Making a Bookmarklet

Credit to Casey Watts for the tutorial.

A bookmark normally takes you to a new web page.

A bookmarklet is a bookmark that runs javascript on the current page instead of taking you to a new page. To declare that it is a bookmarklet, the "location" it points to starts with `javascript:`.

Today, we'll make bookmarklets as a kind of warm-up for the extension project.

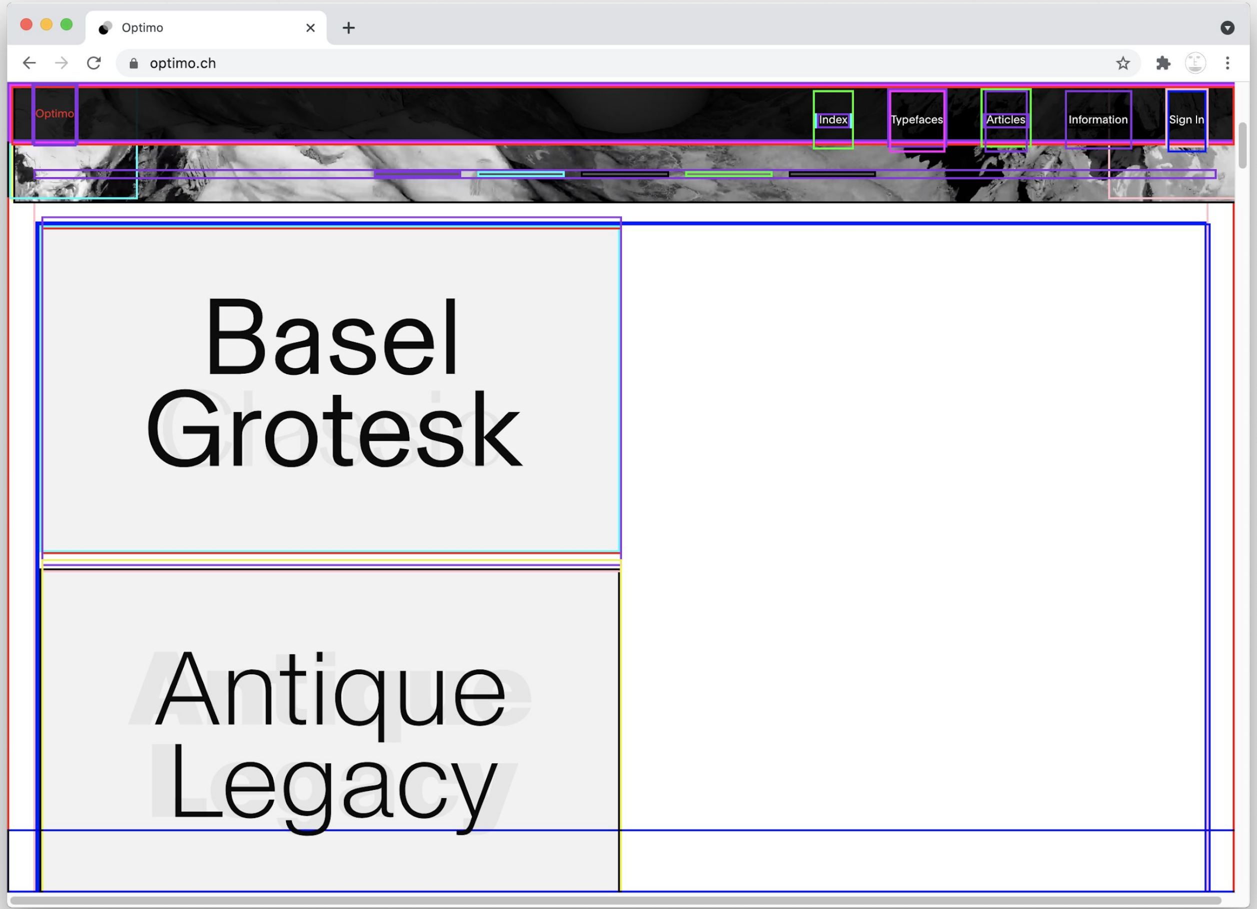
I'll go over two things:
–some JavaScript techniques you might want to use
–how to make a bookmarklet

Example–Boxify

Puts a border around every HTML element, so you can see the box model.

