GENERAL ASSEMBLY

INTRODUCTION TO JAVASCRIPT & JQUERY

Evan Johnson, Developer

EVAN JOHNSON

- Front-end Engineer, Amazon
- Self-taught
- Loves JavaScript

johnson.evan1@gmail.com

evblurbs @ev_blurbs

Source:

AGENDA

- What is JavaScript?
- Tools
- Functions (Intro)
- Variables
- Operators
- Objects
- Functions
- Break

- jQuery Intro
- Inline vs External JS
- jQuery Selector
- jQuery DOC Ready
- jQuery Events
- Control Flow
- jQuery Animations
- jQuery AJAX
- Breakpoints

WHAT IS JAVASCRIPT?

WHAT IS JAVASCRIPT?

“an object-oriented computer programming language commonly used to create **interactive effects within web browsers.**”

- Google Search

“is a dynamic computer programming language.”

- Wikipedia

I Am Devloper
@iamdevloper

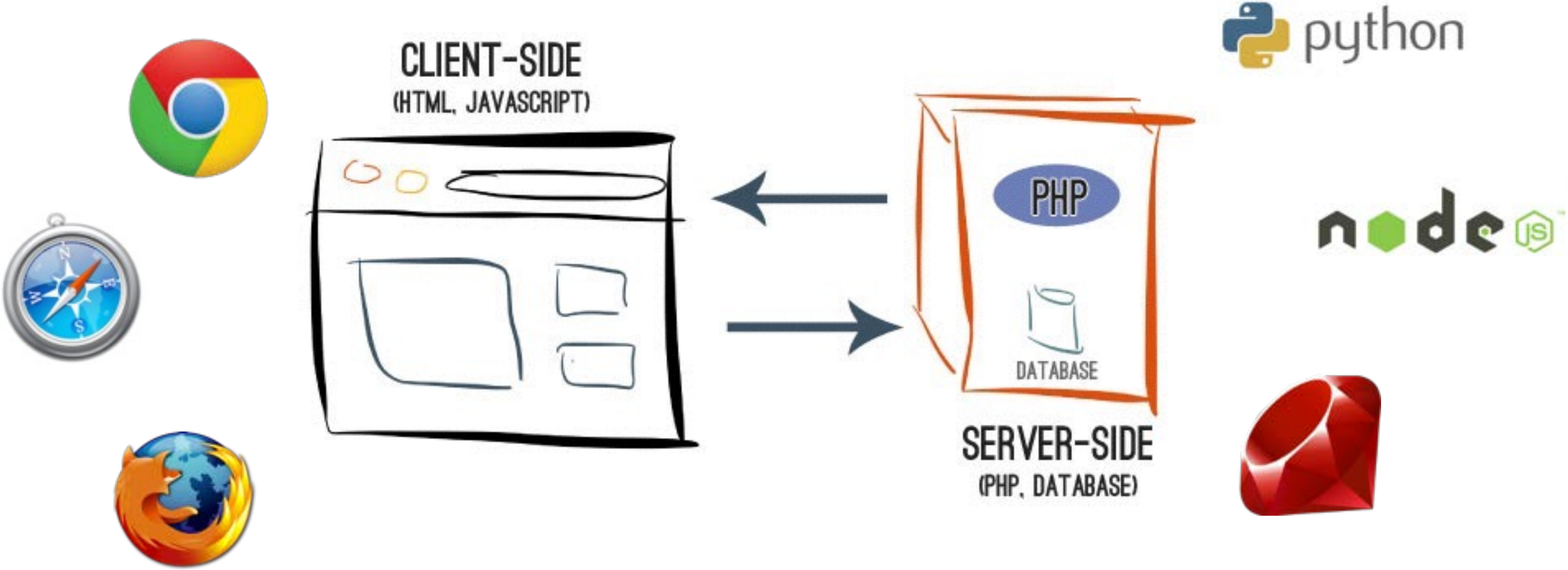
JavaScript is greatest thing in the world, apart from Breaking Bad...or maybe The Wire.

9:31 AM - 5 Oct 2013

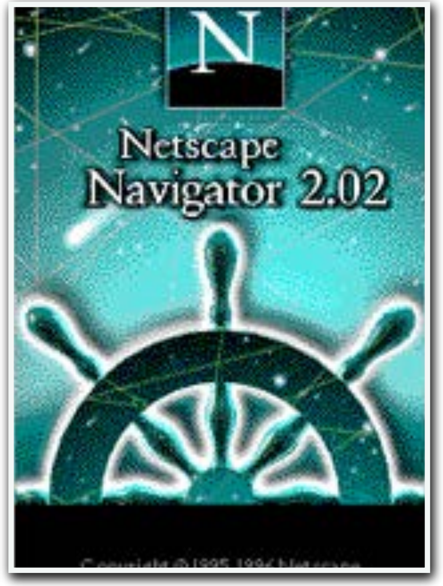
67 RETWEETS 33 FAVORITES

WHAT IS JAVASCRIPT?

CLIENT-SIDE VS. SERVER-SIDE



WHAT IS JAVASCRIPT?



- LiveScript was created by Brendan Eich in 10 days in May 1995
- Eich worked at Netscape, who was trying to enhance Netscape 2.0
- Netscape changed the name to JavaScript in a marketing ploy
- IE adopted JavaScript in August 1996 when releasing v.3.0
- Ecma Internation standardized JavaScript in 1999 (ECMAScript)

TOOLS

SUBLIME TEXT

- A text editor for code
- Can extend with packages (i.e. syntax highlighting, linting, etc.). Requires installing Package Control (https://packagecontrol.io/)
- http://www.sublimetext.com/

CHROME

- Webbrowser
- Advanced developer tools
- https://www.google.com/chrome/browser/desktop/

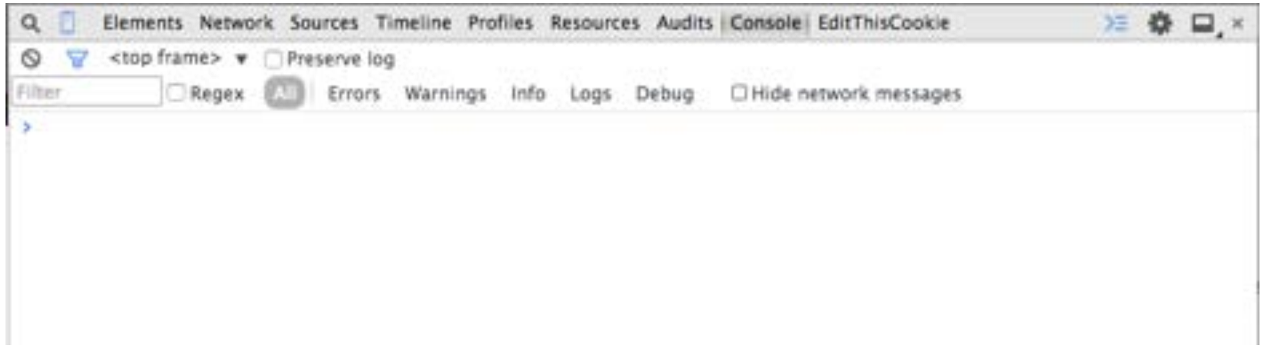
TOOLS

EXERCISE FILES

- https://github.com/evblurbs/intro-to-javascript
- Download files using git clone or the ‘Download Zip’ button

CONSOLE

- Chrome Menu (top right hamburger button) -> Tools -> Developer Tools
- Mac shortcut: Cmd + Opt + i
- PC shortcut: F12, Ctrl + Shift + i



FUNCTIONS (INTRO)

