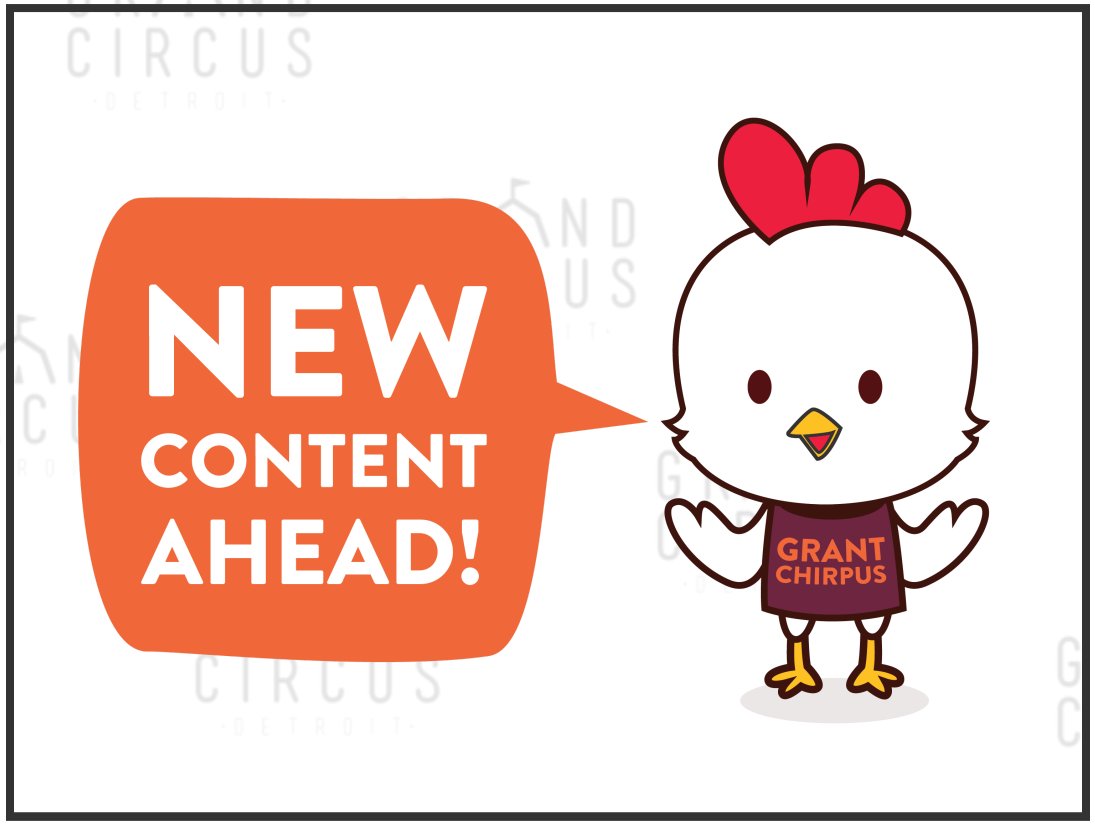

NEW
CONTENT
AHEAD!
GRANT
CHIRPUS

FUNCTIONS, SCOPE, & THE DOM

GOALS FOR THIS UNIT

1. Review
2. More on Functions
3. Variable Scope
4. The Document Object Model
5. JavaScript Events

REVIEW

DEMOS

MORE ON FUNCTIONS

FUNCTIONS

Functions are the special sauce that makes JavaScript such a cool language. Functions in JavaScript are first class objects, meaning:

- A function is an instance of the Object type
- A function can have properties and has a link back to its constructor method
- You can store the function in a variable
- You can pass the function as a parameter to another function
- You can return the function from a function

Let's look at a few of these.

FUNCTIONS

You can store a function in a variable.

```
function feedDog() {
  return "Kibble, canned food, and water";
}

var eveningChores = feedDog;

eveningChores();

// Or you can do this directly with an anonymous function

var feedDog = function() {
  return "kibble, canned food, and water";
};
```

FUNCTIONS

You can pass a function to a function as a parameter.

```
function doEveningChores(chores) {
  chores.forEach(function(chore){
    chore();
  });
}

doEveningChores([feedDog]);
```

FUNCTIONS

WTF was that forEach() thinger?! Glad you asked.

Not only can you pass functions as arguments, you can define them in-line like any other data type literal.

forEach is a method on the Array object that takes a function as an argument. That function is called on each element of the array receiving *it* as an argument.