Open your bookmark manager.
Add a new bookmark. Type "Boxify" as the name.
Paste this code into the URL field.
Visit a website and then click on the Boxify bookmark.

```
javascript:(function()%7Blet%20all%3Ddocument.querySelectorAll(%60*%
60)%3Bfor(let%20i%3D0%3Bi%3Call.length%3Bi%2B%3D1)%7Blet%20colors%3D
%5B%60red%60,%60blue%60,%60lime%60,%60pink%60,%60yellow%60,%60black%
60,%60cyan%60,%60magenta%60,%60blueviolet%60%5D%3Blet%20randomColor%
3DMath.floor(Math.random()*colors.length)%3Blet%20chosenColor%3Dcolo
rs%5BrandomColor%5D%3Ball%5Bi%5D.style.border%3D%602px%20solid%20%24
%7B%20chosenColor%20%7D%60%7D%7D)()%3B
```



Optimo
optimo.ch
Optimo
Index
Typefaces
Articles
Information
Sign In
Basel Grotesk
Antique Legacy

More examples

https://www.squarefree.com/bookmarklets/color.html

Targeting HTML elements

How to target HTML elements

Finding HTML elements by id

```
var myElement = document.getElementById("intro");
```

Finding HTML elements by tag name

```
var x = document.getElementsByTagName("p");
```

Finding HTML elements by class name

```
var x = document.getElementsByClassName("intro");
```

Finding HTML elements by CSS selectors

```
var x = document.querySelectorAll("p.intro");
```

How to target HTML elements

If you've targeted more than a single element, this means you'll have an array.
In order to perform something on the elements returned, you'll need to loop through the array.

```
// store the elements in a variable. This creates an array.
var elements = document.getElementsByTagName("img");

// iterate through the array, doing something to each element.
for (var x = 0; x < elements.length; x++) {
    console.log(elements[x]);
  };
```

Use JS to change HTML

innerHTML

innerHTML lets you get or rewrite the contents of an HTML element.

Get the contents of the body, and store it in a variable:

```
var content = document.body.innerHTML;
```

Rewrite the contents of an element:

```
document.getElementById("demo").innerHTML = "Hello Dolly.";

document.body.innerHTML = "Some new HTML content";
```

document.write

One way to output information with Javascript is to use write. That let's you write something to the HTML page. This will delete all content that has already been loaded.

```
document.write("Hello World!");

document.write("<h1>Hello World!</h1><p>Have a nice
day!</p>");
```

replace

replace lets you substitute one string for another.

```
var str = "Visit Microsoft!";
var res = str.replace("Microsoft", "W3Schools");
// result: Visit W3Schools!
```

More on replace.

Example–replace a word on the page

Let's try using replace and innerHTML to replace a word on a webpage. Here's what we want to accomplish:

1. get the content of the page in a string
2. use replace to replace a particular word with a new word
3. replace the content with the modified content

More on replace.

Example–replace a word on the page

Let's try using replace and innerHTML to replace a word on a webpage.

1. get the content of the page in a string

```
var content = document.body.innerHTML;
```

2. replace the word with new word *have to use a little regex here...*

```
var new_content = content.replace(/Wikipedia/g, "Rosa");
```

3. replace the content with the modified content

```
document.body.innerHTML = new_content;
```

More on replace.

getAttribute / setAttribute

You can get also get and set attributes of an HTML element with JS.

Get an attribute:

```
element.getAttribute(attributename);

var id = document.getElementById('div1').getAttribute('id');
```

You can also set attributes of an HTML element with JS:

```
element.setAttribute(attributename, new attribute value);

document.getElementById('logo').setAttribute("src", "new-logo.png");
```

More on getAttribute.

Example–change img src attribute

Let's try changing all the images on a page by re-writing their source attribute.

```
// first get all the images and store them in a variable. This
creates an array.
var images = document.getElementsByTagName("img");

// next iterate through the array and reset the src attribute.
for (var x = 0; x < images.length; x++) {
    images[x].setAttribute("src",
     "https://d15shllkswkct0.cloudfront.net/wp-content/blogs.dir/1/fi
     les/2019/07/nature-3294543_1280-photo-graphe-pixabay.jpg");
  };
```

createElement + appendChild

Use append to add an element to the page.

```
// first you need to create an element
var footnote = document.createElement("div");
footnote.innerHTML = "Rosa McElheny has appended this page.";

// Then use appendChild to append the element to the body.
document.body.appendChild(footnote);
```

More on append.

alert

Use alert to cause a pop-up message.

```
alert("Hello! I am an alert box!!");

alert("Hello\nHow are you?");

alert(location.hostname);
```

Use JS to change CSS

Changing CSS with JavaScript

The HTML DOM allows JavaScript to change the style of HTML elements.

```
document.getElementById(id).style.property = new style;
```

A couple of examples:

```
document.getElementById("p2").style.backgroundColor = "blue";

document.getElementById("demo").style.display = "none";
```

You'll notice that the property name might not be written exactly the same as it is in CSS–you would omit any dashes, and use camelCase.

