

JAVASCRIPT

WHAT IS JAVASCRIPT

JavaScript is a light-weight and object-oriented scripting language.

JS is a dynamic, cross-platform language originally used for creating interactive websites.

JS is now popular server-side and client side-side.

JavaScript runs in all browsers and most modern browsers have a JavaScript console that can be used for writing, running, and debugging code.

JavaScript is a loosely typed or dynamic language, so you don't have to declare a variable type ahead of time.

CLIENT-SIDE JAVASCRIPT

Client-side JavaScript extends the core language by supplying objects to control a browser and its Document Object Model (DOM).

SERVER-SIDE JAVASCRIPT

Server-side JavaScript extends the core language by supplying objects relevant to running JavaScript on a server.

JAVASCRIPT COMPARED TO C#

JavaScript

Object-oriented. No distinction between types of objects. Inheritance is through the prototype mechanism, and properties and methods can be added to any object dynamically.

Variable data types are not declared (dynamic typing).

Cannot automatically write to hard disk.

C#

Class-based. Objects are divided into classes and instances with all inheritance through the class hierarchy. Classes and instances cannot have properties or methods added dynamically.

Variable data types must be declared (static typing).

Can automatically write to hard disk.

CONSOLE OUTPUT

PRINTING TO THE CONSOLE

When programming in JavaScript, much of the output is tracked via the console.

VARIABLES

VARIABLES

You must first declare a variable then initialize the variable. These can be done in one line or separate statements. The equals sign is used to assign a value.

LOOSE TYPE

Because JavaScript is loosely typed, there are two ways of declaring a variable. You can use the keyword `var` or you can simply assign a value to a variable name.

VARIABLE NAMING

Must begin with a letter, _, or \$.

Must contain letters, numbers, _, or \$.

As with C# and other languages, camel-casing and similar conventions are good practice.

DECLARE A VARIABLE

```
var myVariable;  
myVariable = [value];
```

INITIALIZE A VARIABLE

```
myVariable = [value];
```


DATA TYPES

DATA TYPES

JavaScript has three primitive data types: strings, numbers, and booleans. The value of a variable differentiates its type.

•DETROIT•

GRAND
CIRCUS
•DETROIT•

•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

```
var myVariable = 25;           // number
var myVariable = "word";       // string
var myVariable = "25";         // string
var myVarialbe = true;         // boolean
```

CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

KEYWORDS

break

continue

debugger

do...while

for

function

if...else

return

switch

try...catch

var

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

OPERATORS

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

ARITHMETIC

Addition

+

Subtraction

-

Multiplication

*

Division

/

Modulus

%

Increment

++

Decrement

--

ASSIGNMENT

standard assignment

=

plus equals

+=

minus equals

-=

assignment by multiplication

*=

assignment by division

/=

COMPARISON

Equality

===

Inequality

!==

Greater than

>

Greater than or equal to

>=

Less than

<

Less than or equal to

<=

DOUBLE EQUALS

- 'Shallow' equals ==
- 'Shallow' inequality !=

Performs a type coercion before checking equality.

TYPE COERCION

```
true == "true" // > false
true === "true" // > false
"1" == 1 // > true
"1" === 1 // > false
"1" != 1 // > false
```

Don't use double equals.

The equivalent to C#'s == is ===.

GRAND
CIRCUS
-DETROIT-

GRAND
CIRCUS
-DETROIT-

GRAND
CIRCUS
-DETROIT-

GRAND
CIRCUS
-DETROIT-

GRAND
CIRCUS
-DETROIT-

GRAND
CIRCUS
-DETROIT-

GRAND
CIRCUS
-DETROIT-

GRAND
CIRCUS
-DETROIT-

GRAND
CIRCUS
-DETROIT-

GRAND
CIRCUS
-DETROIT-

GRAND
CIRCUS
-DETROIT-

GRAND
CIRCUS
-DETROIT-

GRAND
CIRCUS
-DETROIT-

GRAND
CIRCUS
-DETROIT-

GRAND
CIRCUS
-DETROIT-

GRAND
CIRCUS
-DETROIT-

GRAND
CIRCUS
-DETROIT-

GRAND
CIRCUS
-DETROIT-

GRAND
CIRCUS
-DETROIT-

IF/ELSE

GRAND
CIRCUS
-DETROIT-

CIRCUS
-DETROIT-

GRAND
CIRCUS
-DETROIT-

GRAND
CIRCUS
-DETROIT-

GRAND
CIRCUS
-DETROIT-

GRAND
CIRCUS
-DETROIT-

GRAND
CIRCUS
-DETROIT-

```
if(condition) {  
    Statement 1;  
} else if {  
    Statement 2;  
} else if {  
    Statement 3;  
} else {  
    Statement 4;  
}
```

