

# The DOM

UC Berkeley Graduate School of Journalism

# Document Object Model (DOM)

<html>

<head>

</head>

<body>

    <h1>Something</h1>

</body>

</html>

# Document Object

## **properties**

`document.images`  
`document.links`  
`document.title`  
`document.URL`  
`document.cookie`

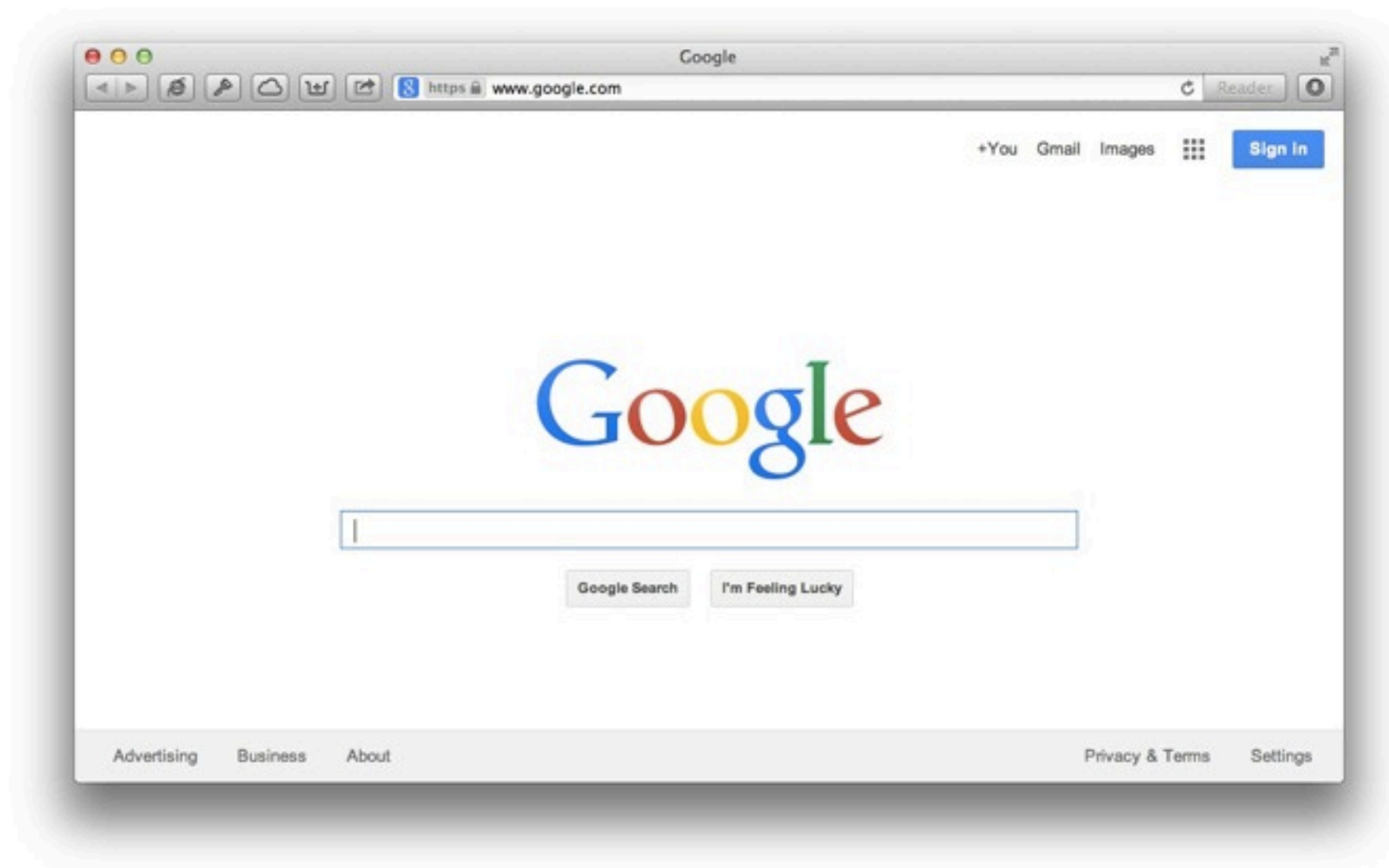
## **methods**

`document.getElementById()`  
`document.write()`  
`document.createElement()`

## **events**

`click`  
`onmousemove`  
`onkeypress`

# Window Object



Has different properties and methods.  
Used for getting window's size, attributes.