Example–make all the divs blue

Here's how you might make all the divs blue on a page.

```
var divs = document.querySelectorAll("div");
for (var x = 0; x < divs.length; x++) {
  divs[x].style.color = "blue";
}
```

Using a variable as a CSS value

One powerful thing about using JavaScript to change CSS is that you can set properties dynamically. Here's a simple example of how you might set the background color using values stored in variables.

```
var red = 32;
var blue = 18;
var green = 22;

document.body.style.backgroundColor =
"rgb(" + red + ", " + blue + ", " + green + ")";
```

Example–backgroundColor as a function of date

Let's say you want to set the background color based on the date.
Here's how this would work in the abstract:

```
var red = day;
var blue = month;
var green = year;
```

And here are the steps to doing this:

1. get the values for day, week, and month.
2. make sure that day, month, and year are all between 0 and 255...
3. use those values to set the background color

Example–backgroundColor as a function of date

1. Get the values for day, week, and month.

```
var today = new Date();

var day = today.getDate();
var month = today.getMonth() + 1;
var year = today.getFullYear();

console.log("day =" + day);
console.log("month =" + month);
console.log("year =" + year);
```

More on JS dates.

Example–backgroundColor as a function of date

2. make sure that day, month, and year are all between 0 and 255...

```
var today = new Date();

var day = today.getDate(); // 23 ✓
var month = today.getMonth(); // 3 ✓
~~var year = today.getFullYear();~~ // 2021 X

var red = day;
var blue = month;
var green = year / 255;
```

Example–backgroundColor as a function of date

Here's a slightly more sophisticated version. Using math, we've mapped the month and day into a range– the value for day will be 0 on the first day of the month, and 255 on the 31st day, and so on.

```
input * upper limit / highest possible input value

var red = day * 255 / 31;
var blue = month * 255 / 11;
var green = year / 255;

document.body.style.backgroundColor =
"rgb(" + red + ", " + blue + ", " + green + ")";
```

Example–set a random font size

Another thing that's easily done with JavaScript is randomizing. Let's say you want to set a random font size between 16 and 164px. Here's how you would do that in the abstract.

```
var size = some random number;

document.queryGetElementbyTag("p").style.fontSize = size + "px";
```

Example–set a random font size

To get the random number, use this function:

```
Math.random();  // returns a random number between 0 and 1
```

To get a random number in a range, modify that result:

```
Math.random() * 10 // returns a random number between 0 and 10
```

To get a random integer:

```
Math.floor(Math.random() * 10)
// returns a random integer between 0 and 10
```

Example–set a random font size

Now let's modify our code. We want a value between 16 and 164px, so we need to use some math:

```
var size = Math.floor(Math.random() * 164) + 16;

var fontSize = size + "px";

var links = document.getElementsByTagName("a");
for (var x = 0; x < links.length; x++) {
  links[x].style.fontSize = fontSize;
}
```

Example–set a random font size

One variation on this: let's say we want every link to be a different font size on the page. We need to re-organize our code a little bit.

Instead of:
1. pick a random number; 2. get all the links; 3. one by one, make each link that font size

<u>We want to do:</u>
1. get all the links; 2. one by one, find a link, pick a random number, and set the font size; 3. go to the next link

```
var links = document.getElementsByTagName("a");
for (var x = 0; x < links.length; x++) {
    var size = Math.floor(Math.random() * 164) + 16;
    links[x].style.fontSize = size + "px";
}
```

Making a Bookmarklet

Credit to Casey Watts for the tutorial.

To make a bookmarklet we have three steps:

1. Write some javascript code that you want in a bookmarklet (using the console).
2. Put `javascript:` in front of the code.
3. Wrap everything in an IIFE (immediately invoked functional expression) so the page doesn't freak out.

1. Write some javascript code that you want in a bookmarklet (using the console).

```
document.body.style.background = 'pink';
```

2. Put `javascript:` in front of the code.

```
javascript:document.body.style.background =
'pink';
```

3. Wrap everything in an IIFE (immediately invoked functional expression) so the page doesn't freak out.

```
javascript:(function(){CONTENTGOESHERE})();
```

```
javascript:(function(){document.body.style.background =
'pink';})();
```

4. Test your bookmarklet!

You can preview a bookmarklet quickly by pasting it into the URL bar.

You'll have to retype `javascript:` at the beginning, that gets stripped out.

Spend some time working on a bookmarklet, as a warmup for your extension project next week. We'll share towards the end of class.