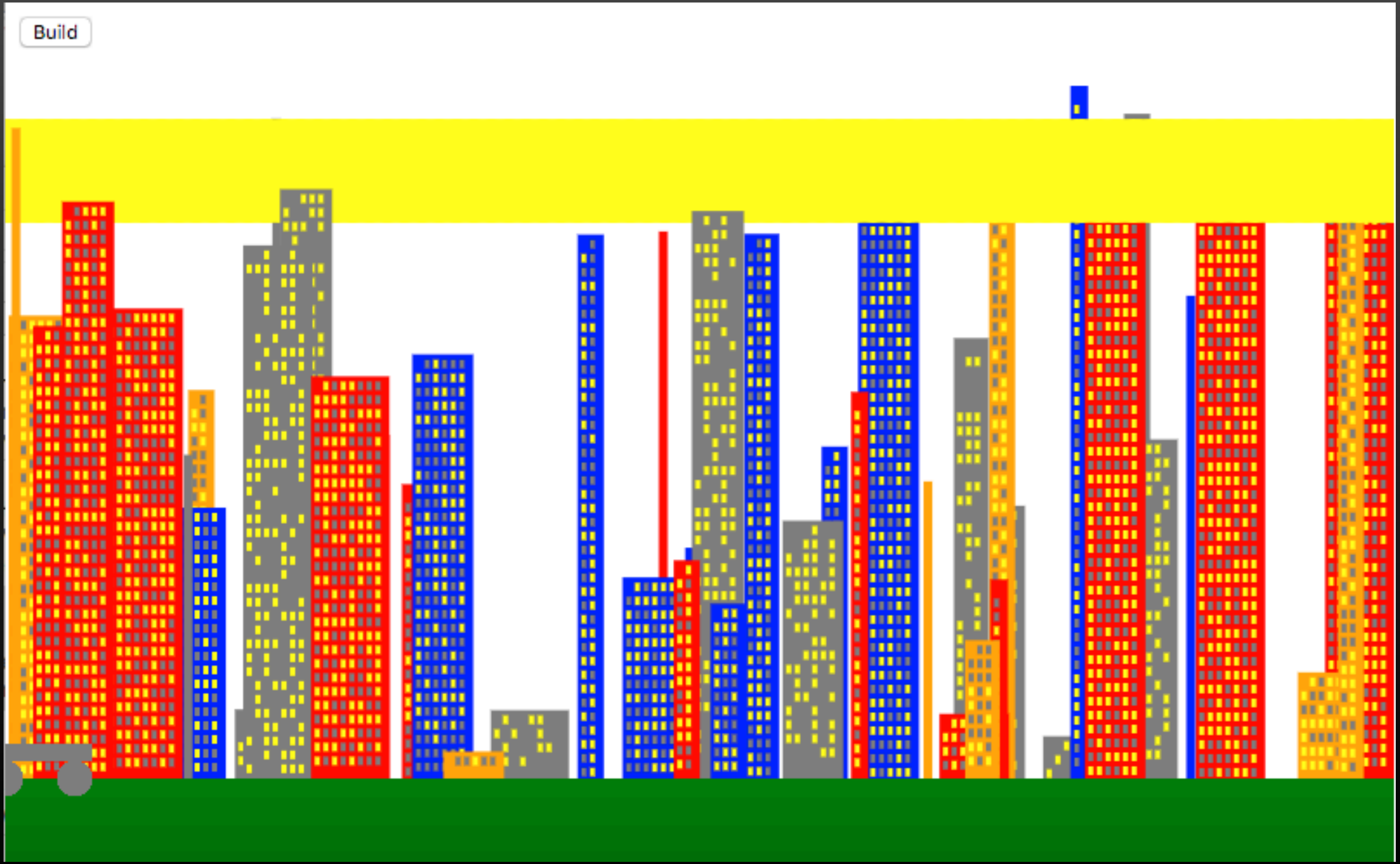
Design Decisions: Skyline Game

1. How should the sun move across the sky?
    - How do we keep it from hitting buildings?
    - How do we prevent a sun streak?
    - Should the sun move in front or behind the buildings?

