jQuery Selectors

SELECTORS

Selectors in jQuery work in a very similar fashion to CSS selectors, which is great for us. We can select elements by type, id, class, and many more, which we will look at during this lesson.

Click on the ≣ items to expand

ELEMENT SELECTOR +

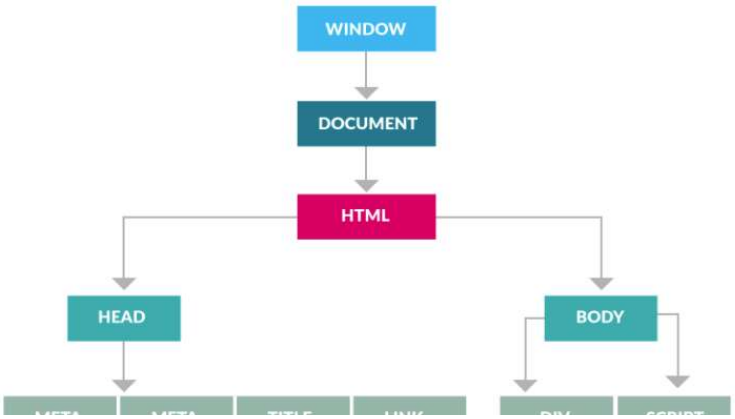
.CLASS SELECTOR +

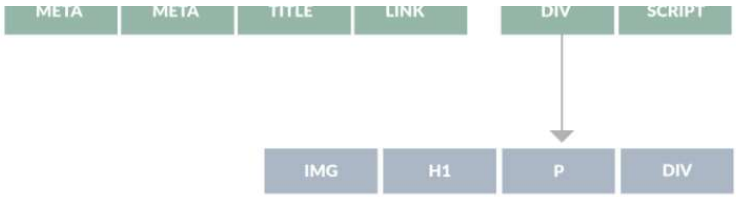
#ID SELECTOR +

THE DOM

One of the most important aspects of JavaScript, and thereby jQuery, is manipulation of the DOM.

DOM stands for Document Object Model and looks like a huge tree, with delicious elements hanging from hard to reach branches, tempting us to find them without getting lost. Getting a little more technical, elements in your DOM are arranged in a hierarchy – which defines what you eventually see in the browser:



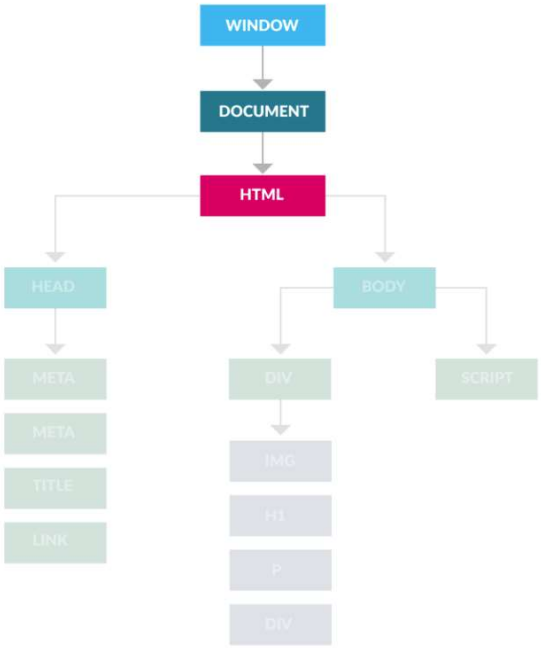


DOM Hierarchy

The DOM hierarchy is used to help organize your HTML elements. In turn, this allows you the ability to navigate and manipulate elements in your HTML documents. It is also used to help your CSS style rules make sense of what styles to apply to which elements. In addition, when discussing the DOM, elements are often referred to as nodes (there are also text nodes and attribute nodes).