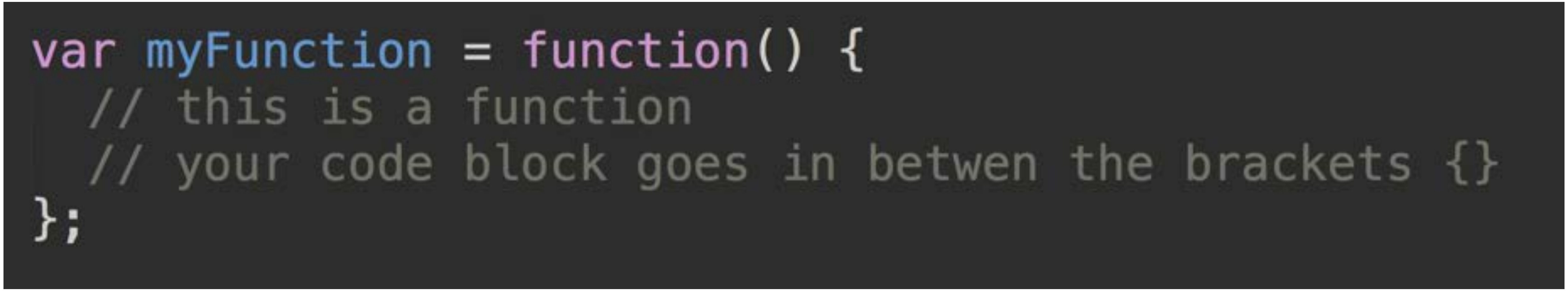
FUNCTIONS (INTRO)

- "A **JavaScript function** is a block of code designed to perform a particular task. A JavaScript function is executed when "something" **invokes it** (calls it)."
  - W3Schools

  Invoke: “to make use of (a law, a right, etc.)” - Merriam-Webster

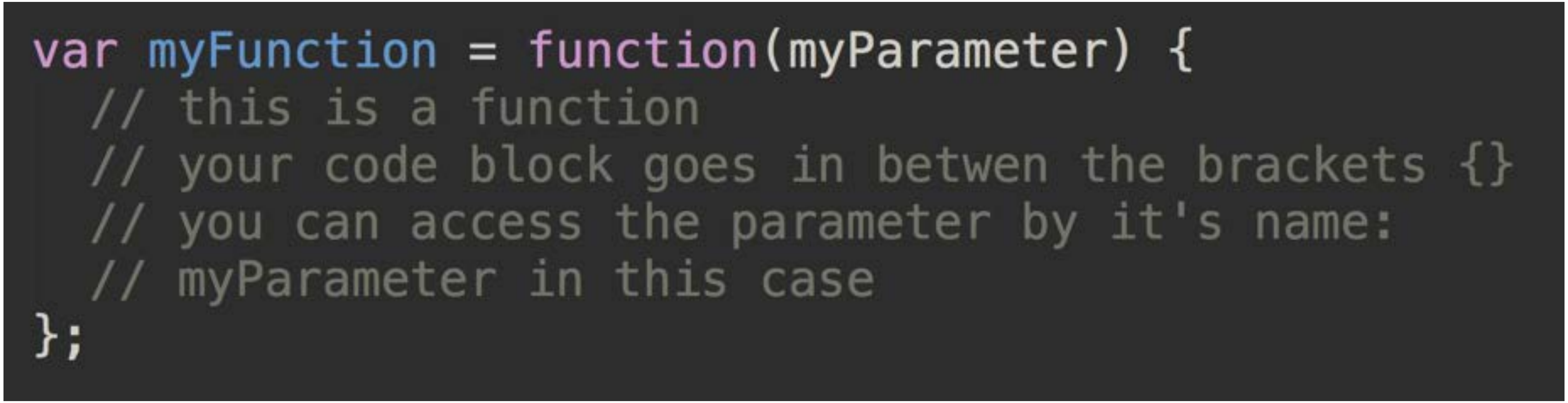
- Invoke (JS): To initiate a block of code, that would otherwise not be executed.

BASIC FUNCTION



```
var myFunction = function() {
  // this is a function
  // your code block goes in betwen the brackets {}
};
```

A FUNCTION WITH A PARAMETER



```
var myFunction = function(myParameter) {
  // this is a function
  // your code block goes in betwen the brackets {}
  // you can access the parameter by it's name:
  // myParameter in this case
};
```

Open the exercise file 001-function-intro.html in Chrome, and we’ll start invoking some functions!

EXERCISE

File: 001-functions-intro.hmtl

Let me know if you get stuck, or have any questions!

VARIABLES

VARIABLES

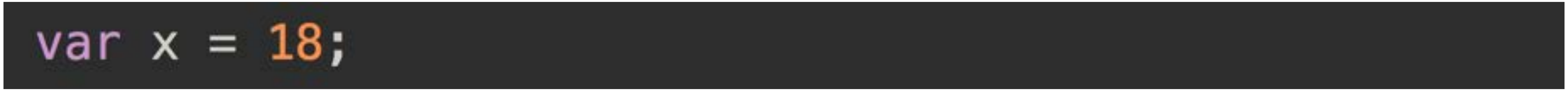
> "You use variables as **symbolic names** for values in your application. The names of variables, called **identifiers**, conform to certain rules."
>
> – Mozilla Developer Network (MDN)

Rules for variable names:

- Must start with a letter, underscore (_), or dollar sign($)
- Subsequent letters can also be digits (0-9)
- Variable names are case sensitive

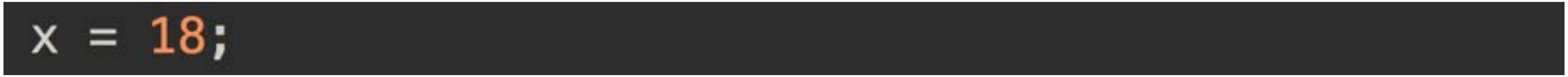
DECLARING VARIABLES (TWO METHODS)

1) With the keyword var, you can declare both local and global variables.



```
var x = 18;
```

2) By simply assigning it a value. This always declares a global variable. You shouldn’t use this variant.



```
x = 18;
```

PRIMITIVE VARIABLES

“A primitive (primitive value, primitive data type) is data that is not an Object and does not have any methods.” - MDN

Type	Example
Numbers	42, 3.14159
Boolean	true / false
Strings	“howdy"
undefined	value is undefined

EXERCISE

File: 002-variables.hmtl

Let me know if you get stuck, or have any questions!

OPERATORS

OPERATORS

JavaScript operators are symbols that can be used to assign values, compare values, preform arithmetic, test logic, manipulate strings, and more.

Let’s cover these operators first:

- Arithmetic operators
- Assignment operators
- String operators

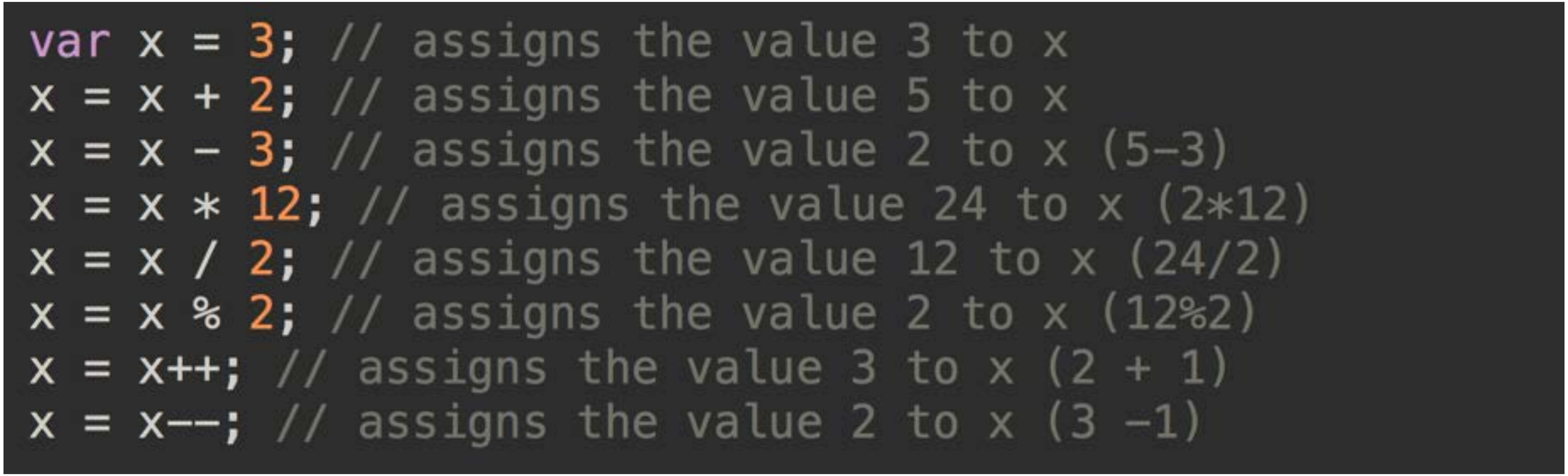
OPERATORS

ARITHMETIC OPERATORS

Operator	Description
+	Addition
-	Subtraction
*	Multiplaction
/	Division
%	Modulus
++	Increment
—	Decrement