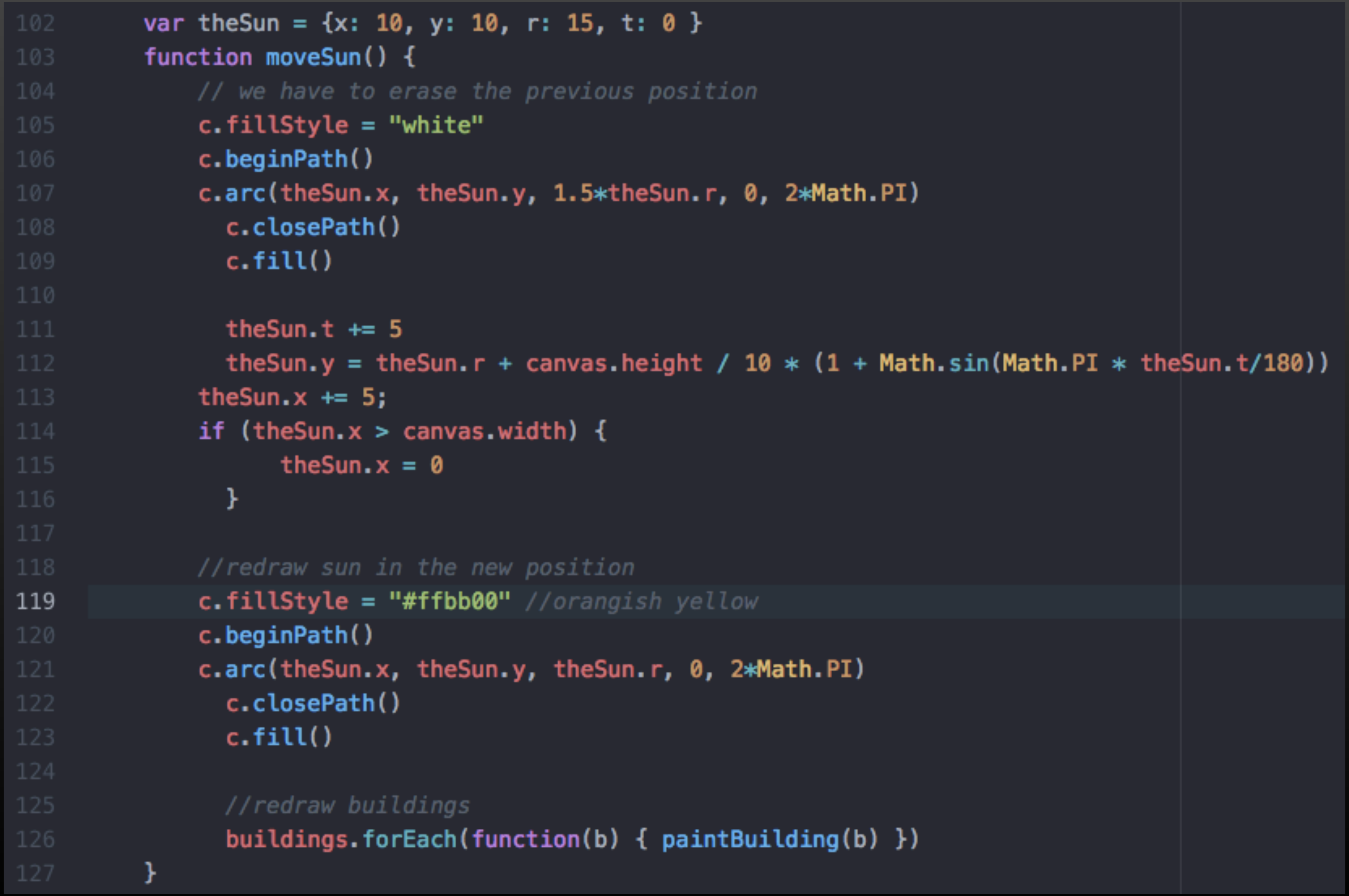
2. Which building should grow when the buildings overlap?
    - How big can the building grow?

3. Is the ground affected when the car moves?
    - Do we need to repaint the ground?

Design Decisions: Skyline Game



Design Decisions: How should the sun move across the sky?



```
    var theSun = {x: 10, y: 10, r: 15, t: 0 }
    function moveSun() {
        // we have to erase the previous position
        c.fillStyle = "white"
        c.beginPath()
        c.arc(theSun.x, theSun.y, 1.5*theSun.r, 0, 2*Math.PI)
          c.closePath()
          c.fill()

          theSun.t += 5
          theSun.y = theSun.r + canvas.height / 10 * (1 + Math.sin(Math.PI * theSun.t/180))
        theSun.x += 5;
        if (theSun.x > canvas.width) {
              theSun.x = 0
          }

        //redraw sun in the new position
        c.fillStyle = "#ffbb00" //orangish yellow
        c.beginPath()
        c.arc(theSun.x, theSun.y, theSun.r, 0, 2*Math.PI)
          c.closePath()
          c.fill()

          //redraw buildings
          buildings.forEach(function(b) { paintBuilding(b) })
    }
```

Design Decisions: How to handle overlapped buildings?