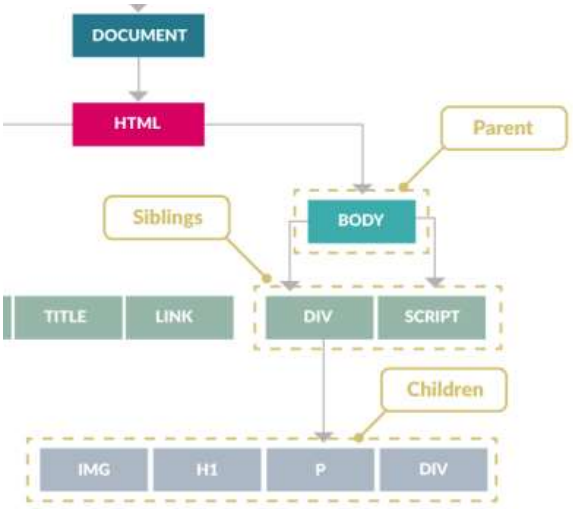
To navigate the DOM, you will have to figure out where in the DOM hierarchy you currently are. You will also need to know where you need to be in order to find, manipulate, or even add required elements. This concept is something that will become more apparent as we continue the lesson and put some of this theory into practice.

Before we can find elements and work our magic on them, we need to first know where the elements are. Let us start at the top of the DOM tree hierarchy and climb all the way down. The view from the top of your DOM is made up of your window, document, and html elements, as shown below:



DOM Hierarchy

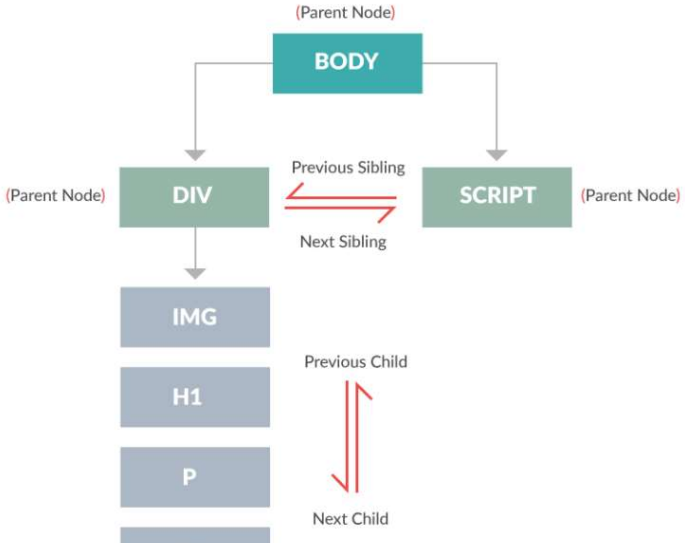
Below the HTML element level, the DOM starts to branch out and get more interesting. To help you navigate the DOM, it is important to remember that all of the elements in the DOM are related in some way. This includes parents, siblings, or children and descendants. To better explain the relationships between DOM elements, take a look at the row containing the div and script elements in the following diagram:



In the above example, both the div and script elements are siblings. The reason they are siblings is because they share the parent body element. The script element has no children, but the div element does. The img, h1, p, and div are children of the div element. These div elements are also siblings, as they have the same div parent. We can also say that the img, h1, and p elements are descendents of the body element (a bit like grandchildren).

DEALING WITH SIBLINGS AND PARENTS

The following diagram gives you an idea of how the relationship between the three properties of parent, child, and sibling work:



(http://codeinstitute.wpengine.com/)

Daniel Hoksza (http://codeinstitute.wpengine.com/members/danielhoksza/)

Of the three properties mentioned above, the easiest ones to deal with are the parents and siblings. Let's see how jQuery deals with

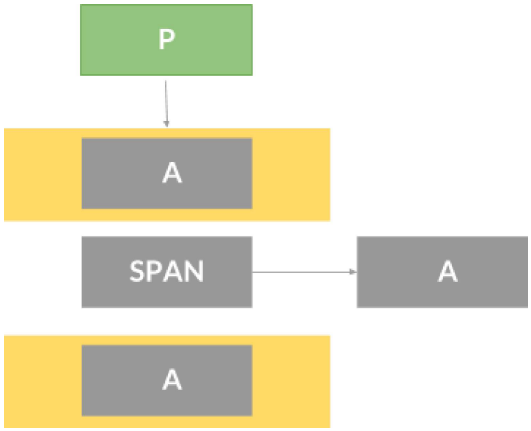
parents and children.