OPERATORS

ARITHMETIC OPERATORS

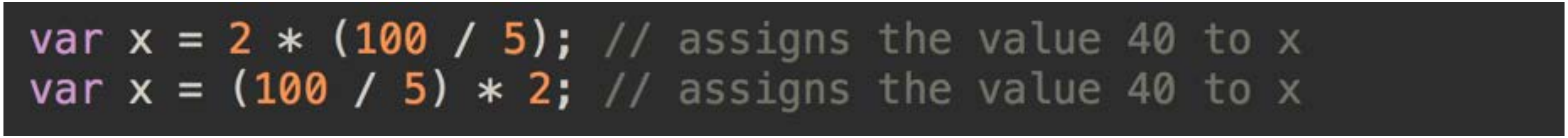


```
var x = 3; // assigns the value 3 to x
x = x + 2; // assigns the value 5 to x
x = x - 3; // assigns the value 2 to x (5-3)
x = x * 12; // assigns the value 24 to x (2*12)
x = x / 2; // assigns the value 12 to x (24/2)
x = x % 2; // assigns the value 2 to x (12%2)
x = x++; // assigns the value 3 to x (2 + 1)
x = x--; // assigns the value 2 to x (3 -1)
```

OPERATORS

ARITHMETIC OPERATORS

Parenthesis can be used to indicate which part of an equation you want to evaluate first.

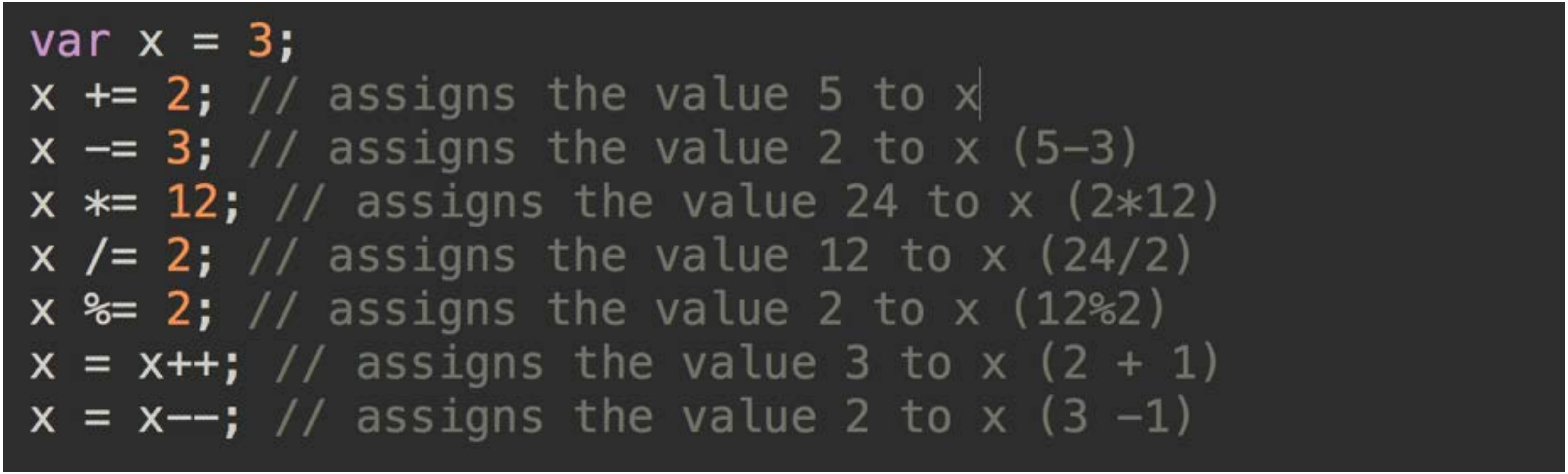


```
var x = 2 * (100 / 5); // assigns the value 40 to x
var x = (100 / 5) * 2; // assigns the value 40 to x
```

ASSIGNMENT OPERATORS

Shorthand	Meaning
x += y	x = x + y
x -= y	x = x - y
x *= y	x = x * y
x /= y	x = x / y
x %= y	x = x % y

ASSIGNMENT OPERATORS



```
var x = 3;
x += 2; // assigns the value 5 to x
x -= 3; // assigns the value 2 to x (5-3)
x *= 12; // assigns the value 24 to x (2*12)
x /= 2; // assigns the value 12 to x (24/2)
x %= 2; // assigns the value 2 to x (12%2)
x = x++; // assigns the value 3 to x (2 + 1)
x = x--; // assigns the value 2 to x (3 -1)
```

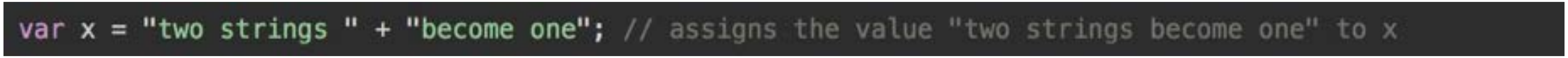
EXERCISE

File: 003.1-operators.hmtl

Let me know if you get stuck, or have any questions!

STRING OPERATORS

The + symbol is used as the addition operator with numbers, but it become the concatenation operator with strings. It combines two string values together.



```
var x = "two strings " + "become one"; // assigns the value "two strings become one" to x
```

EXERCISE

File: 003.2-operators.hmtl

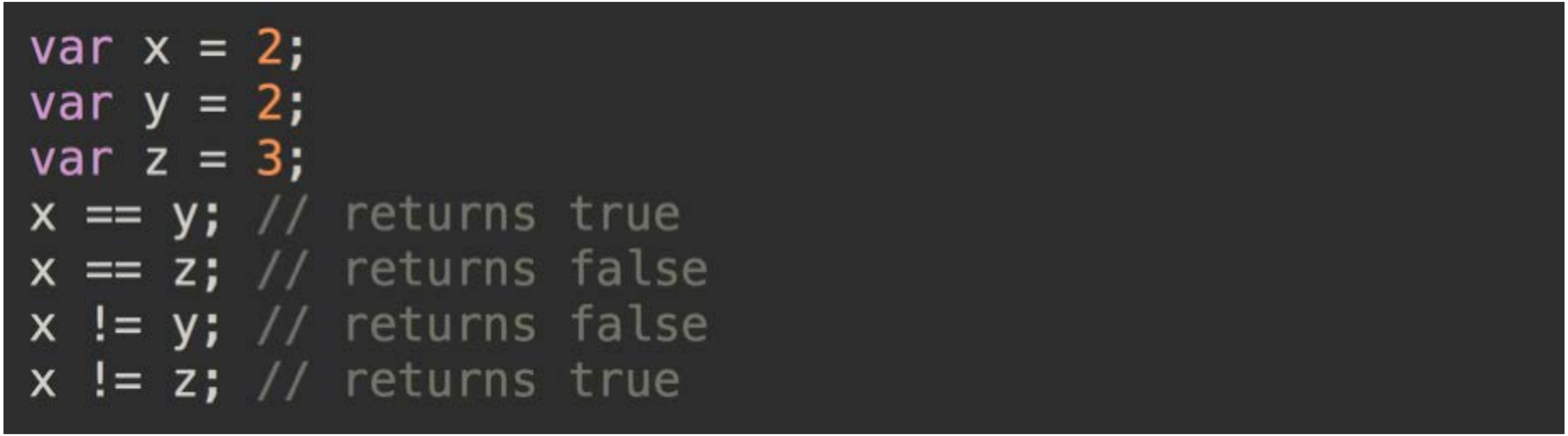
Let me know if you get stuck, or have any questions!

