Udacity - jQuery -

$ → jquery collection
    $(string) $.ajax()
    $(function)
    $(DOM Element)

<script src = "https://ajax.googleapis.com/ajax/libs/
             jquery/3.3.1/jquery.min.js"> </script>

$('tag')   *not very useful
$('.class')   ⎫ selectors
$('#id')   ⎭

.parent() ¹ˡᵉᵛᵉˡ   .children() ¹ˡᵉᵛᵉˡ   .siblings()
.parents() ᵐᵃⁿʸ   .find() ᵐᵃⁿʸ
                '*' - all; universal selector

navList = $('.navList');
firstItem = navList.children().first();
link = firstItem.find('a');
link.attr('href', '#1');

.html() - shows all html of an element
.text() - only returns text, without html tags

$('#input').on('change', function() {
    var val = $(this).val();
    $(this).next().text(val);
});

Udacity - jQuery

jQuery -
```
var articleItems = $('.article-item');
var ul = articleItems.find('ul');
ul.remove();
```

first ArticleItem = $('.article-item').first

.append() *last child

firstArticleItem.append('<img ... >');

.prepend() *first child

.prepend

.insertBefore() *sibling before

.insertAfter() *sibling after

```
var family1 = $('#family1');
var family2 = $('<div id="family2"><h1>Family2</h1></div>');
var bruce = $('<div id="bruce"><h2>Bruce</h2></div>');
var madison =        "madison"  h3>Madison</h3
var hunter =         "hunter"      >Hunter<
family2.insertAfter(family1);
family2.append(bruce);
bruce.append(madison);
bruce.append(hunter);
```

.each

monitorEvents(elementToWatch); → console on Chrome

1. target element to listen to: $('#input').on('keypress',
2. event we want to react to   function() {
3. actions to take in response

```
$('#my-button').on('click', function() {
    $(this).remove();
    $('body').addClass('success');
```

jauery-GWG (cont.)-

e/ evt/ event → event object containing info
about the event
evt. target - holds page element that is target of event
evt. preventDefault();
evt. keyCode - what key was pressed
evt. pageX - (pageY) where on page click occurred
evt. type - what event happened

event delegation -
$('.container').on('click', 'article', function() {});
↑check if event target is article

→ - set size of table as an N by M grid (makeGrid())
- choose a color
- click a cell fill in color

- submit button
- table
- color picker
- value of color clicked

- lowercase
X whitespace
- 2 space between rules
.. X type = text/css
" X ' HTML
' X " javascript

~~* style pixel maker /± random color~~
~~* add working pixel maker webpage to repository~~
~~* add project to GWG, my website~~
~~* style animal trading card~~
~~* add project to GWG, my website~~
* add brief of GWG experience to website
* share projects on forum & slack