Option1: Do not allow overlapped buildings

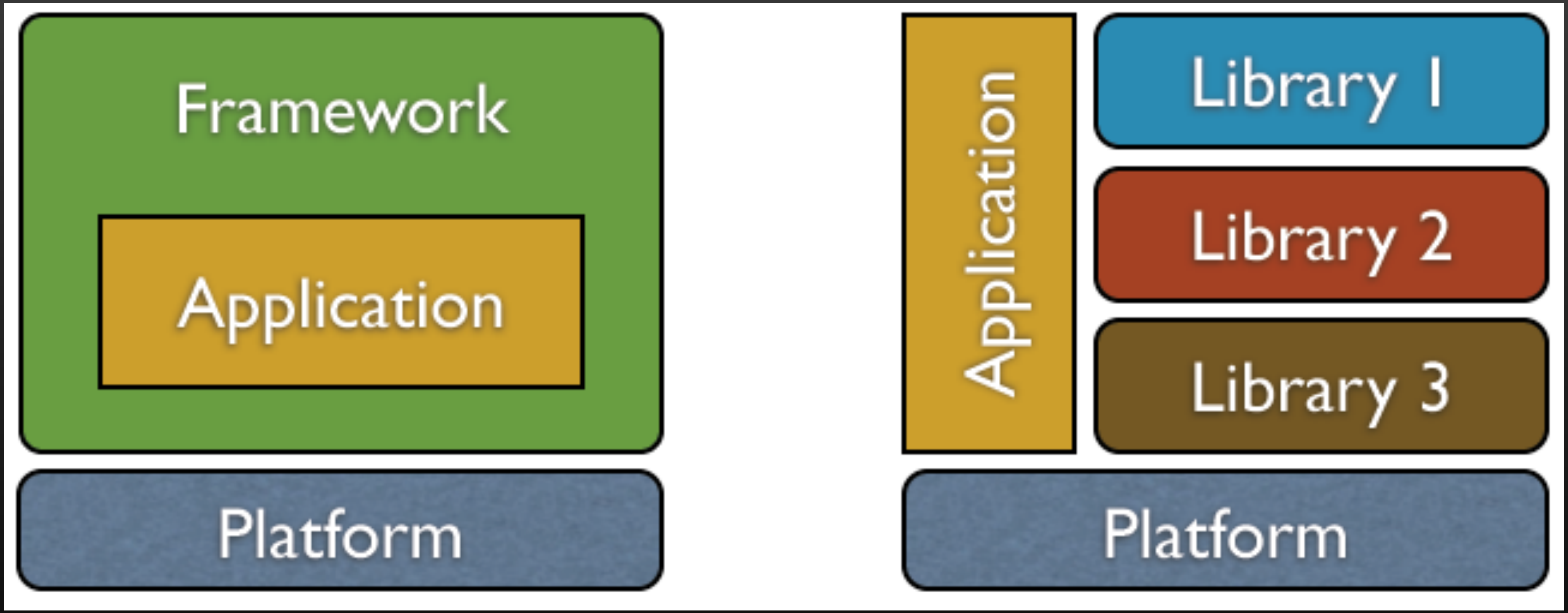
Option 2: Grow the last overlapped building drawn

Option 3: Grow all overlapped buildings

Design Decisions: Should moving car affect the ground?