PARENT/CHILD RELATION SELECTORS

jQuery allows you to traverse the DOM tree and select elements based on their parent element. There are two variations. The first variation will only match elements which are a direct child to parent elements, and the second variation will match all descendants of a family, e.g. grandchildren and everything below.

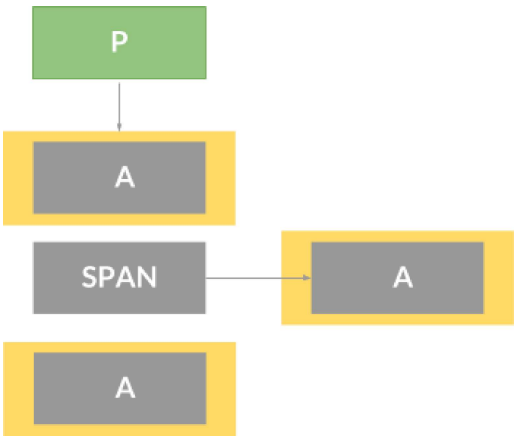
In the below example, we want the selector to return children who are direct descendants of an element. The selector will return all anchor link (a) children that are direct descendants of a paragraph element (p):

```
$("p>a");  // returns anchors that are direct children of paragraphs
```



For the selector to return all descendants of a parent, regardless of whether they are directly related or not, we can use the following syntax:

```
$("p a");  // returns all paragraph elements descended from div
```

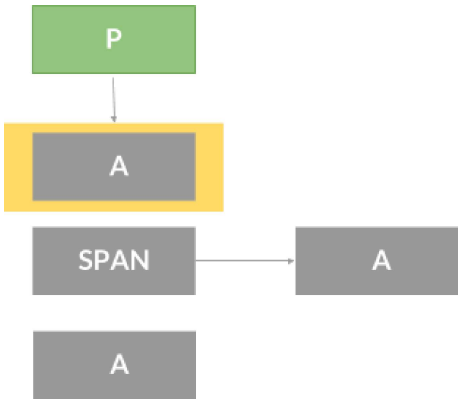


TOP

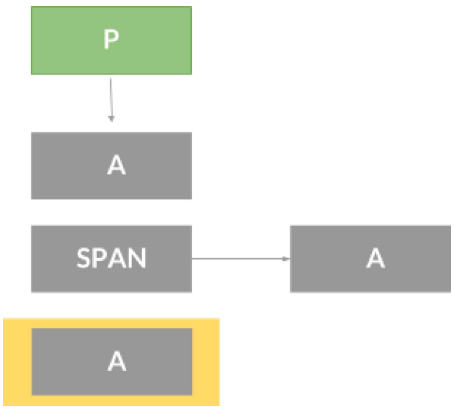
PSEUDO SELECTORS

If you remember from previous lessons, pseudo selectors are CSS selectors with a colon (:) preceding them. Similar to CSS, jQuery also has pseudo selectors which can come in handy. Below you can see a few of these pseudo selectors:

```
$("a:first");  // selects the first anchor on the page
$("a:last");  // selects the last anchor on the page
$(":header");  // selects header elements (h1, h2, h3, etc)
```



a:first



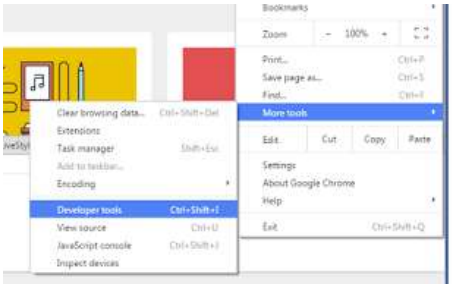
a:last

There are many other types of jQuery selectors you can use, and these are listed at the end of the lesson.