GRAND
CIRCUS
-DETROIT-

CIRCUS
-DETROIT-

GRAND
CIRCUS
-DETROIT-

GRAND
CIRCUS
-DETROIT-

GRAND
CIRCUS
-DETROIT-

GRAND
CIRCUS
-DETROIT-

GRAND
CIRCUS
-DETROIT-

GRAND

GRAND

The background of the slide is a repeating pattern of the Grand Circus Detroit logo in a light gray color. The logo consists of a stylized building icon above the words "GRAND CIRCUS" and "DETROIT" below it.

FUNCTIONS

FUNCTION DECLARATION

In JavaScript, a function can be written for reusable blocks of code. You must declare a function, give the function some instructions, and call the function.

FUNCTION DECLARATION

```
function myFunction( ) {  
    Do something;  
}  
myFunction( );
```

```
function myFunction(parameters) {  
    Do something;  
}  
myFunction(parameters);
```


FUNCTION DECLARATION

```
function myFunction(x,y) {  
    return x*y;  
}  
myFunction(3,4);           // returns 12
```

PART TWO: DOM

GOALS FOR TODAY

1. Objects in JavaScript
2. The Document Object Model (DOM)

INTERACTIVITY

PROMPT()

prompt() is a DOM method that allows a program to (you guessed it) prompt a user for input.

If the method is just called by itself, it simply returns the user's input in the form of a string. But the return from the method can also be stored to a variable and used. i.e.

```
var name = prompt('Please enter your name');
```

CODE CHALLENGE

The background of the image is a repeating pattern of a light gray watermark logo. The logo consists of the words "GRAND CIRCUS" in a serif font, with a stylized building icon above the word "CIRCUS". Below the main text, the word "DETROIT" is written in a smaller, sans-serif font, flanked by two small horizontal lines.

SETUP

Go to jsbin.com

STEP 1

Work in Pairs

1. Write a simple program that accepts user input from a prompt. In the prompt, say “Please enter your name”.
2. Store the result in a variable.
3. Use an alert() function to print “Hello ” where is the input entered by the user.

Step 2 – Simple Error Checking What happens if the user types some kind of unusable input (i.e. only one letter for their name). No one has a name that short! (For this example, assume this is true)

- Add some logic to your function that checks the length of the input (ex. shorter than 2 characters) and alerts an error if their name is too short.
Bonus: If the user input is not accepted, have the program ask again until the input is acceptable.

```
var name = "";
var done = false;
function ask() { return prompt("Please enter your name");}
function helloName() {
  while(!done) {
    userInput = ask();
    if(isTooShort(userinput)) {
      alert("Your name is short. Too short.\n Do it again!");
    } else {
      alert("Hello " + userInput);
      done = true;
    }
  }
}
```

Sample solution

```
function isTooShort(inputName) {  
  return inputName.length < 2;  
}  
helloName(name);
```

Sample solution continued

DOM

Document Object Model

The document object model (DOM) is an interface which allows programs and scripts to dynamically access and update the content, style and structure of a document

The DOM is a W3C (World Wide Web Consortium) standard and includes the Core DOM, XML DOM, and HTML DOM

The HTML DOM is a standard model for HTML documents and defines how to get, change, add, or delete HTML elements

In order to integrate services like Twitter, Facebook, etc. into our own applications, we have to use their APIs (application programming interfaces). The HTML DOM is an API for manipulating HTML documents.

```
var byNode = document.body; //  
var htmlNode = document.body.parentNode; //  
var childNodes = document.body.childNodes[]; // all elements inside
```