COMPARISON OPERATORS

> "A **comparison operator** compares its operands and returns a logical value based on whether the **comparison is true**."
>
> – Mozilla Developer Network (MDN)

Operator	Description
Equal (==)	Returns true if the operands are equal
Not equal (!=)	Returns true if the operands are not equal

COMPARISON OPERATORS (EQUAL AND NOT EQUAL)



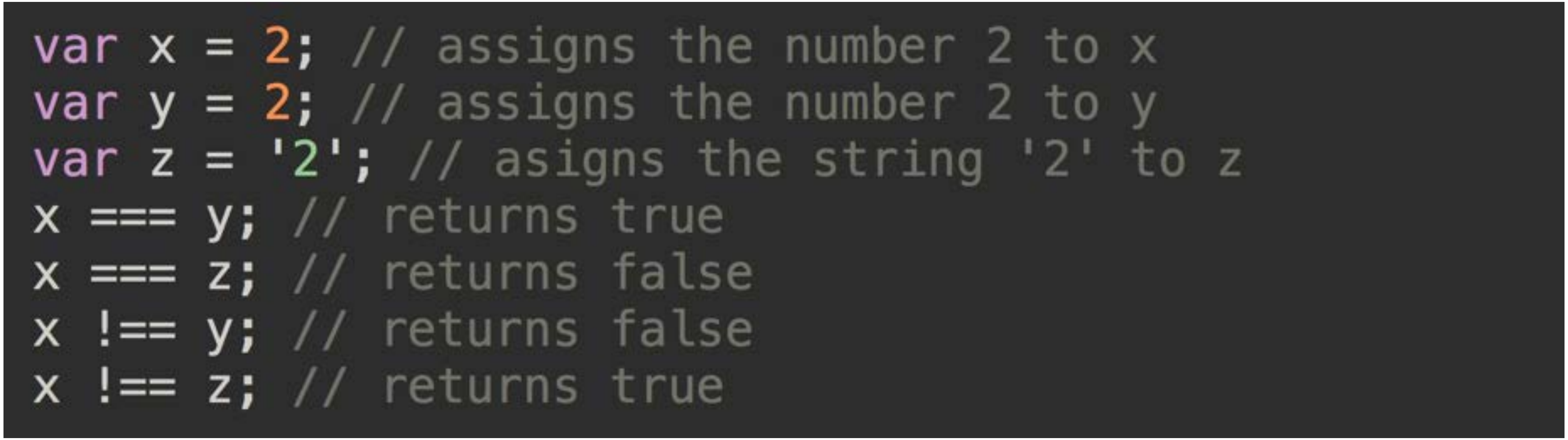
```
var x = 2;
var y = 2;
var z = 3;
x == y; // returns true
x == z; // returns false
x != y; // returns false
x != z; // returns true
```

COMPARISON OPERATORS

Strict equal and strict not equal operators not only compare the value, but also the data type.

Operator	Description
Strict equal (===)	Returns true if the operands are equal and of the same type.
Strict not equal (!==)	Returns true if the operands are not equal and/or not of the same type.

COMPARISON OPERATORS (STRICT EQUAL AND STRICT NOT EQUAL)

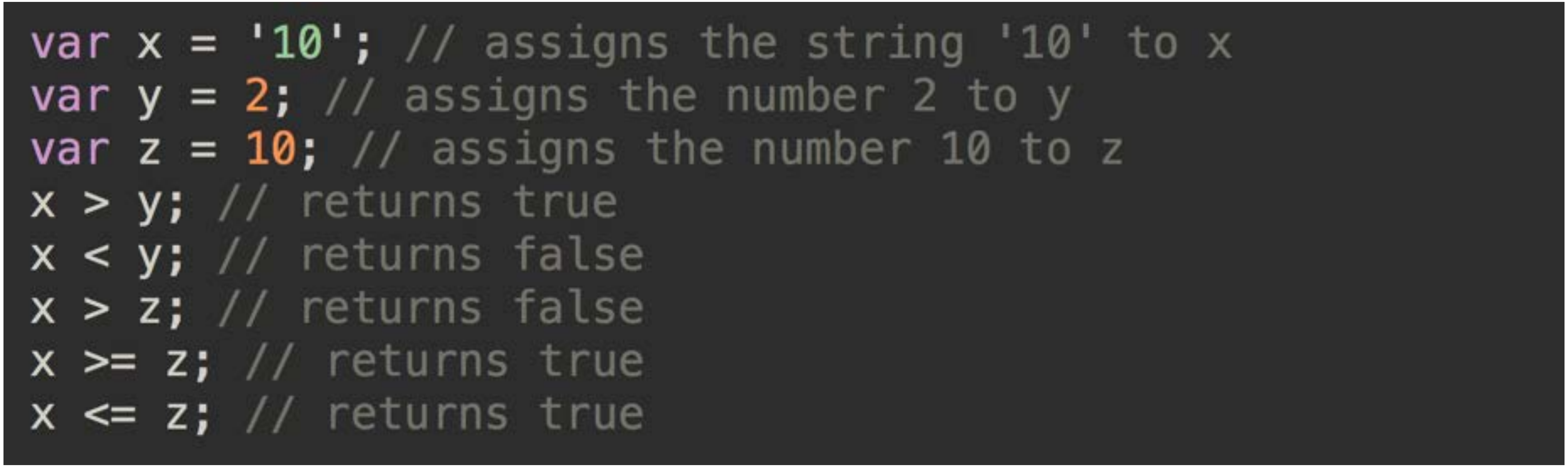


```
var x = 2; // assigns the number 2 to x
var y = 2; // assigns the number 2 to y
var z = '2'; // asigns the string '2' to z
x === y; // returns true
x === z; // returns false
x !== y; // returns false
x !== z; // returns true
```

COMPARISON OPERATORS

Operator	Description
Greater than (>)	Returns true if the left operand is greater than the right operand.
Greater than or equal (>=)	Returns true if the operands are not equal and/or not of the same type.
Less than (<)	Returns true if the left operand is less than the right operand.
Less than or equal (<=)	Returns true if the left operand is less than or equal to the right operand.

COMPARISON OPERATORS (<, <=, >, >=)



```
var x = '10'; // assigns the string '10' to x
var y = 2; // assigns the number 2 to y
var z = 10; // assigns the number 10 to z
x > y; // returns true
x < y; // returns false
x > z; // returns false
x >= z; // returns true
x <= z; // returns true
```

OPERATORS

EXERCISE

File: 003.3-operators.hmtl

Let me know if you get stuck, or have any questions!

INTRODUCTION TO JAVASCRIPT & JQUERY

OBJECTS

WHAT IS AN OBJECT

> "The Object constructor creates an object wrapper."
>
> – Mozilla Developer Network (MDN)

- Object in layman's terms: While a variable is a symbol for one specific values, an object is a symbol that can contain many variables.