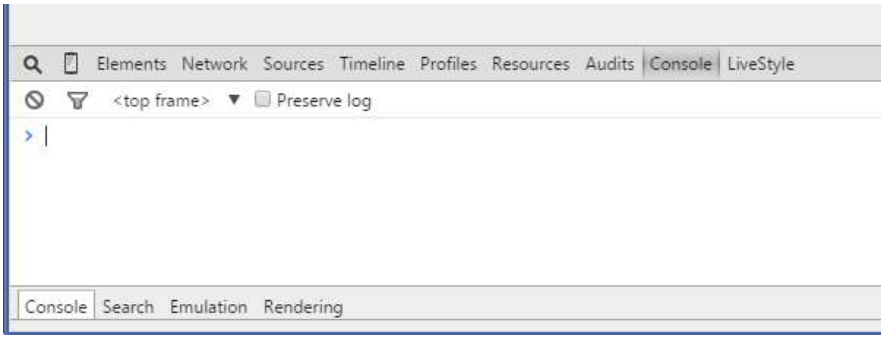
A WALK THROUGH: USING CHROME DEVELOPMENT TOOLS

jQuery is usually written in a separate file to the HTML and then linked, like an external CSS file. However, you can also use Chrome dev tools to write jQuery statements right in the browser. Give this a try yourself:

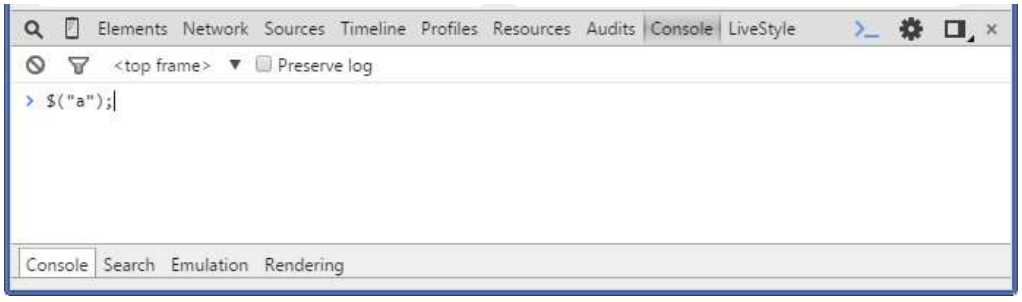
- Let's begin by opening Sample Site in the lesson folder.
- Take 6 minutes to familiarise yourself with the code.
- Open up Chrome and development tools as shown below:



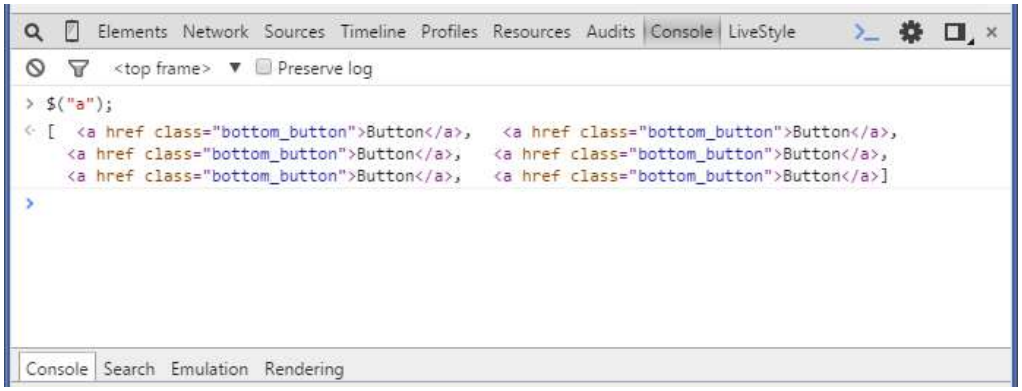
- Click console on the Developer tools menu, as shown below:



- Now we can type jQuery statements directly into the browser and hit Enter to see the result. For example, return all link (<a>) elements:



- jQuery returns all the <a> tags on the page, similar to what is shown below:



CHALLENGE

Click on the tabs to reveal Challenge and Hint.