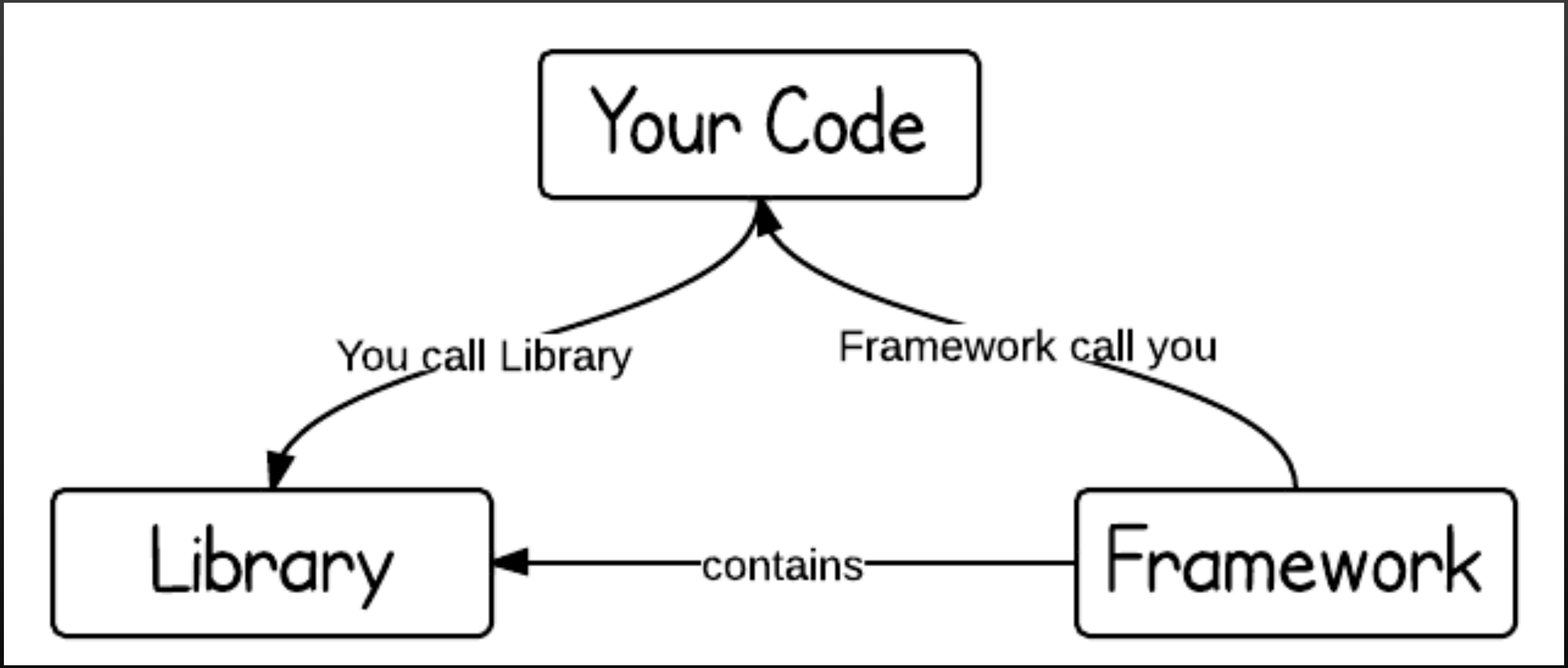
If no, then we don't have to redraw the ground

Libraries vs Frameworks



http://tom.lokhorst.eu/2010/09/why-libraries-are-better-than-frameworks

Inversion of Control



Library or Framework?

- Bootstrap

Library

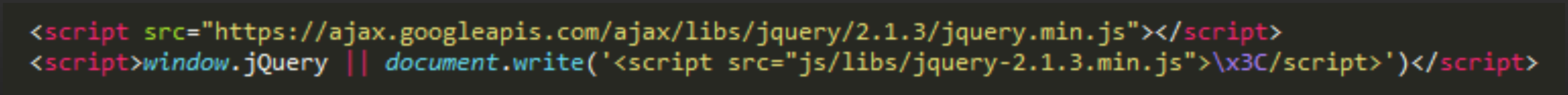
- jQuery

Library

- AngularJS

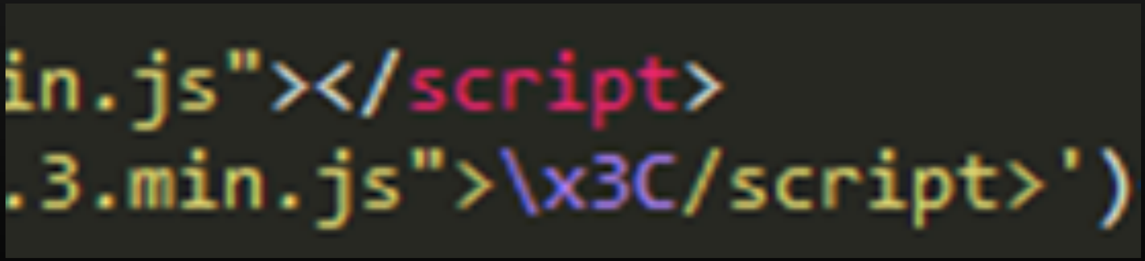
FRAMEWORK

Get jQuery



```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>
<script>window.jQuery || document.write('<script src="js/libs/jquery-2.1.3.min.js">\x3C/script>')</script>
```