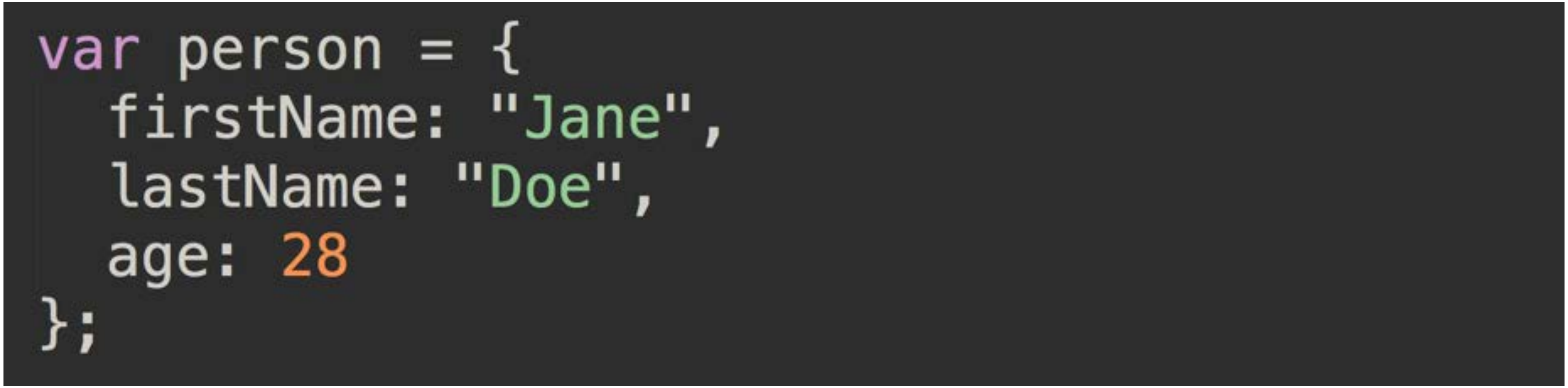
OBJECTS

SYNTAX

- Are created with brackets ({})
- Contain key/value pairs separated by a semi-colon (:)
- The key/value pairs are separated by a comma (,)
- Object values are accessed by combining the object name and key with a period (i.e. objectName.key)

Source:

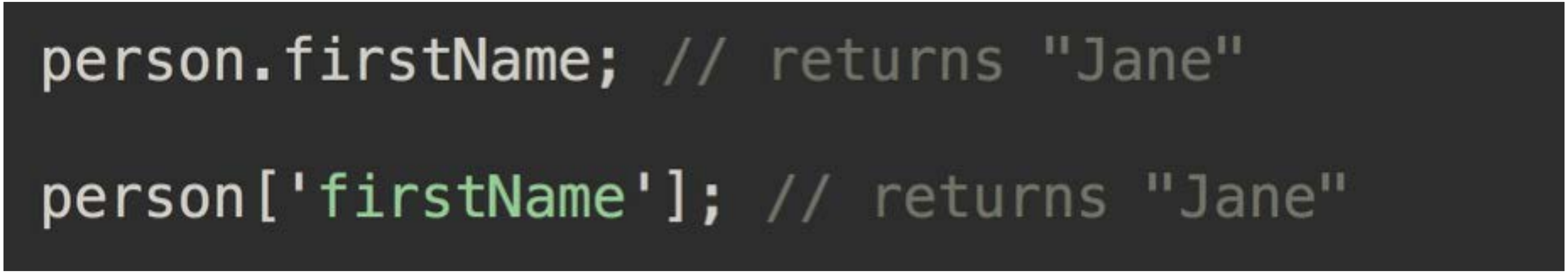
EXAMPLE OBJECT



```
var person = {
  firstName: "Jane",
  lastName: "Doe",
  age: 28
};
```

OBJECT

ACCESSING AN OBJECTS’S VARIABLES



```
person.firstName; // returns "Jane"

person['firstName']; // returns "Jane"
```

EXERCISE

File: 004-objects.hmtl

Let me know if you get stuck, or have any questions!

INTRODUCTION TO JAVASCRIPT & JQUERY

FUNCTIONS

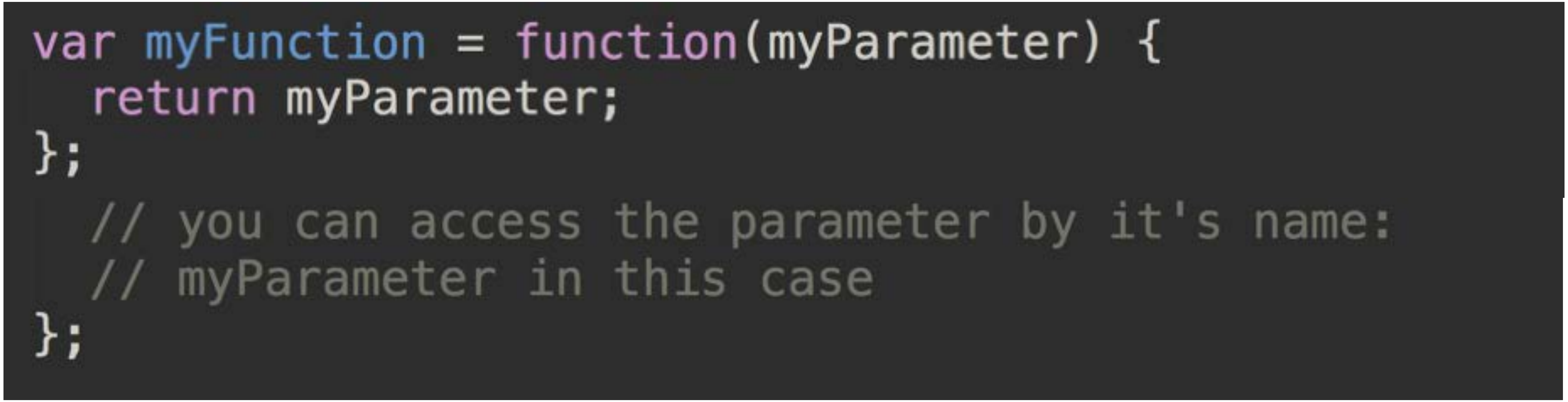
FUNCTIONS

- "A **JavaScript function** is a block of code designed to perform a particular task. A JavaScript function is executed when "something" **invokes it** (calls it)."

  – W3Schools

- Invoke (JS): To initiate a block of code, that would otherwise not be executed.

A FUNCTION IS ACTUALLY AN OBJECT?



```
var myFunction = function(myParameter) {
  return myParameter;
};
  // you can access the parameter by it's name:
  // myParameter in this case
};
```

SYNTAX

- Are created with the keyword: *function name()*
- You can pass in as many parameters inside the parenthesis: *function name(param)*
- Multiple parameters are separated with commas: *function name(param1, param2, param3)*
- You statement/code is wrapped in brackets after the parenthesis: *function name() { [statement] };*
- After the closing bracket, you end the function with a semi-colon (;)

Source:

EXERCISE

File: 005-functions.hmtl

Let me know if you get stuck, or have any questions!

JQUERY INTRO

WHAT IS IT?

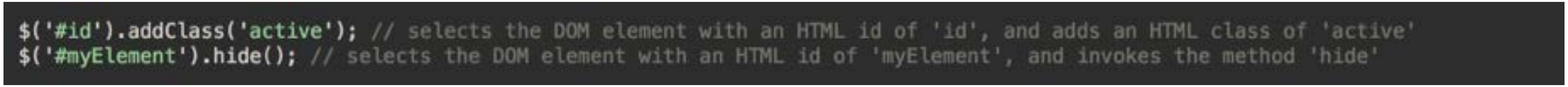
jQuery is a JavaScript library, or a collection of functions/methods and objects that you can utilize. “It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers.” - jQuery

jQuery motto: **“write less, do more”**

$ SYMBOL

jQuery, unless assigned otherwise, assigns its methods to the **$** symbol.