The DOM has a tree structure and identifies objects using nodes

Watch what happens when you put this command
in your js file.

```
document.write("JavaScript all the things!");
```

For extra fun, try that in the console on any
website. In the DOM, your HTML page is known as
the 'document'

It's possible to find elements on the HTML page by parent, sibling, or child node, but that's time-consuming and will make you crazy.

```
<li class="food">Pizza</li>
<li class="food">Sushi</li>
<li class="food">Hummus</li>
```

```
var listItems = document.getElementsByTagName('li');

for (var index = 0; index < listItems.length; index++) {
  var listItem = listItems[index];
  // Do stuff with listItem
}
```

HTML elements can be located by tag

```
<p id="happy">A happy paragraph!</p>
```

```
var paragraph = document.getElementById('happy');
```

... or by id or class

```
var img = document.getElementById('myImage');  
img.getAttribute('src');  
img.setAttribute('src', './images/newImage');
```

The attributes of HTML elements can also be accessed and modified through the DOM

DOM nodes have a property called innerHTML which lets you access and modify the contents of the node


```
<!DOCTYPE html>
<html>
  <head>
    <title>JavaScript Demo</title>
    <script src="sample.js"></script>
  </head>
  <body>
    <p id="happy">A happy paragraph!</p>
  </body>
</html>
```

Set up a simple HTML file

```
document.body.innerHTML = '<p>I changed the whole page!</p>'
```

Set up sample.js like this. What happens when you load the page?

Now change sample.js to this (remove all previous statements)

What happens? To make sure you notice, add border to all tags via CSS

```
var paragraph = document.createElement('p');  
paragraph.innerText = "I made a new tag!";  
document.body.appendChild(paragraph);
```

QUESTIONS

JQUERY

GOALS FOR TODAY

1. Introduction to jQuery
2. jQuery 101

INTRODUCTION TO JQUERY

WHAT IS A LIBRARY?

A software library is a collection of functions. When you include a library in your code, you have access to all the functions contained in the library. jQuery is a big honkin' library packed full with functions intended to help you 'write less, do more' when working with the DOM

WHAT DOES JQUERY HELP WITH?

jQuery simplifies DOM scripting and other common JavaScript tasks including: HTML element selection and manipulation, CSS manipulation, and JavaScript events and animations

FACTS ABOUT JQUERY

- The most popular JavaScript library
- Extensive Docs
- Numerous tutorials online
- Used by 20 million websites

1 FACT

jQuery is still just JavaScript. If you understand JavaScript, you can understand jQuery.

I'M SOLD! HOW DO I GET ME SOME JQUERY?

Two ways:

1. Download the library and store it locally in your project folder.
2. Link to a live version of the library via CDN.

Bonus: Display each object with a statement using concatenation. Ex: Sparky the dog. Coco the cat.

WHICH WAY IS BEST?

Each has pros and cons. Live code can change but jQuery is popular enough that there are stable versions of each major release of jQuery out there. Keeping it local is stable but bloats the overall size of your site.

WHICH WAY IS BEST? (CONT'D)

For this class, download it. The network might flake and then your site will not work. Plus a call will be made to get the whole library every time you refresh (which will be a lot).

WHICH WAY IS BEST? (CONT'D)

For active development it's fine to link. For production sites, I tend to lean toward sourcing it locally with your project.

HOW TO DOWNLOAD JQUERY FOR THIS CLASS

1. Go to jquery.com/download
2. Click the link Download the uncompressed, development jQuery 1.11.1. This takes you to a raw text file.
3. Select All > copy > paste into empty file > save As 'jquery-1.11.1.js'
4. Place the saved file into a 'lib' folder in your site project

Or just link to a hosted CDN in a script tag on your page. Both script tags link to a working jQuery file.

```
>head<
  <script src="lib/jquery-1.11.1.js"></script>
  <script src="http://code.jquery.com/jquery-1.11.0.min.js"></script>
</head>
```