...you can do the same with objects and arrays

FUNCTIONS

You can return functions from functions

```
function tonightChores(){
  return feedDog;
}

var tonight = tonightChores();
tonight();
```

VARIABLE SCOPE

LOCAL SCOPE

A variable declared within a function has local scope. It is only available to the function in which it's declared.

GLOBAL SCOPE

Variables with global scope are declared outside of a function. They can be accessed anywhere, but can cause problems with bigger, more complex applications. They take up memory and may cause namespace conflicts. (Bad things happen when two entirely separate variables have the same name.)

GLOBAL VS. LOCAL

Variables can have a global or a local scope.

- Variables in a local scope can access global variables.
- Global variables cannot access local variables.

VARIABLE SCOPE

```
var one = 1;

function doStuff() {
  console.log(one);
  var meaningOfLife = 42;
}

doStuff();
console.log(meaningOfLife);

// > 1
// > Syntax error: meaningOfLife is not defined.
```

VARIABLE SCOPE

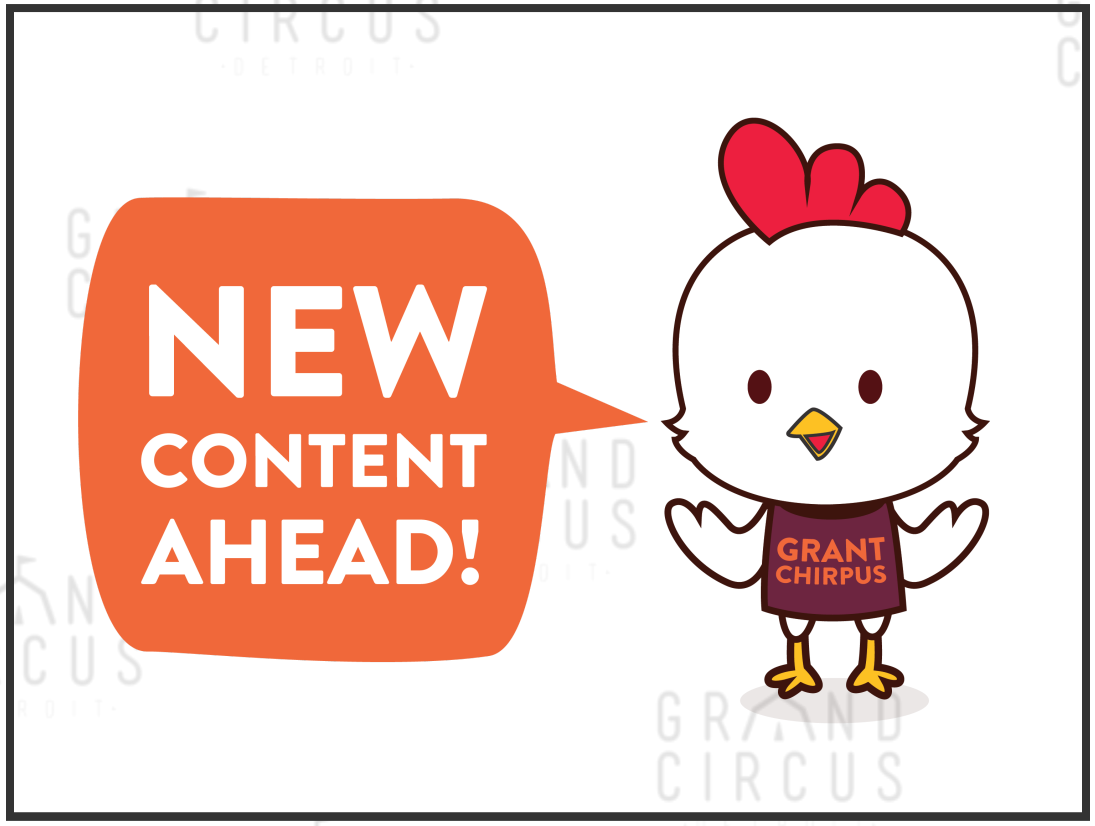
Key difference - there is no block level scoping

- What would you normally expect the output of this code to be?

```
var meaningOfLife = 0;
function doStuff() {
  console.log(meaningOfLife);
  if(true) {
    var meaningOfLife = 42;
  }
}

// > undefined (Note: not a syntax error and not 0? Why?)
```

QUESTIONS?


NEW
CONTENT
AHEAD!
GRANT
CHIRPUS

DOCUMENT OBJECT MODEL (DOM)

DOM

The document object model (DOM) is an interface which allows programs and scripts to dynamically access and update the content, style and structure of an HTML document.