This means when calling jQuery methods, you typically start your call with the **$** symbol.



```
$('#id').addClass('active'); // selects the DOM element with an HTML id of 'id', and adds an HTML class of 'active'
$('#myElement').hide(); // selects the DOM element with an HTML id of 'myElement', and invokes the method 'hide'
```

CONTENT DELIVERY NETWORKS (CDN)

- “A **content delivery network** or **content distribution network** (CDN) is a large distributed system of servers deployed in multiple data centers across the Internet. The goal of a CDN is to serve content to end-users with high availability and high performance.”

  - Wikipedia

EXERCISE

File: 006-jquery-cdn.hmtl

Let me know if you get stuck, or have any questions!

INLINE VS EXTERNAL JS

INLINE VS EXTERNAL JS

INLINE JAVASCRIPT

Inline JavaScript is added directly to the DOM between a <script> and </script> tag. If your not using the HTML5 Document type, then you need to define a type on the opening script tag:
<script type=“text/javascript”></script>

Not a maintainable format. Use this only for code that should be fired immediately, and doesn’t deserve it’s own file (I.e. Google Analytics code).

INLINE VS EXTERNAL JS

EXTERNAL JAVASCRIPT

External JavaScript is JavaScript that lives in a separate file than the DOM, but is referenced by the DOM.
<script src=“/path/to/file.js“></script>

- Maintainability
- Asynchronous loading
- Compiled files

INLINE VS EXTERNAL JS

EXERCISE

File: 007-external-js.hmtl

Let me know if you get stuck, or have any questions!

JQUERY SELECTOR

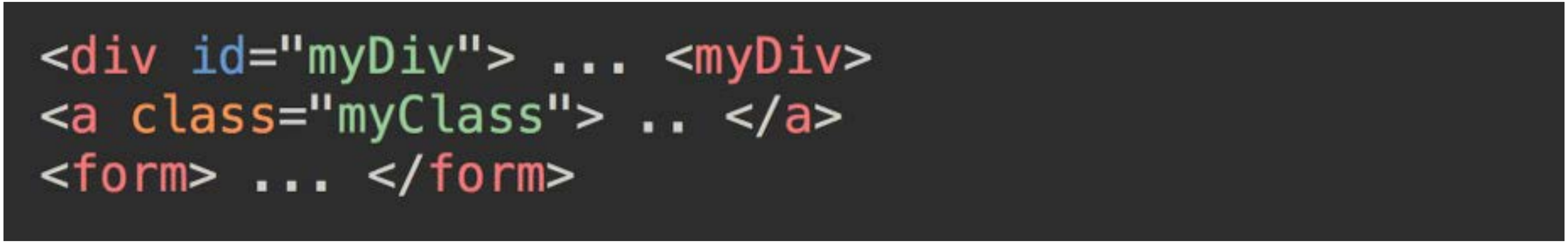
HOW DO WE SELECT ELEMENTS ON THE DOM

A huge problem jQuery solved was how to select elements in the DOM. The first step of manipulating data in the browser, is being able to select that data.

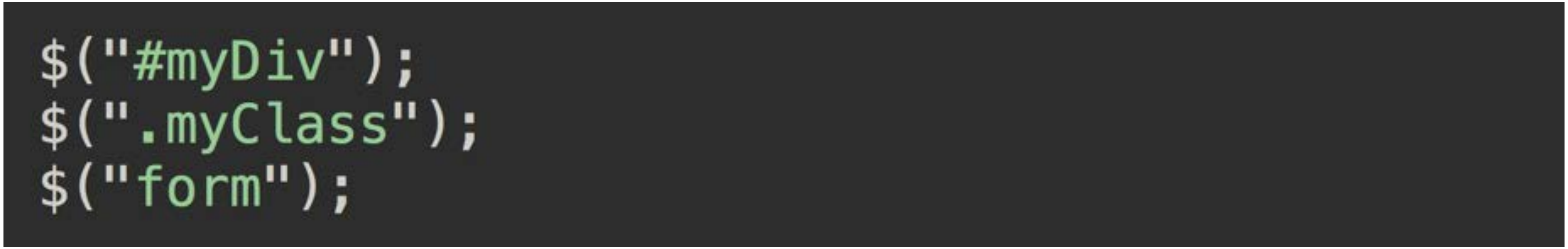
> “The **Document Object Model (DOM)** is a cross-platform and language-independent convention for representing and interacting with objects in HTML, XHTML, and XML documents.”

- Wikipedia

SELECT ELEMENTS BY THEIR CSS SELECTOR



```
<div id="myDiv"> ... <myDiv>
<a class="myClass"> .. </a>
<form> ... </form>
```



```
$("#myDiv");
$(".myClass");
$("form");
```

JQUERY SELECTOR

EXERCISE

File: 008-jq-selector.hmtl

Let me know if you get stuck, or have any questions!

JQUERY DOC READY

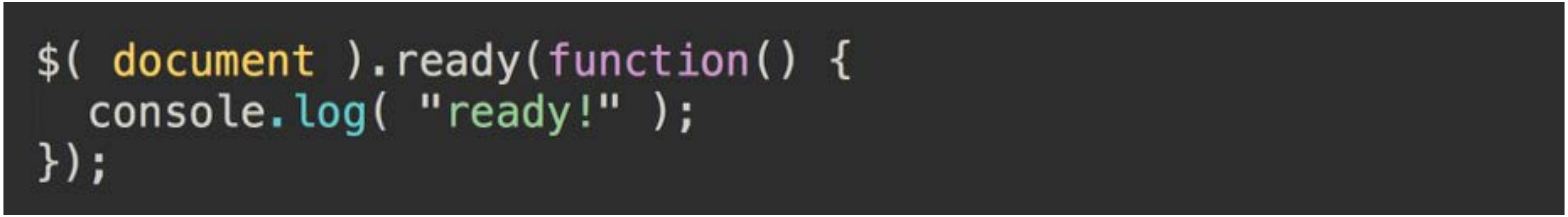
JQUERY DOC READY

IS OUR DOM READY?

The browser loads some content synchronously and other content asynchronously. This can cause a problem with our JavaScript, as a lot of it is dependent on the state of the DOM.

Good News! jQuery has a method to handle this. “A page can't be manipulated safely until the document is ‘ready.' jQuery detects this state of readiness for you.” - jQuery

DOCUMENT READY



```
$( document ).ready(function() {
  console.log( "ready!" );
});
```

The console log statement is fired after the Document has loaded. http://learn.jquery.com/using-jquery-core/document-ready/

EXERCISE

File: 009-jq-doc-ready.hmtl

Let me know if you get stuck, or have any questions!

JQUERY EVENTS

WHAT ARE EVENTS?

Events are actions that happen in the browser. jQuery allows you to bind functions to events that happens in the invent.