JQUERY SELECTORS

Does that selector syntax look familiar?

```
var divs = $("div");  
// All divs on the page  
var happyThing = $("#happy");  
// Element with id "happy"  
var rounds = $(".roundedCorner");  
// Elements with class "roundedCorner"
```

jQuery has hundreds of action that can be performed on any element

All actions are functions (or methods).

```
$(selector).action();
```

jQuery makes it simple to accomplish a huge number of common tasks such as updating element's attributes or CSS (and pretty much anything else)

```
var img = $('#myPicture');  
img.attr('src');  
img.attr('src', 'http://www.myPictureLivesHere.com');  
  
img.css('width');  
img.css('width', 200px);
```

QUESTIONS

The background of the slide is a repeating pattern of the Grand Circus Detroit logo in a light gray color. The logo consists of the words "GRAND CIRCUS" in a serif font, with "DETROIT" in a smaller font below it, and a stylized building icon above the word "GRAND".

JQUERY 101

JQUERY

If you haven't done so yet, please download the example project named 'jquery-demo.zip' and 'jquery101-exercises.zip' from Slack.

jQuery interacts with the DOM to access and modify HTML elements.

`$()` is a jQuery function which turns whatever is inside the parentheses into a jQuery object.

.ready() is a jQuery function which tells a script to wait to execute until an HTML document has loaded.

GRAND
OLD HOUSE

```
$ (document) .ready();
```

CIRCUS
-DETROIT-

CIRCUS
• D E T R O I T •

The argument for the `.ready()` function is an anonymous function which will be executed as soon as the HTML document loads.

We can put other jQuery functions inside this outer function to make sure our code waits for the whole document to load before executing.

GRAND
CIRCUS
-DETROIT-

GRAND
CIRCUS
-DETROIT-

GRAND
CIRCUS
-DETROIT-

GRAND
CIRCUS
-DETROIT-

GRAND
CIRCUS
-DETROIT-

GRAND
CIRCUS
-DETROIT-

GRAND
CIRCUS
-DETROIT-

GRAND
CIRCUS
-DETROIT-

```
$(document).ready(function() {  
  $("header").slideDown('slow')  
});
```

GRAND
CIRCUS
-DETROIT-

GRAND
CIRCUS
-DETROIT-

GRAND
CIRCUS
-DETROIT-

GRAND
CIRCUS
-DETROIT-

GRAND
CIRCUS
-DETROIT-

GRAND
CIRCUS
-DETROIT-

GRAND
CIRCUS
-DETROIT-

GRAND
CIRCUS
-DETROIT-

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

```
$('.img').fadeTo("slow", 0.5);  
var $img = $("img");  
$img.fadeTo("slow", 0.5);
```

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

In addition to elements, jQuery can use class and id selectors.

Any element that can be targeted with CSS can also be selected and modified using jQuery.

jQuery can also be used to modify an element's CSS styles.

THE .CSS() METHOD

The .css() method is used to dynamically modify CSS styles.

```
$("#first").click(function() {  
  $("#first").css({  
    'background-color': 'blue'  
  });  
});
```

.css()

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

JQUERY

EFFECTS

In jQuery, functions like `fadeOut()` and `slideDown()` that add animations to a web page are called effects.

In jQuery, functions like `click()` and `hover()` that refer to user interactions with the browser are called events.

QUESTIONS

EXERCISE

IN YOUR PROJECT:

- Find the img tag in the page.
- In your JavaScript file add the following code (some of it is already there).

```
$(document).ready(function() {  
    $("img").mouseenter(function() {  
        $("img").fadeTo("fast", 0.5);  
    });  
});
```

IN YOUR PROJECT (CONT'D):

- Save and reload the page. Mouseover the image, what happens?
- Can you figure out how to make the img return to full opacity when your mouse is no longer inside the element?

•DETROIT•

•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

```
$(document).ready(function() {  
  $("img").mouseenter(function() {  
    $("img").fadeTo("fast", 0.5);  
  });  
  $("img").mouseleave(function() {  
    $("img").fadeTo("fast", 1.0);  
  });  
});
```

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

JQUERY

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

.CLICK()

.click() is one of the most frequently-used event handlers in jQuery.

.click() captures a click on the selected element.

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND

GRAND
CIRCUS
•DETROIT•

```
$("#div").click(function() {  
  // Do stuff when I click a div!  
});  
$("#img").click(function() {  
  // Do stuff when I click an img!  
});
```

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

this is a keyword which tells jQuery, “I only want to do stuff to the specific element with which I’m interacting”.