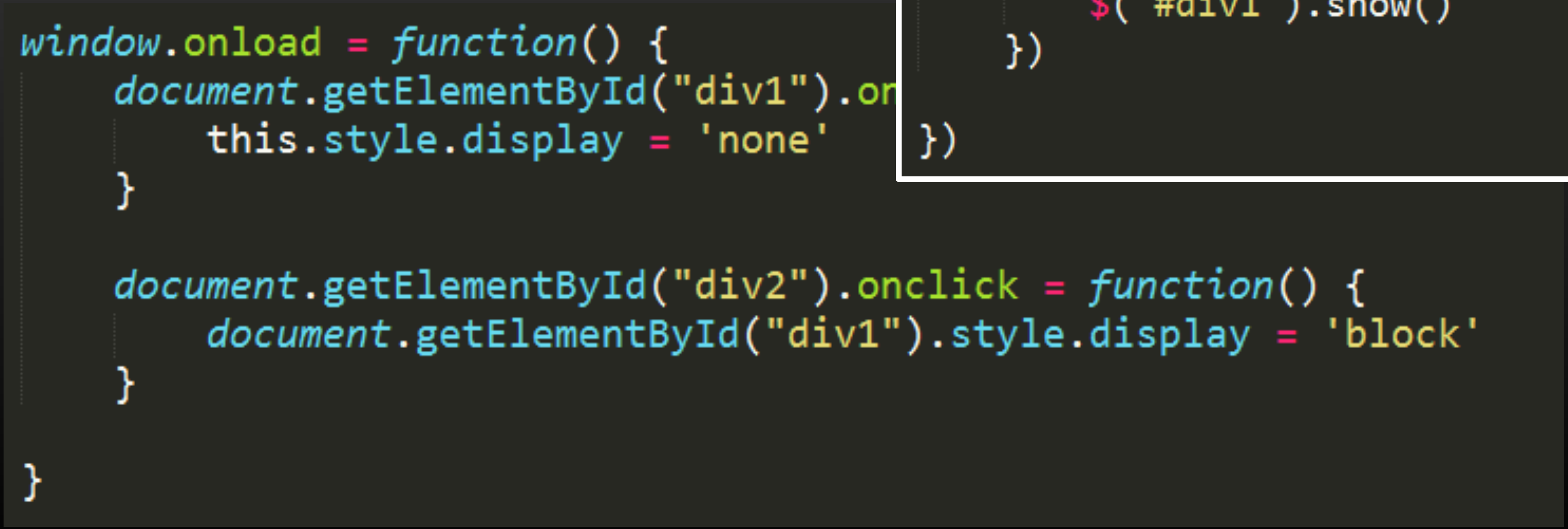
```
<script>window.jQuery || document.write('<script
```



```
in.js"></script>
.3.min.js">\x3C/script>')
```

jQuery JavaScript Library

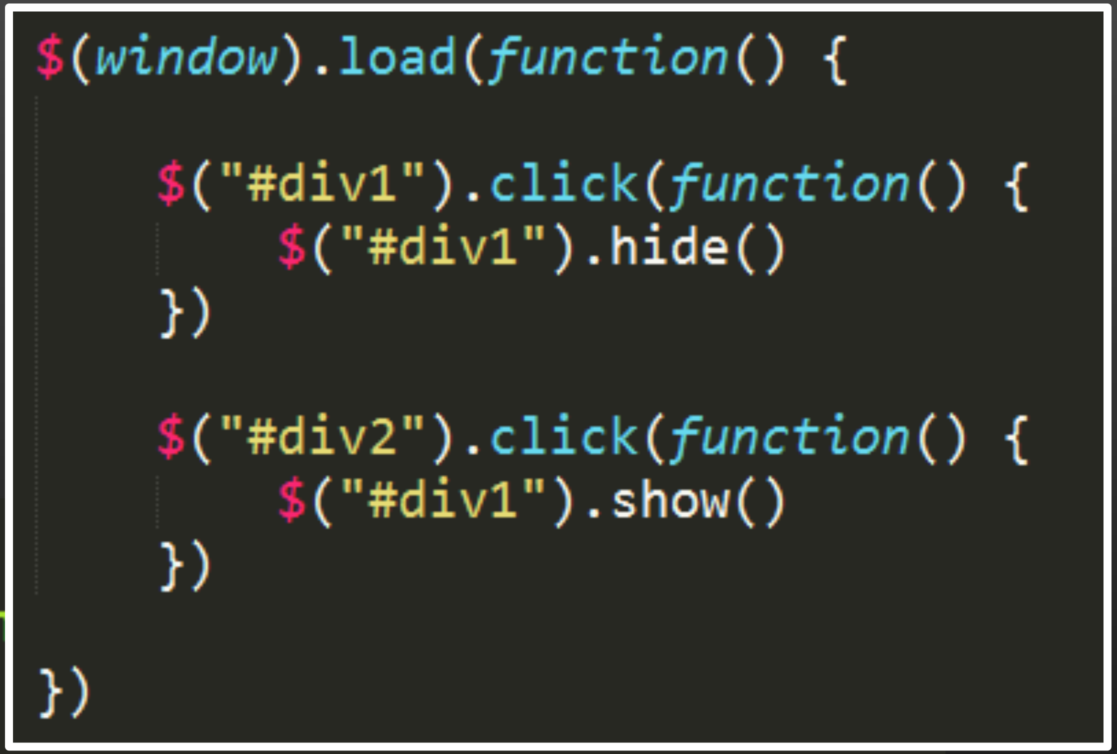
- Wrapper around DOM manipulation
- Instead of



```
window.onload = function() {
    document.getElementById("div1").on
        this.style.display = 'none'
    }

    document.getElementById("div2").onclick = function() {
        document.getElementById("div1").style.display = 'block'
    }

}
```



```
$(window).load(function() {

    $("#div1").click(function() {
        $("#div1").hide()
    })

    $("#div2").click(function() {
        $("#div1").show()
    })

})
```