Event types:

- ready
- hover
- click

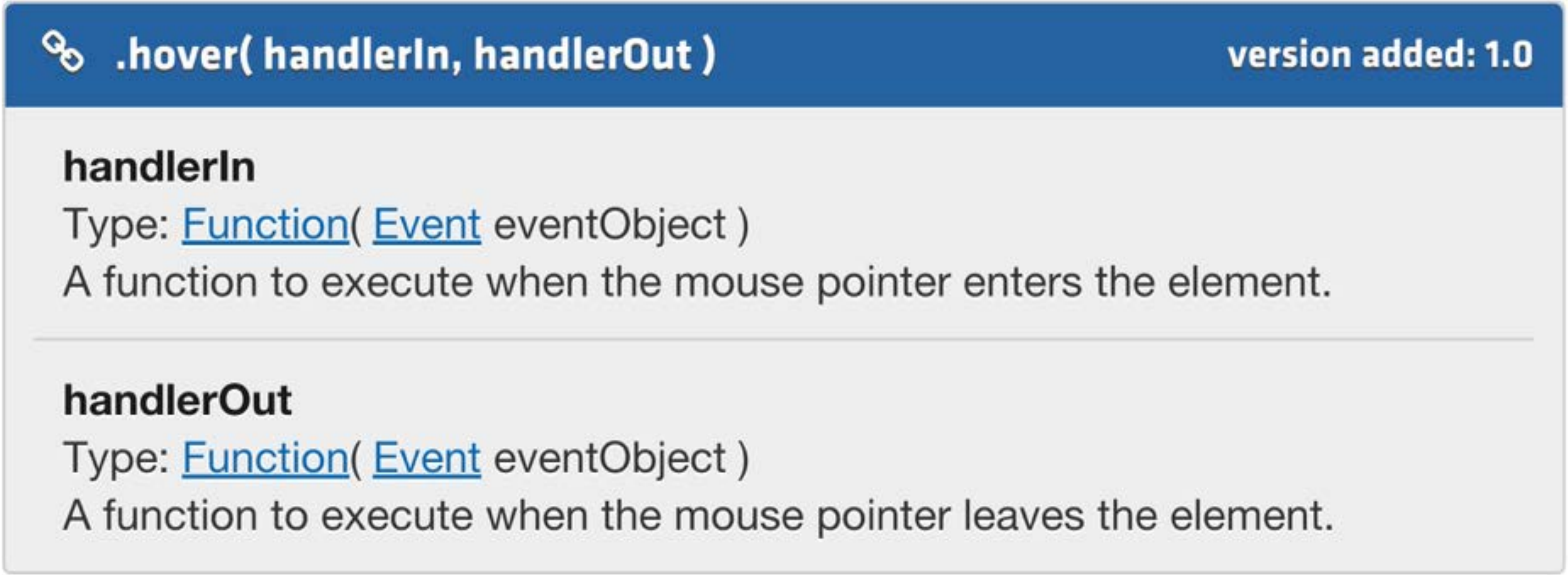
WHY ARE EVENTS IMPORTANT?

Up to this point, we have manually invoked functions. With jQuery events, we can bind functions to actions that occur in the browser.

Examples:

- Once the Document has loaded, preload popular assets to improve latency.
- On hover over button, show tooltip with additional info.
- On click of button, show contact form in modal.

HOVER() EVENT



.hover(handlerIn, handlerOut) version added: 1.0

handlerIn

Type: Function(Event eventObject)

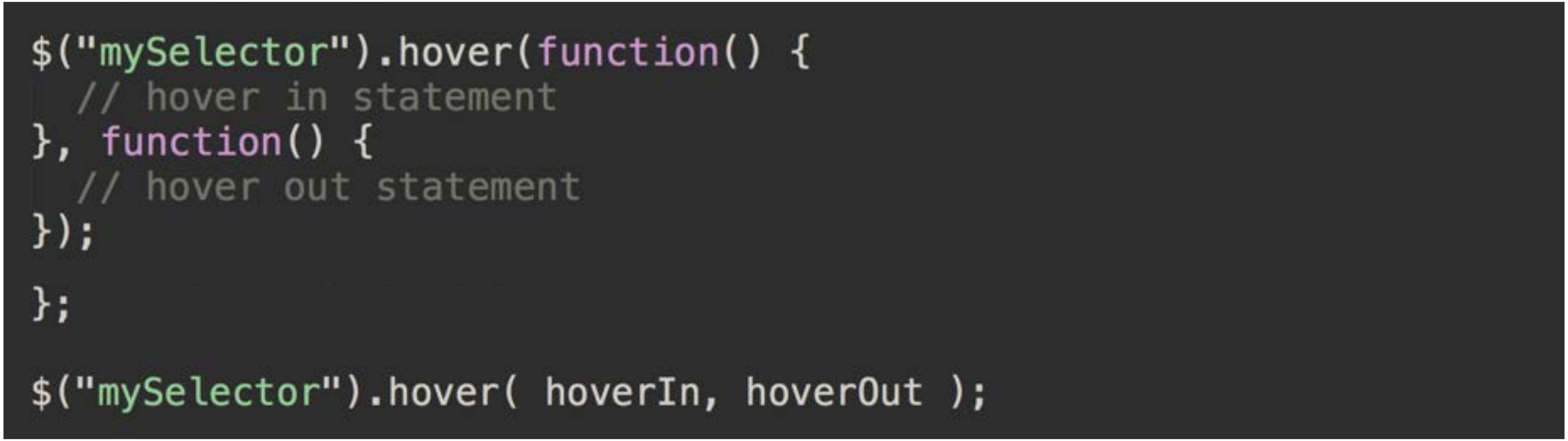
A function to execute when the mouse pointer enters the element.

handlerOut

Type: Function(Event eventObject)

A function to execute when the mouse pointer leaves the element.

HOVER() EVENT



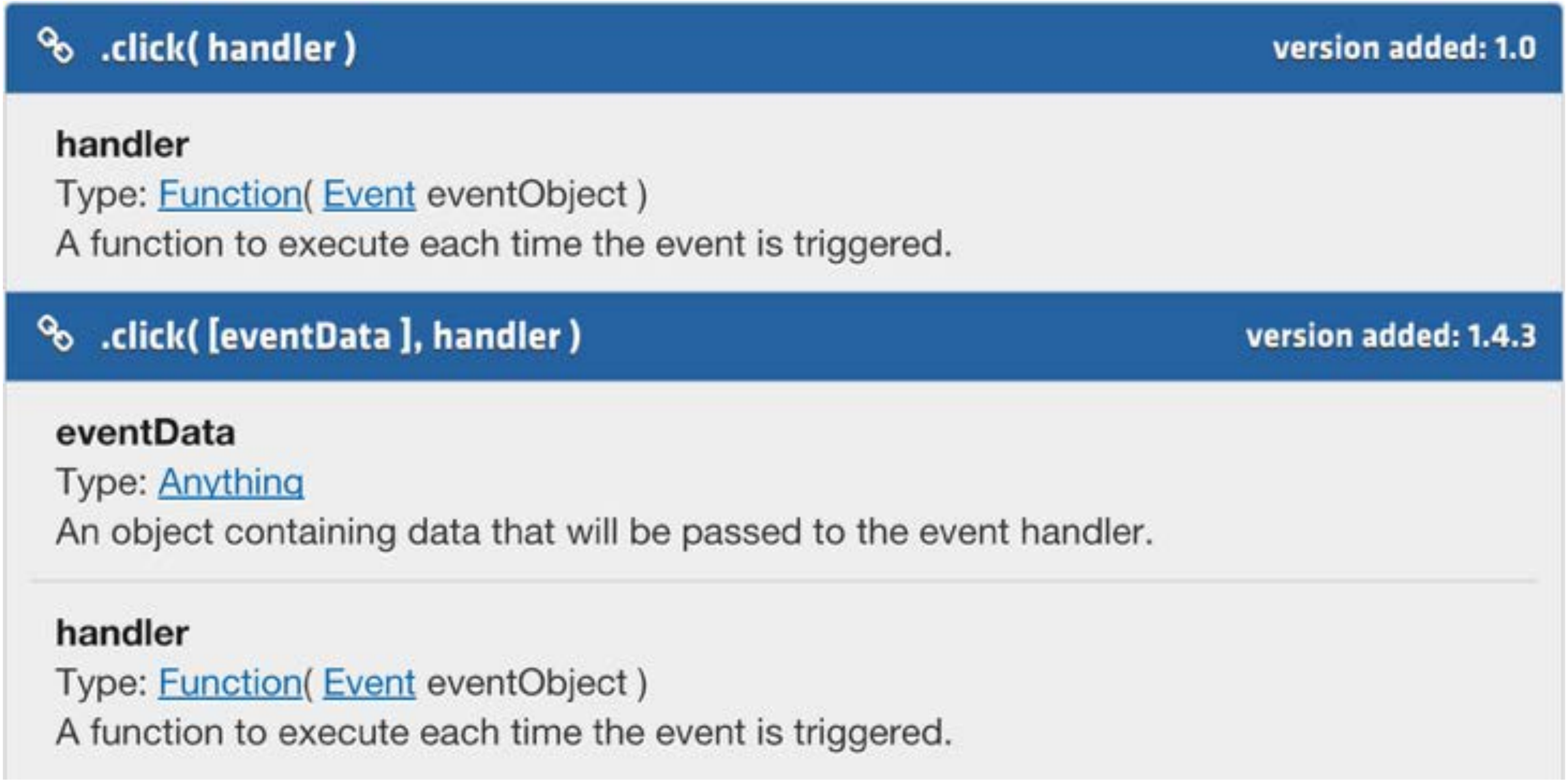
```
$("mySelector").hover(function() {
  // hover in statement
}, function() {
  // hover out statement
});

};

$("mySelector").hover( hoverIn, hoverOut );
```

EXERCISE

File: 010.1-jq-hover.hmtl

Let me know if you get stuck, or have any questions!