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

```
$(document).ready(function() {  
  $("div").click(function() {  
    $(this).fadeOut("slow");  
  });  
});
```

CIRCUS
•DETROIT•

this

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS

GR
CIR

The background of the slide is a repeating pattern of a light gray watermark logo. The logo consists of a stylized house icon with a flag on top, followed by the text "GRAND CIRCUS" and "DETROIT" in a smaller font below it.

JQUERY EFFECTS

.HIDE()

.hide() hides a visible element

```
$("#myPicture").hide();
```

.SHOW()

.show() hides a visible element

```
$("#myPicture").show();
```

.TOGGLE()

.toggle() hides or shows an element depending on its current state (hides a visible element and shows a hidden element).

.FADEIN()

.fadeIn() displays an element by fading it to full opacity.

.FADEOUT()

.fadeOut() hides an element by fading it to full transparency.

.FADETO()

.fadeTo() adjusts the opacity of an element according to the opacity specified in the arguments.

```
$("button").click(function() {  
    $("div").fadeTo(0.25);  
});
```

.FADETOGGLE()

.fadeToggle() hides or shows an element depending on its current state by animating its opacity.

.SLIDEUP()

.slideUp() hides an element with an upward sliding motion.

.SLIDEDOWN()

.slideDown() displays an element with a downward sliding motion.

.SLIDETOGGLE()

.slideToggle() hides or shows an element with a sliding motion depending on its current state.

.DELAY()

.delay() specifies the time between each event queued on an element.

```
$("#button").click(function() {  
    $("#div.first").slideUp(300).delay(800).fadeIn(400);  
    $("#div.first").slideUp(300).delay(800).fadeIn(400);  
});
```

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

QUESTIONS

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

JQUERY EVENTS

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

JQUERY EVENTS

An event refers to any kind of action a web page can recognize, including mouse clicks, hovering, scrolling, page loading, and resizing windows.

jQuery uses event handlers to define specific behavior which should occur in the web page when certain events are triggered.

What are some real-world examples of events and the types of responses they trigger?

.CLICK()

.click() captures a click on the selected element.

```
$("#div").click(function() {  
    //Do stuff when I click a  
  
    !  
});  
$("#img").click(function() {  
    //Do stuff when I click an <img>!  
});
```

.dblclick()

.dblclick() captures a double-click on the selected element.

.MOUSEENTER()

.mouseenter() is triggered when the mouse is moved over a selected element.

.MOUSELEAVE()

.mouseleave() is triggered when the mouse is moved out of the area of a selected element.

.MOUSEDOWN()

.mousedown() is triggered when the mouse is moved over a selected element and the mouse button is pressed.

.MOUSEUP()

.mouseup() is triggered when the mouse is moved over the selected element and the mouse button is released.

.MOUSEMOVE()

.mousemove() is triggered when the mouse is moved within the area of a selected element.

.LOAD()

The .load() event is triggered on an element when the element and all its sub-elements have loaded.

```
$("#book").load(function() {  
    // Do stuff!  
});
```


.UNLOAD()

The .unload() event is triggered when a user navigates away from the page (don't do that).

```
$(window).unload(function() {  
    alert("Thanks for visiting");  
});
```

.ERROR()

The .error() event is triggered when an element is not loaded correctly (for example, an image isn't found).

```
$("img").error(function() {  
    $(this).hide();  
}).attr("src", "missing.png");
```

.RESIZE()

The .resize() event is sent to the window when the size of the browser window changes.

```
$(window).resize(function() {  
    console.log("You resized the browser window");  
});
```

.SCROLL

The .scroll() event is sent to an element when the user scrolls to a different place in the element.

```
$(window).scroll(function() {  
    $("#pictureDiv").fadeOut("slow");  
});
```

QUESTIONS

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

LAB

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

GRAND
CIRCUS
•DETROIT•

ROCKIN' WITH JQUERY

Create a website that lists and demonstrates the jQuery topics we've covered so far. Include:

- Selectors
- Effects
 - Slide actions
 - Fade actions
 - Show / hide
- Events
 - click
 - hover
 - dblClick