# Manipulating the DOM with jQuery

jQuery == \$

jQuery( '#response' )  
\$( '#response' )

# Manipulating the DOM with jQuery

element (css selector)

arguments/options  
passed into this command

```
$( '#response' ).method( arg, arg, arg )
```


jQuery

jQuery  
object

jQuery  
command

# html and text methods

```
$( '#response' ).html( '<h3>Hello</h3>' );  
$( '#response' ).text( '<h3>Hello</h3>' );
```



Both of these will **replace** the contents of this tag. They both take a string for their argument. The first will interpret html, the second will print out the literal string.

# append and prepend methods

```
$( '#response' ).append( '<h3>Hello</h3>' );  
$( '#response' ).prepend( '<h3>Hello</h3>' );
```

Similar to html, but instead of replacing the contents of the tag, it will add to whatever is already there.



# before and after methods

```
$( '#response' ).before( '<h3>Hello</h3>' );  
$( '#response' ).after( '<h3>Hello</h3>' );
```

Insert content *before/after* the jQuery object tag reference, rather than inside it.

before <h3>Hello</h3>      prepend  
          <div id="response">content</div>  
after <h3>Hello</h3>      append

# Creating new jQuery objects

`$( '#response' )`

Selected element  
which is already in  
the DOM

brackets

`$( '<div />' )`

Create a new element,  
but it's not yet in the  
DOM (our website)

# appendTo method

```
$(' <h1>hello</h1> ' ).appendTo( '#response' );
```

AppendTo is different than append, because it takes the jQuery object in question and appends it to the argument (right of the command)

```
$(' #response' ).append( ' <h1>hello</h1> ' );
```

# attr method

```
$( '#myimage' ).attr( 'src', 'photo2.jpg' );
```

```

```



The diagram consists of two dotted lines. The first line starts from the 'src' argument in the jQuery code and extends downwards, then turns left to point at the 'src' attribute in the HTML tag. The second line starts from the 'photo2.jpg' argument in the jQuery code and extends downwards, then turns left to point at the 'photo1.jpg' value in the HTML tag. This illustrates how the jQuery method updates the attribute value in the DOM.

# attr method in use

our html

```
$( '<img />' )
```

# attr method in use

our html

```
<img>
```

```
$( '<img />' ).appendTo( 'body' )
```

# attr method in use

our html

```

```

```
$( '<img />' ).appendTo( 'body' )  
                  .attr( 'src', 'photo.jpg' )
```

# attr method in use

our html

```
<a href="http://yahoo.com">
```

```
$('a').attr('href', 'http://google.com')
```



# attr method in use

our html

```
<a href="http://google.com">
```

```
$('a').attr('href', 'http://google.com')
```

# addClass removeClass methods

```
$( '#response' ).addClass( 'redparagraph' );
```

```
$( '#response' ).removeClass( 'redparagraph' );
```

Add a CSS class name to an element, or remove it. The effects of that CSS class will be immediate.

```
$( '#res' ).attr( 'class', 'redparagraph' );
```

# css method

```
$( '#response' ).css( 'color', 'black' );  
$( '#response' ).css({  
    "color"      : "black",  
    "background" : "green",  
    "width"      : "220px"  
});
```

Change the CSS of an element (add it inline using the style attribute.) This method can take either two arguments, or one. If one, it must be an object of multiple styles.

# addClass removeClass methods

```
$( '#response' ).animate({ "left": 300 },  
{  
    duration : 3000,  
    queue     : false,  
    complete : function() {  
    }  
});
```

Will animate CSS properties from their current value.  
For example, if the current CSS left property is at 0,  
the example above will animate it to 300px.  
Takes two objects: first is CSS properties, second is  
several optional properties for controlling animation.

# Chaining

```
$( '#response' ).css( 'color', 'black' )  
$( '#response' ).text( 'Hey there!' )  
$( '#response' ).fadeIn(400)  
$( '#response' ).append( '<p>Why, hello</p>' )
```

```
$( '#elm' ).do().do().do().do()
```



Order sometimes matters

# eq method

```
$( 'div' ).eq(0).fadeOut();  
$( '.bylines' ).eq(5).append( 'Hello' );
```

Eq method takes a zero-index argument, and selects the specific element which we should affect. If the main jQuery object returns only one element, only eq(0) will work.

# on method (for events)

```
$('#button').on("click", function(){  });
```

On method attaches an event to a function to be called when that event is fired. Events must exist on the jQuery object already.

# on method (for events)

```
$('video').on("ended", function(){  
    $(this).fadeOut();  
});
```

Attaching an event from the video tag to trigger once the video has ended.

`$(this)` is a jQuery object which refers to the specific element which triggered this event.