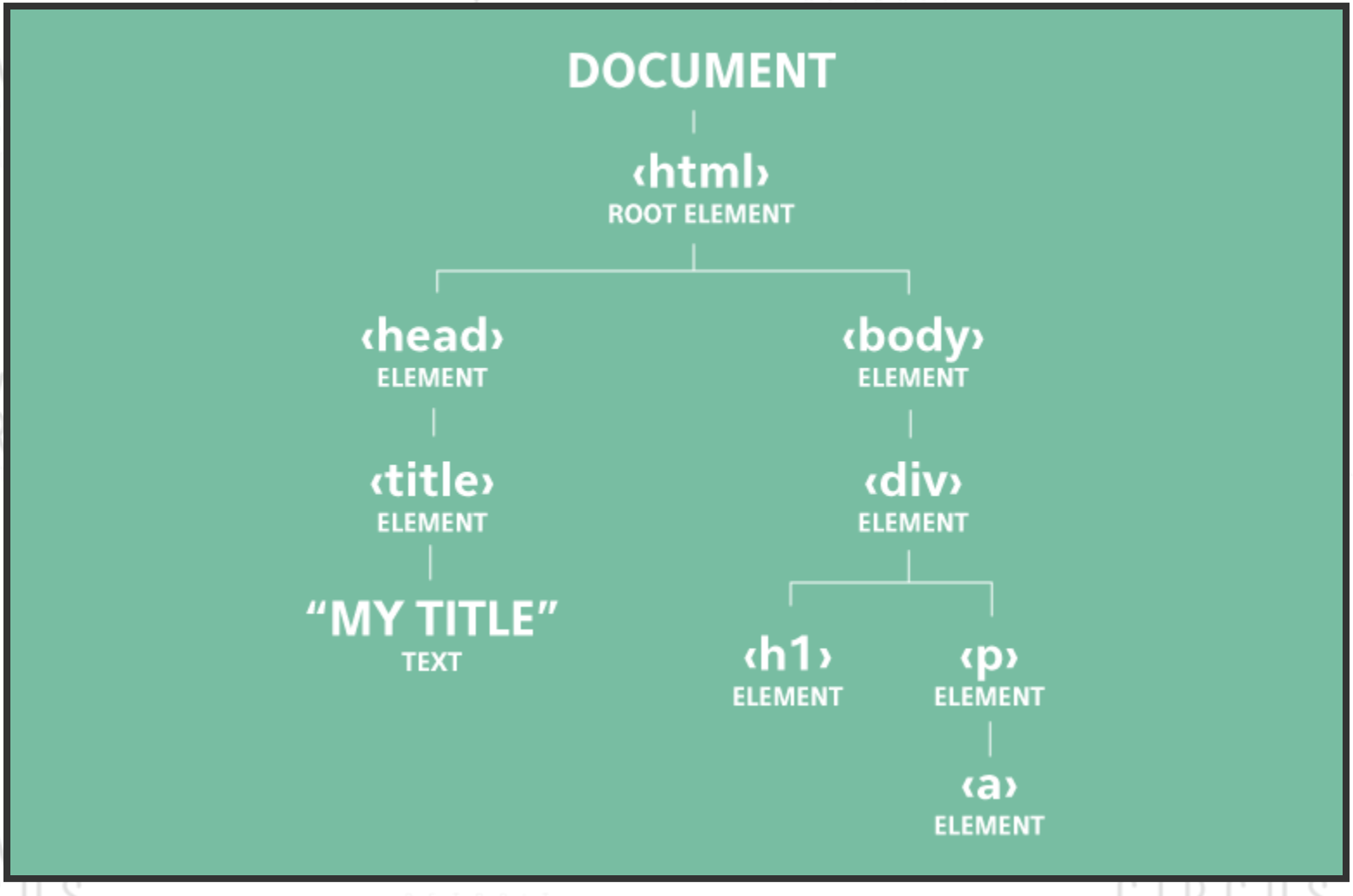
DOM

The DOM is a W3C (World Wide Web Consortium) standard and includes the Core DOM, XML DOM, and HTML DOM. The HTML DOM is a standard model for HTML documents and defines how to get, change, add, or delete HTML elements.

