Class Project

Auto-generate a TOC

- Start with **AutoTOC.html**
- Look at existing HTML and JS
- Inside the JS function, create a UL to hold the list of links to named anchors (which will also have to be created)

Logic map:

1. Create new UL HTML element to hold our eventual bookmarks
2. Specify which existing HTML tag to use as the bookmark/anchor
3. Grab all the existing HTML tags to use as the bookmark/anchor, then for each one…
4. …give it a unique name (shoot – we need a counter for this) and text
5. Create new LIs to use as the bookmark links
6. Append each new LI bookmark link to the UL container
7. Add the container to the page
8. Oh – do all this in an abstracted function so we reuse it or call it when needed

Oh – do all this in an abstracted function so we reuse it or call it when needed

```
function buildBookmarks(whichTag, bookmarkContainer) {


}
```

Create new HTML element to hold our eventual bookmarks

```
function buildBookmarks(whichTag, bookmarkContainer) {
var ul = $('<ul id="bookmarksList" >');
}
```

Specify which existing HTML tag to use as the bookmark/anchor

```
function buildBookmarks(whichTag, bookmarkContainer) {
 var ul = $('<ul id="bookmarksList" >');
}
```

Grab all the existing HTML tags to use as the bookmark/anchor, then for each one…

```
function buildBookmarks(whichTag, bookmarkContainer) {
 var ul = $('<ul id="bookmarksList" >');
 $('#content h3')
}
```

But we need to abstract this and use the passed whichTag

```
function buildBookmarks(whichTag, bookmarkContainer) {
 var ul = $('<ul id="bookmarksList" >');
 $('#content ' + whichTag) //maintain the space after #content
}
```

…give it a unique name (shoot – we need a counter for this) and text

```
function buildBookmarks(whichTag, bookmarkContainer) {
 var anchorCount = 0;
 var ul = $('<ul id="bookmarksList" >');
 $('#content ' + whichTag)
}
```

Give each a unique name and unique text

```
function buildBookmarks(whichTag, bookmarkContainer) {
 var anchorCount = 0;
 var ul = $('<ul id="bookmarksList" >');
 $('#content ' + whichTag).each();
}
```

```
function buildBookmarks(whichTag, bookmarkContainer) {
 var anchorCount = 0;
 var ul = $('<ul id="bookmarksList" >');
 $('#content ' + whichTag).each(function(){}); //create space inside function’s curly braces
}
```

```
function buildBookmarks(whichTag, bookmarkContainer) {
 var anchorCount = 0;
 var ul = $('<ul id="bookmarksList" >');
 $('#content ' + whichTag).each(function(){
  $(this) //the current whichTag
 });
}
```

```
function buildBookmarks(whichTag, bookmarkContainer) {
 var anchorCount = 0;
 var ul = $('<ul id="bookmarksList" >');
 $('#content ' + whichTag).each(function(){
  $(this).html(); //change the HTML for each whichTag (H3) to make it a named anchor
 });
}
```

We want to turn something like this:
`<h3>Introduction</h3>`

Into this:
`<h3><a id="bookmark0"></a>Introduction</h3>`

Start with hard coding, then abstract it

```
function buildBookmarks(whichTag, bookmarkContainer) {
 var anchorCount = 0;
 var ul = $('<ul id="bookmarksList" >');
 $('#content ' + whichTag).each(function(){
  $(this).html('<a id="bookmark0"></a>Introduction');
 });
}
```

To abstract it:

1. Select the number
2. Type `' + anchorCount + '`

```
function buildBookmarks(whichTag, bookmarkContainer) {
 var anchorCount = 0;
 var ul = $('<ul id="bookmarksList" >');
 $('#content ' + whichTag).each(function(){
  $(this).html('<a id="bookmark' + anchorCount + '"></a>Introduction');
 });
}
```

We need to abstract the original HTML text, so tag it on to the end

1. Select `Introduction'`
2. Type `' + $(this).html()`
   Remember $(this) means 'This object we are working with'. In this case, it is the current H3. So – we are getting the current H3s existing HTML content.

```
function buildBookmarks(whichTag, bookmarkContainer) {
 var anchorCount = 0;
 var ul = $('<ul id="bookmarksList" >');
 $('#content ' + whichTag).each(function(){
  $(this).html('<a id="bookmark' + anchorCount + '"></a>' + $(this).html());
 });
}
```

That takes care of the H3s. Now we need to create the list of links.