CHALLENGE HINT

Let's use this technique to select the following:

- All images – element
- card_image – class
- footer – id
- All paragraph tags within the footer
- All header elements
- The first element on the page
- The last div on the page
- The last image on the page
- All elements with an alt attribute

USEFUL RESOURCES

DISCUSSION

ASK QUESTION

PREVIOUS UNIT

NEXT UNIT

TIME REMAINING : 10 WEEKS

100%

STREAM ONE - INTRODUCTION

STREAM ONE—INTRODUCTION

BEFORE YOU START

MEET YOUR MENTOR

REOPENING THE BOX MODEL

- REOPENING THE BOX MODEL – INTRO
- HTML REVISION
- CODING IN AN EDITOR
- THE BOX MODEL
- BORDER RADIUS
- REOPENING THE BOX MODEL QUIZ

REOPENING THE BOX MODEL 2

- MORE BOX MODEL
- DISPLAY PROPERTIES
- SHORTHAND
- THE VALIDATOR
- THE BOX MODEL CONTINUED QUIZ

POSITIONING & CSS

- POSITIONING & CSS – INTRO
- CSS3 BORDER BOX
- POSITIONING & FLOW
- ABSOLUTE POSITIONING
- DEVELOPER TOOLS
- FIXED POSITIONING
- THE CASCADE
- POSITIONING QUIZ

MEDIA

- MEDIA – INTRO
- INTRODUCTION TO <IMG>
- BACKGROUND IMAGE
- CSS FILTERS
- HTML VIDEO
- HTML AUDIO
- MEDIA QUIZ

GRADIENTS, TRANSFORMS & TRANSITIONS

- GRADIENTS, TRANSFORMS & TRANSITIONS – INTRO
- GRADIENTS
- TRANSFORMS
- TRANSITIONS
- ANIMATION WITH CSS
- TRANSITIONS QUIZ

NAVBAR - CSS SELECTORS

- NAVBAR – CSS SELECTORS – INTRO
- MOBILE FIRST
- BUILDING SITE NAVIGATION
- CSS SELECTORS
- PSEUDO-CLASSES (:)
- PSEUDO-ELEMENTS(::)
- TABLE RELATED ELEMENTS AND SELECTORS
- RESPONSIVE QUIZ

FLEXBOX

- FLEXBOX – INTRO
- GETTING STARTED WITH FLEXBOX
- THIS WAY UP
- ALIGNMENT
- CALL TO ORDER
- FLEXBOX QUIZ

SEMANTIC WEB & FONT

- SEMANTIC WEB & FONTS – INTRO
- SEMANTIC WEB
- FONTS
- TEXT ATTRIBUTES
- MEASUREMENTS
- SEMANTIC QUIZ

BALSAMIQ

- BALSAMIQ – INTRO
- THE BASICS
- ADDING CONTROLS
- IMAGE PAPER CONTROLS
- MOCKUP COMPLETION
- THE FINISHING TOUCHES

USER EXPERIENCE DESIGN

- USER EXPERIENCE DESIGN – INTRO
- USER EXPERIENCE (UX)
- USERS
- USABILITY
- UX QUIZ

FORMS

- FORMS – INTRO
- WHAT IS A FORM?
- INPUT TYPES
- OTHER FORM TAGS
- EXPLORING INPUT ATTRIBUTES
- NUMBERS & PATTERNS
- MAKING CHOICES
- FORMS QUIZ

GIT

- GIT – INTRO
- WHAT IS GITHUB?
- THE COMMAND LINE – AN INTERLUDE
- SET UP GITHUB

GIT REMOTE

- GIT REMOTE – INTRO
- CONNECTING TO GITHUB
- WORKING ON SOMEONE ELSE'S PROJECT

- GIT – QUIZ

BOOTSTRAP

- BOOTSTRAP – INTRO
- WHAT IS BOOTSTRAP?
- THE GRID SYSTEM
- BOOTSTRAP COMPONENTS
- BOOTSTRAP 1- QUIZ

BOOTSTRAP NAV ELEMENTS

- BOOTSTRAP NAV ELEMENTS – INTRO
- NAVIGATION
- TABS & PILLS
- COMPONENTS
- ACCORDION PANELS
- BOOTSTRAP 2 – QUIZ

BOOTSTRAP RESPONSIVE DESIGN

- BOOTSTRAP RESPONSIVE DESIGN – INTRO
- RESPONSIVE BOOTSTRAP
- RESPONSIVE NAVBAR
- RESPONSIVE GRID SYSTEM
- OVERRIDING BOOTSTRAP
- BOOTSTRAP 3 – QUIZ