DOM

DOM

The DOM is a tree structure and identifies objects (elements) using nodes



STORING DOM OBJECTS

```
// body element
var bodyNode = document.body;

// html element
var htmlNode = document.body.parentNode;

// Array of all body's children
var childNodes = document.body.childNodes;
```

DOM

The DOM exposes a number of methods that are used to manipulate the structure of a page. Open any web page in your browser, open your developer tools, and run this command in the console. Each web page loaded in the browser has its own `document` object.

```
document.write('I changed the whole page! #rekt');
```

It’s possible to find elements on the HTML page by parent, sibling, or child node, but that’s time consuming and will make you crazy.

DOM

The better method is to use some of the provided DOM methods by tag, class, or ID.

```
<li class="food">Pizza</li>
<li class="food">Sushi</li>
<li class="food" id="favorite">Schwarma</li>
```

```
var listItems = document.getElementsByTagName('li');       // Array
var listItems = document.getElementsByClassName('food');  // Array
var favItem = document.getElementById('favorite');         // Object
```

DOM

The attributes of HTML elements can also be accessed and modified through the DOM

```
var img = document.getElementById('myImage');

img.getAttribute('src');
img.setAttribute('src', './images/newImage.jpg');
```

DOM

You can create nodes using the DOM and manipulate their contents using the `innerHTML` property. You can add them to a page by using `appendChild.`

```
var newElement = document.createElement('div');

newElement.innerHTML = '<h1>Hi Everybody!</h1> <p>Hi Dr. Nick!</p>';
document.body.appendChild(newElement);
```

EVENTS

Many user interactions, or *events* are registered by the browser.

We can write code that *triggers* or runs, when a certain event is registered.

UI EVENTS

Event	Description
`load`	when a page finishes loading
`unload`	when a page is unloading
`error`	when a JavaScript or asset error occurs
`resize`	when the browser window is resized
`scroll`	when the user scrolls

KEYBOARD EVENTS

Event	Description
`keydown`	while key is depressed down
`keyup`	when depressed key is released
`keypress`	when a character is inserted

MOUSE EVENTS

Event	Description
`click`	button is pressed and released on same element
`dblclick`	button is pressed and released twice on same element
`mousedown`	button is pressed on an element
`mouseup`	button is released over an element
`mousemove`	mouse moved (not on touch screen)
`mouseover`	mouse moved over an element (not on touch screen)
`mouseout`	mouse moved off of an element (not on touch screen)

EVENTS

There are also:

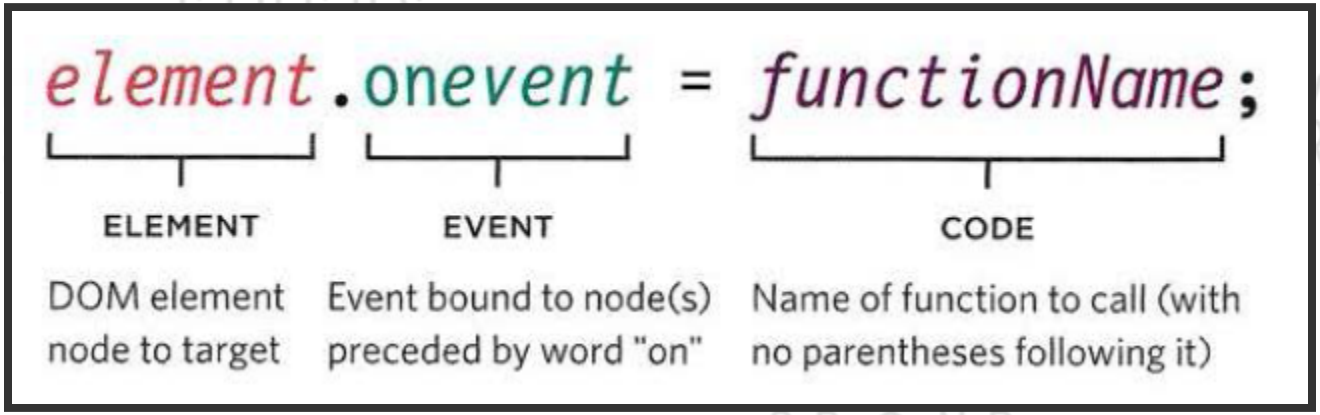
- focus events when focus is on or leaves an object
- form events for form interactions
- and many others

EVENTS

When an event *fires (is raised)* or occurs it can trigger a function.

We can *bind* or connect an element with an event with an *event handler*

DOM EVENT HANDLERS



QUESTIONS?