CLICK() EVENT



.click(handler) version added: 1.0

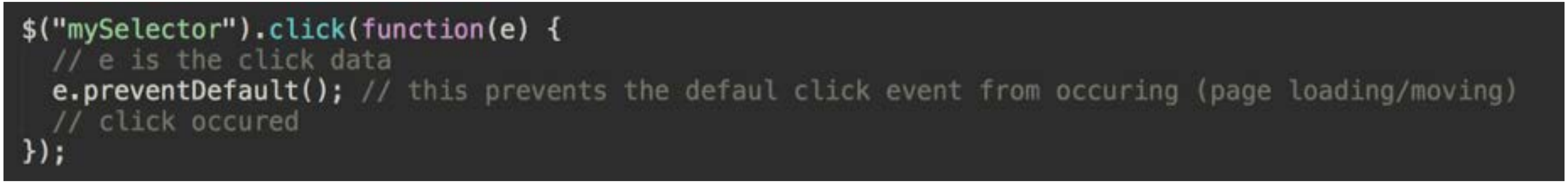
handler
Type: Function(Event eventObject)
A function to execute each time the event is triggered.

.click([eventData], handler) version added: 1.4.3

eventData
Type: Anything
An object containing data that will be passed to the event handler.

handler
Type: Function(Event eventObject)
A function to execute each time the event is triggered.

CLICK() EVENT



```
$("mySelector").click(function(e) {
  // e is the click data
  e.preventDefault(); // this prevents the defaul click event from occuring (page loading/moving)
  // click occured
});
```

EXERCISE

File: 010.2-jq-hover.hmtl

Let me know if you get stuck, or have any questions!

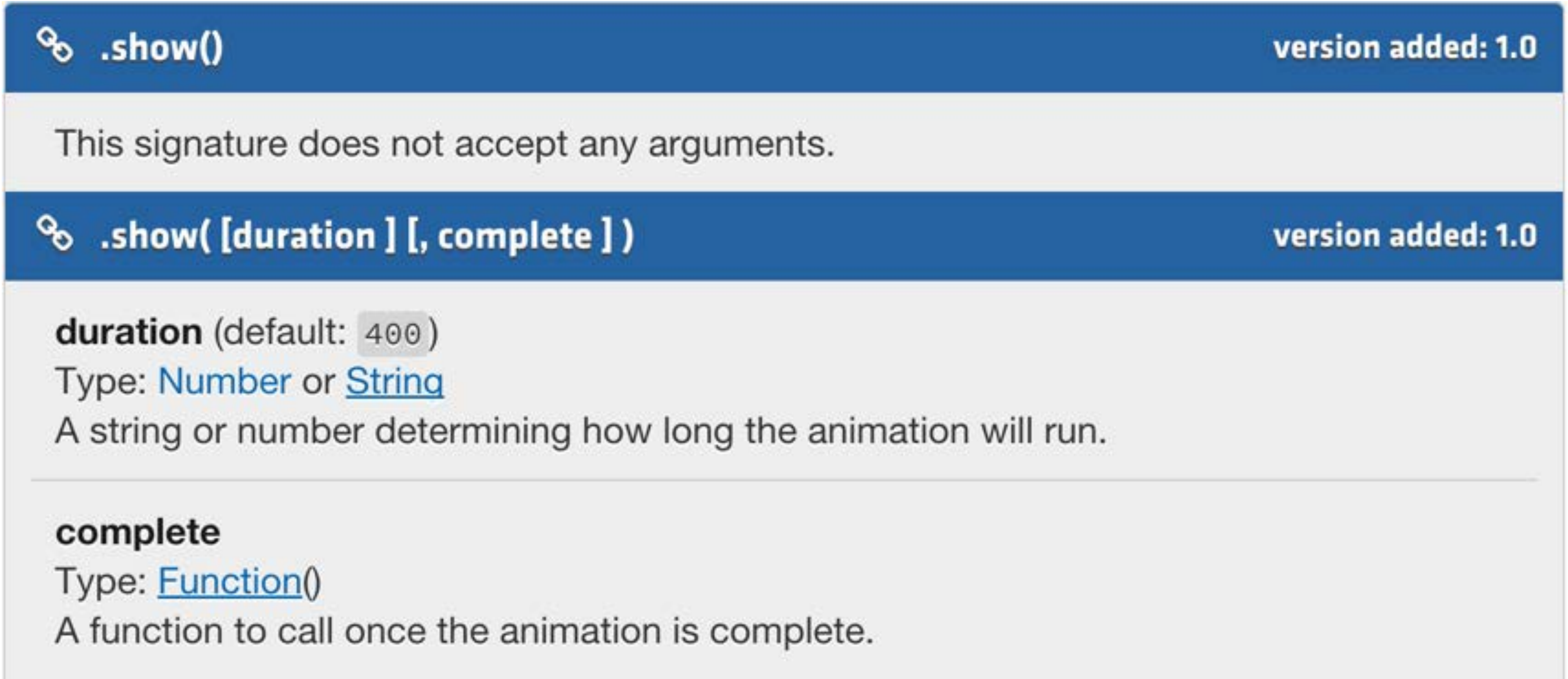
JQUERY ANIMATIONS

JQUERY ANIMATIONS

jQuery animations allow you to add visual effects to the browser with JavaScript. There are a number of jQuery animations, but will go over a commonly used one called show()

JQUERY EVENTS

SHOW()



.show()

version added: 1.0

This signature does not accept any arguments.

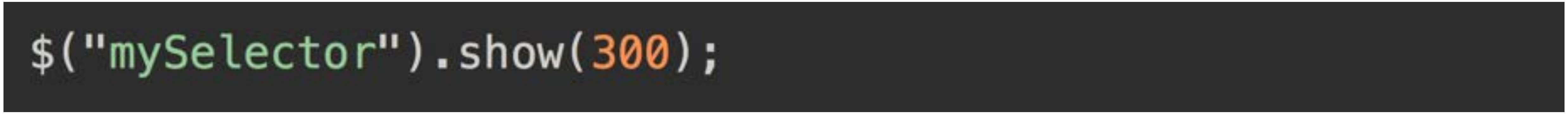
.show([duration] [, complete])

version added: 1.0

duration (default: `400`)
Type: Number or String
A string or number determining how long the animation will run.

complete
Type: Function()
A function to call once the animation is complete.

SHOW() EVENT



```
$("mySelector").show(300);
```

- The selector of what we want to pass in is exactly how we bind events
- The first parameter is the duration of the animation (i.e. how long until the element is showed).
- An option second parameter can passed it. It has to be a function, or callback for when the animation is complete.

Q&A

THANKS!

NAME

- Optional Information: Want more assignments? Email me, as I'll be putting together "homework"
- Email: johnson.evan1@gmail.com
- Twitter: @ev_blurbs