jQuery uses CSS Selectors

- In fact, jQuery came first…

```
<div id="div1" style="background-color:blue;">
Click to Hide
</div>

<div id="div2" style="background-color:red;">
Click to Show
</div>
```

```
$(window).load(function() {

    $("#div1").click(function() {
        $("#div1").hide()
    })

    $("#div2").click(function() {
        $("#div1").show()
    })

})
```

Selector Rules (the easy ones)

- Tag

```
body {
    background-color: #FFFFFF;
}
```

- Class

```
.linkInverted {
    color: #FFFF00;
}
```

- Id

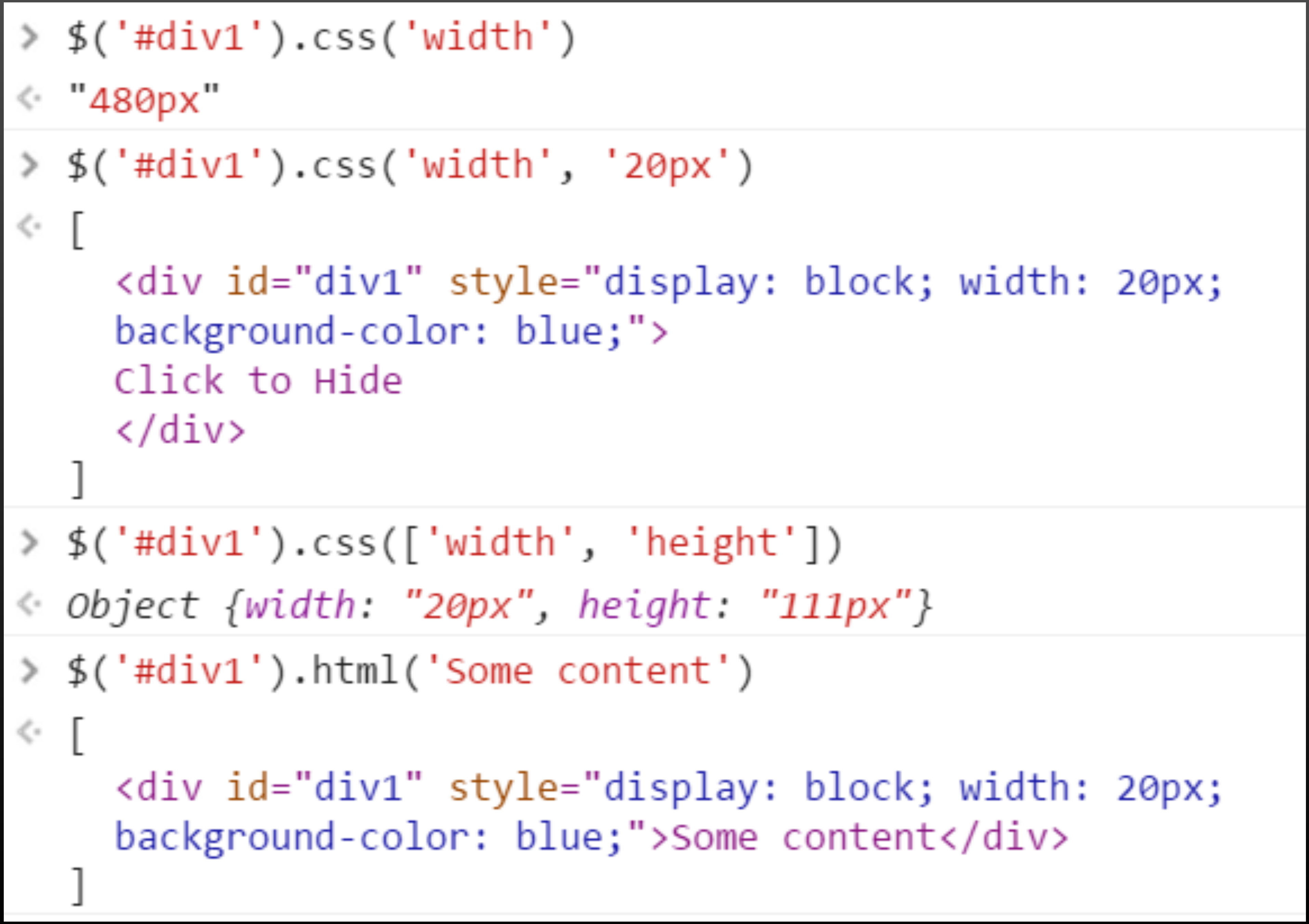
```
#riceLogo {
    width: 6em;
    margin-top:-1em;
    margin-bottom:-1em;
}
```