RECAP

You should understand and be able to use:

- Features of functions
- Local vs global scope
- The document object model
- Events

HOMEWORK

MDN's Intro to Object-Oriented JavaScript

From JavaScript & jQuery:

- Chapter 3: 97
- Chapter 10: 456-457

LAB 8

DOM SHOPPING LIST

INSTRUCTIONS

Extend the shopping list program from the last lab.

1. Set up a basic HTML page.
2. Append the items and their prices from the shopping list to the page.
3. Show the total somewhere on the page.

BONUS!

Add a form with text inputs for Name and Price and a button that allows you to add elements to the shopping list.

- Clicking 'Add' updates the list on the page.
- Clicking 'Add' also updates the total.

Be prepared to demo your work.

PRE-GAME

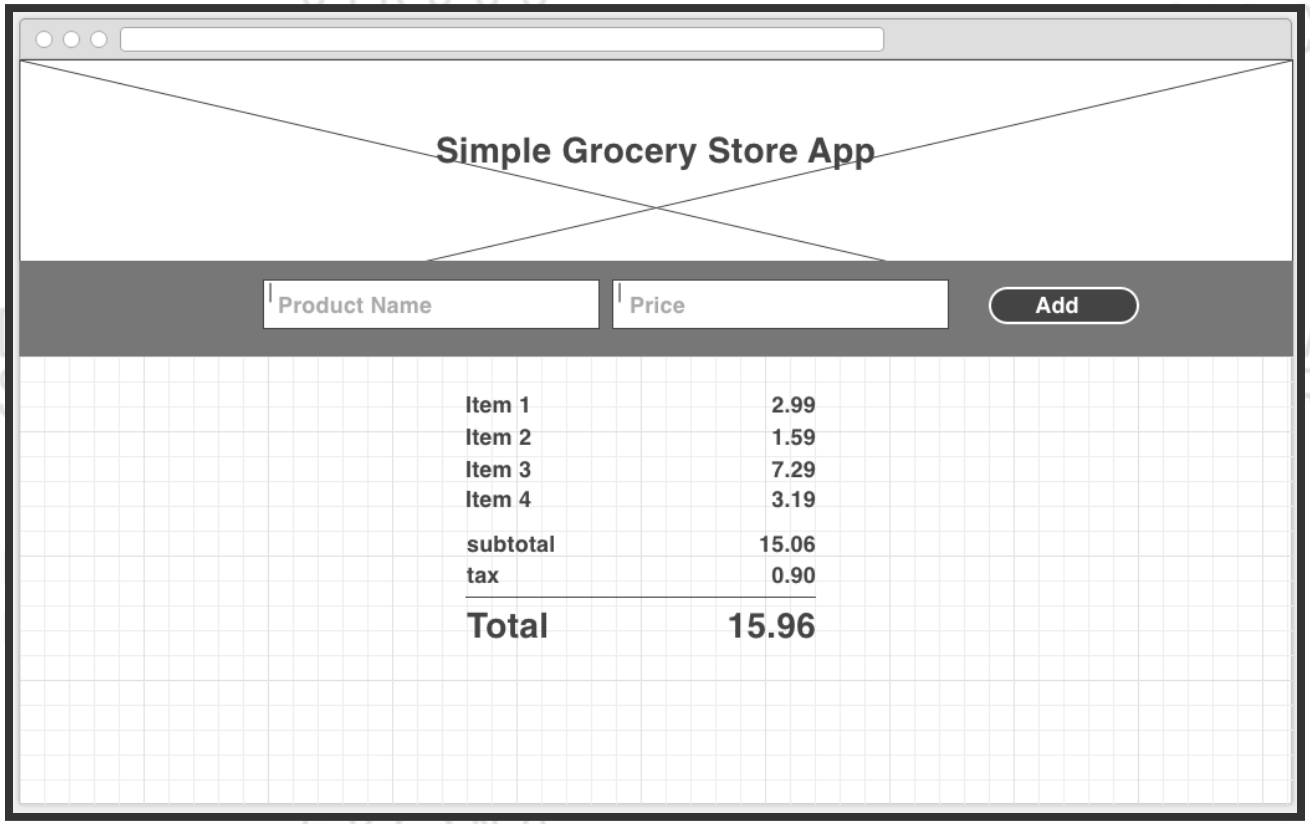

Simple Grocery Store App
Product Name
Price
Add
Item 1 2.99
Item 2 1.59
Item 3 7.29
Item 4 3.19
subtotal 15.06
tax 0.90
Total 15.96

FIGURE IT OUT

Write a JavaScript program to calculate the volume of a sphere from a user's input. Include appropriate error messages as alerts if the input is a negative number or not a number at all.