Create new LIs to use as the bookmark links

Ultimately, we need this:
`<li><a href="#bookmark0">Introduction</a></li>`

Start with hard coding, then abstract it

```
function buildBookmarks(whichTag, bookmarkContainer) {
 var anchorCount = 0;
 var ul = $('<ul id="bookmarksList" >');
 $('#content ' + whichTag).each(function(){
  $(this).html('<a id="bookmark' + anchorCount + '"></a>' + $(this).html());
  var li = $('<li><a href="#bookmark0">Introduction</a></li>'); // need to abstract this
 });
}
```

First, abstract the # in the HREF

1. Select the number
2. Type `' + anchorCount + '` to replace it

```
function buildBookmarks(whichTag, bookmarkContainer) {
 var anchorCount = 0;
 var ul = $('<ul id="bookmarksList" >');
 $('#content ' + whichTag).each(function(){
  $(this).html('<a id="bookmark' + anchorCount + '"></a>' + $(this).html());
  var li = $('<li><a href="#bookmark' + anchorCount + '">Introduction</a></li>');
 });
}
```

Next, abstract the text `Introduction`

1. Select the text `Introduction`
2. Type `' + $(this).text() + '` to replace it
   Remember $(this) means 'This current object we are working with'. In this case, it is the current H3. So – we are getting the current H3s existing text.

```
function buildBookmarks(whichTag, bookmarkContainer) {
 var anchorCount = 0;
 var ul = $('<ul id="bookmarksList" >');
 $('#content ' + whichTag).each(function(){
  $(this).html('<a id="bookmark' + anchorCount + '"></a>' + $(this).html());
  var li = $('<li><a href="bookmark' + anchorCount + '">' + $(this).text() + '</a></li>');
 });
}
```

Append each new LI bookmark link to the UL container and increment the anchorCount to track the next one

```
function buildBookmarks(whichTag, bookmarkContainer) {
 var anchorCount = 0;
 var ul = $('<ul id="bookmarksList" >');
 $('#content ' + whichTag).each(function(){
  $(this).html('<a id="bookmark' + anchorCount + '"></a>' + $(this).html());
  var li = $('<li><a href="#bookmark' + anchorCount + '">' + $(this).text() + '</a></li>');
  ul.append(li);
  anchorCount++;
 });
}
```

Finally, outside the EACH loop, append the UL to the page

```
function buildBookmarks(whichTag, bookmarkContainer) {
 var anchorCount = 0;
 var ul = $('<ul id="bookmarksList" >');
 $('#content ' + whichTag).each(function(){
  $(this).html('<a id="bookmark' + anchorCount + '"></a>' + $(this).html());
  var li = $('<li><a href="#bookmark' + anchorCount + '">' + $(this).text() + '</a></li>');
  ul.append(li);
  anchorCount++;
 });
 $('#header').append(ul); //but it needs to be abstracted
}
```

So, abstract it:

1. Select `header'`
2. Type `' + bookmarkContainer`

```
function buildBookmarks(whichTag, bookmarkContainer) {
 var anchorCount = 0;
 var ul = $('<ul id="bookmarksList" >');
 $('#content ' + whichTag).each(function(){
  $(this).html('<a id="bookmark' + anchorCount + '"></a>' + $(this).html());
  var li = $('<li><a href="#bookmark' + anchorCount + '">' + $(this).text() + '</a></li>');
  ul.append(li);
  anchorCount++;
 });
 $('#' + bookmarkContainer).append(ul);
}
```

Final function code:

```
function buildBookmarks(whichTag, bookmarkContainer) {
 var anchorCount = 0;
 var ul = $('<ul id="bookmarksList" >');
 $('#content ' + whichTag).each(function(){
  $(this).html('<a id="bookmark' + anchorCount + '"></a>' + $(this).html());
  var li = $('<li><a href="#bookmark' + anchorCount + '">' + $(this).text() + '</a></li>');
  ul.append(li);
  anchorCount++;
 });
 $('#' + bookmarkContainer).append(ul);
}
```

Remember to actually call the function!

```
$('document').ready(function() {
       buildBookmarks('h3', 'header');
});
```

On your own

Using any of the demo files, see if you can:

- Alert the HTML of an element
- Alert the text of an element
- Set new HTML for an element
- Set new text for an element
- Alert an attribute's value
- Set a single attribute
- Set multiple attributes at once
- Remove an attribute
- Append/prepend content to an existing element
- Move an element so that it is appended/prepended to another element
- Insert new content before/after existing content (differentiate between append/prepend and before/after)
- Wrap existing elements inside of new tags
- Wrap a group of existing elements inside a single new tag
- Replace a tag with a different tag
- Empty/remove an element
- Alert a CSS property
- Set single CSS property
- Set multiple CSS properties at once
- Add a class
- Test for the existence of a class
- Remove a class
- Alert the CSS sizing and positioning of an element