- Attribute

```
[name="fancy"] {
    font-size: 2em;
}
```

jQuery Manipulation

- $(…) returns a jQuery object or collection that is "easier" to manipulate than a DOM HTTP object.
- We just saw hide() and show() for updating the display style
- addClass(), hasClass(), removeClass(), toggleClass()
- parent(), siblings() → parents('div').last().siblings().children()
- $(..).get() = DOM element, useful for some things still

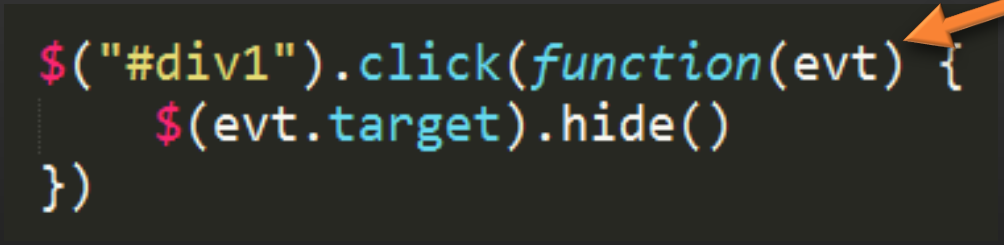


```
> $('#div1').css('width')
<· "480px"
> $('#div1').css('width', '20px')
<· [
     <div id="div1" style="display: block; width: 20px;
     background-color: blue;">
     Click to Hide
     </div>
   ]
> $('#div1').css(['width', 'height'])
<· Object {width: "20px", height: "111px"}
> $('#div1').html('Some content')
<· [
     <div id="div1" style="display: block; width: 20px;
     background-color: blue;">Some content</div>
   ]
```

jQuery Events

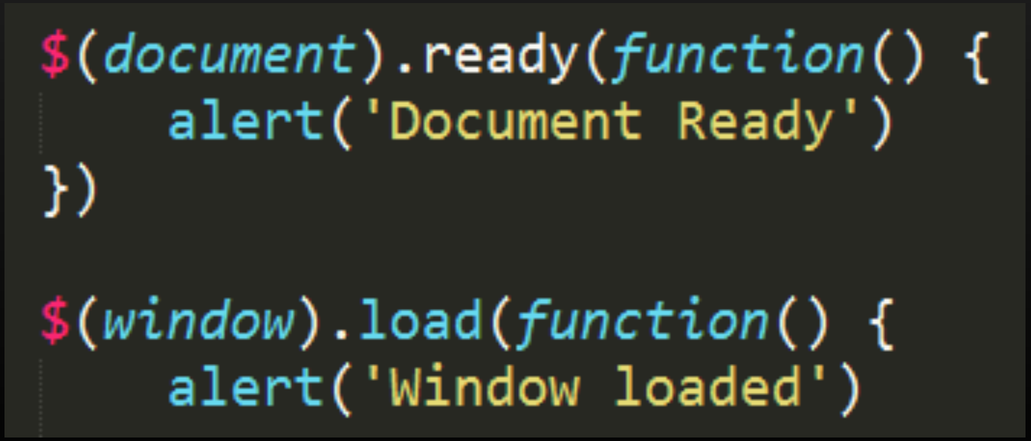
Regular DOM Event object

- Naming ala Level 2 DOM Events, onclick → click



```
$("#div1").click(function(evt) {
    $(evt.target).hide()
})
```