JAVASCRIPT REVISITED

- JAVASCRIPT REVISITED – INTRO
- JAVASCRIPT DATA STRUCTURES & CONTROLS
- JAVASCRIPT FUNCTIONS
- JAVASCRIPT OBJECTS

JQUERY

- JQUERY – INTRO
- DOWNLOADING JQUERY
- JQUERY SELECTORS
- CHANGING HTML/CSS WITH JQUERY
- WRITING OUR FIRST SCRIPT
- JQUERY 1 – QUIZ

JQUERY EVENTS

- JQUERY EVENTS—INTRO
- EVENTS IN JQUERY
- JQUERY EFFECTS
- METHOD CHAINING
- THE IMPORTANCE OF *THIS*

JQUERY: TRAVERSING MT. DOM

- JQUERY: TRAVERSING MT. DOM – INTRO
- TRAVERSING UP AND DOWN THE DOM TREE
- TRAVERSING SIDEWAYS
- OTHER TRAVERSING METHODS

JAVASCRIPT PART II

- JAVASCRIPT PART II – INTRO
- JAVASCRIPT IN THE BROWSER
- JAVASCRIPT & EXTERNAL COMMUNICATION

DATA VISUALISATION & D3

- DATA VISUALISATION & D3 – INTRO
- WHAT IS D3?
- SCALABLE VECTOR GRAPHICS
- D3 SETUP
- D3 SELECTORS

D3 & DATA

- D3 & DATA – INTRO
- USING DATA
- DRAWING A BAR CHART
- FITTING A CHART INSIDE A SVG

D3 SCALES

- D3 SCALES – INTRO
- D3 SCALES
- COLOUR SCALES
- TOOLTIPS

D3 AXES & EXTERNAL FILES

- D3 AXES & EXTERNAL FILES – INTRO
- AXES
- EXTERNAL FILES

INTRODUCTION TO ANGULARJS

- ANGULARJS – INTRO
- FRAMEWORK VS LIBRARY
- THE MVC/MVVM DESIGN PATTERNS
- ANGULARJS APPLICATION STRUCTURE
- CREATING AN ANGULAR APPLICATION
- $SCOPE AND BUILT-IN DIRECTIVES
- ANGULAR FILTERS

ANGULAR ROUTING

- ANGULAR ROUTING – INTRO
- SINGLE PAGE APPLICATIONS
- SETTING UP THE ROUTING CONFIG.
- INTRODUCTION TO GIT
- SETTING UP NPM
- ADDING A PARAMETER TO A ROUTE

ANGULAR FORMS

- ANGULAR FORMS – INTRO
- ANGULAR FORM DIRECTIVES & SERVICES
- FORM VALIDATION OUTSIDE THE FORM

- FORM SELECTION MENU

ANGULAR SERVICES

- ANGULAR SERVICES – INTRO
- THE ANGULAR SERVICE MODULE
- THE ANGULAR $HTTP SERVICE
- $HTTP SERVICE IN ACTION
- THE MOVIEDB API

ANGULAR DIRECTIVES

- ANGULAR DIRECTIVES – INTRO
- ANATOMY OF A CUSTOM DIRECTIVE
- MOVIES DB REVISITED
- CREATING A NAVIGATION MENU
- CREATING A GOOGLE MAP DIRECTIVE
- PULLING IT ALL TOGETHER

PROJECT FOR STREAM 1

- FINAL PROJECT FOR STREAM ONE

BACK TO COURSE (http://codeinstitute.wpengine.com/course/front-end-development/)

Sample Course

(0 REVIEWS)

65 STUDENTS

(http://codeinstitute.wpengine.com/course/full-stack-development-sample-units/)

Back End Development

(0 REVIEWS)

6 STUDENTS

(http://codeinstitute.wpengine.com/course/back-end-development/)

Front End Development

(0 REVIEWS)

41 STUDENTS

(http://codeinstitute.wpengine.com/course/front-end-development/)

WHO'S ONLINE

RECENTLY ACTIVE MEMBERS

(http://co tu ng m be ie a/)

(http://co

mansfield/)

(http://codeinstitute.wpengine.com/m

CODE INSTITUTE 2015

(HTTP://CODEINSTITUTE.WPENGINE.COM/)