- Initial visit event ordering



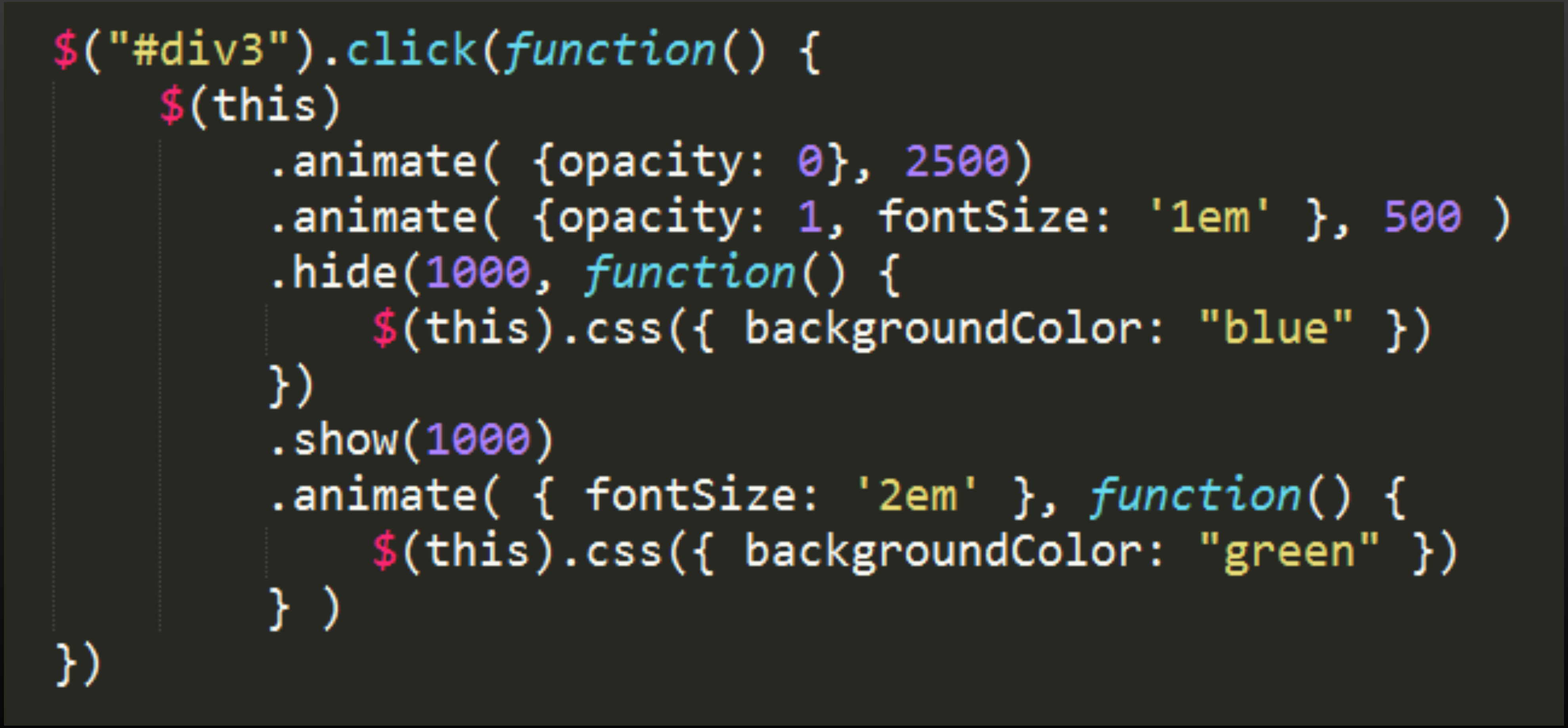
```
$(document).ready(function() {
    alert('Document Ready')
})

$(window).load(function() {
    alert('Window loaded')
```

jQuery Effects

- slideUp()
- hide(2000)
- fadeOut()
- delay()
- toggle()

jQuery Callback vs Chaining



```
$("#div3").click(function() {
    $(this)
        .animate( {opacity: 0}, 2500)
        .animate( {opacity: 1, fontSize: '1em' }, 500 )
        .hide(1000, function() {
            $(this).css({ backgroundColor: "blue" })
        })
        .show(1000)
        .animate( { fontSize: '2em' }, function() {
            $(this).css({ backgroundColor: "green" })
        } )
})
```