# Handout #5 - JavaScript

## Scripts

| in <head> | at end of <body> | in <head> with defer |
| --- | --- | --- |
| ❌ pauses rendering while script is being downloaded, parsed, and executed | ✔️ rendering continues until script tag is encountered | ✔️ execution is paused until DOM has been parsed |
| ✔️ download begins immediately | ❌ download does not start until script tag is encountered | ✔️ download begins immediately |
| ✔️ works with inline scripts | ✔️ works with inline scripts | ❌ not supported for inline scripts |

[**https://flaviocopes.com/javascript-async-defer/**](https://flaviocopes.com/javascript-async-defer/)

## Variable Declarations

| var | let | const |
| --- | --- | --- |
| ❌ function-scoped,  hoisted to top of function | ✔️ block-scoped | ✔️ block-scoped |
| ❌in global scope, overrides window object | ✔️ in global scope, does not override window object | ✔️ in global scope, does not override window object |
| ❌ can be redeclared within same scope | ✔️ cannot be redeclared within same scope | ✔️ cannot be redeclared within same scope |
| ❌ can be reassigned | ❌ can be reassigned | ✔️ cannot be reassigned |
| ❌ objects are mutable | ❌ objects are mutable | ❌ objects are mutable |

[**https://hackernoon.com/why-you-shouldnt-use-var-anymore-f109a58b9b70**](https://hackernoon.com/why-you-shouldnt-use-var-anymore-f109a58b9b70)

## Data Types

| data type | description |
| --- | --- |
| [undefined](https://developer.mozilla.org/en-US/docs/Glossary/undefined) | no assigned value |
| [boolean](https://developer.mozilla.org/en-US/docs/Glossary/boolean) | true/false |
| [number](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Number) | stored in [64-bit floating point format (IEEE 754)](https://en.wikipedia.org/wiki/Double-precision_floating-point_format) |
| [bigint](https://developer.mozilla.org/en-US/docs/Glossary/bigint) | integers stored in an [arbitrary precision format](https://en.wikipedia.org/wiki/Arbitrary-precision_arithmetic) |
| [string](https://developer.mozilla.org/en-US/docs/Glossary/string) | a sequence of characters |
| [symbol](https://developer.mozilla.org/en-US/docs/Glossary/symbol) | *advanced topic, not covered in this course* |
| [object](https://developer.mozilla.org/en-US/docs/Glossary/object) | null, objects, and arrays |
| [function](https://developer.mozilla.org/en-US/docs/Glossary/function) | can be invoked/called, can receive parameters and return a value, created with function notation or arrow notation |

[**https://developer.mozilla.org/en-US/docs/Web/JavaScript/Data\_structures**](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Data_structures)

## Falsy/Truthy

The following values are defined to be **Falsy:**

**false, 0, -0, 0n, '', null, undefined, NaN**

Everything else is **Truthy.** Including:

**[] and {}**

Source: <https://developer.mozilla.org/en-US/docs/Glossary/Falsy>

## Looping through an array

| // define the array **const** arr = [ 'cat', 'dog', 'unicorn' ];  // with a while loop **let** i = 0; **while** (i < arr.length) {  console.log(arr[i]);  ++i; }  // with a do-while loop **let** i = 0; **if** (arr.length > 0) {  **do** {  console.log(arr[i]);  ++i;  } **while** (i < arr.length); }  // with a counted for loop **for** (**let** i = 0; i < arr.length; ++i) {  console.log(arr[i]); }  // with a for-in loop **for** (**const** index **in** arr) {  console.log(arr[index]); }  // with a for-of loop **for** (**const** item **of** arr) {  console.log(item); } |
| --- |

## Functions

| // function declaration **function** **add**(x, y) { **return** x + y; }  // function expression **const** add = **function** **add**(x, y) { **return** x + y; }  // arrow function **const** add = (x, y) => x + y; |
| --- |

## DOM API Quick Reference

**document methods & properties**

| document.querySelector(selector) | Find the first node that matches a selector. |
| --- | --- |
| document.querySelectorAll(selector) | Find all nodes that match a selector. |
| document.addEventListener(eventName, handler) | Attach an event handler to the document. |

**element methods & properties**

| element.querySelector(selector) | Find the first descendent that matches a selector. |
| --- | --- |
| element.querySelectorAll(selector) | Find all descendents that match a selector. |
| element.closest(selector) | Find the closest ancestor that matches a selector. |
| element.parentNode | Get the parent node. |
| element.children | Get all direct child nodes. *(May contain text nodes.)* |
| element.nextElementSibling | Get the next sibling element. *(Skips text nodes.)* |
| element.previousElementSibling | Get the previous sibling element. *(Skips text nodes.)* |
| element.innerHTML | Get/Set the HTML contents. |
| element.textContent | Get/Set the text contents. |
| element.value | Get/Set the current value. |
| element.checked | Get/Set the checked state of a radio button or checkbox. |
| element.hasAttribute(name) | Returns true if the element has the named attribute. |
| element.getAttribute(name) | Get the value of a named attribute. |
| element.setAttribute(name, value) | Set the value of a named attribute. |
| element.removeAttribute(name) | Remove the named attribute. |
| element.classList.contains(className) | Returns true if the element has the specified class. |
| element.classList.add(className) | Adds specified class to the element. |
| element.classList.remove(className) | Removes specified class from the element. |
| element.dataset.*myDataName* | Get/Set arbitrary data on the element. |
| element.addEventListener(eventName, handler) | Attach an event handler to the element. |
| element.removeEventListener(eventName, handler) | Remove an event handler from the element. |
| element.dispatchEvent(eventObject) | Trigger an event on the element. |
| element.append() | Insert content at the end of the element. *(as child)* |
| element.prepend() | Insert content at the beginning of the element. *(as child)* |
| element.after() | Insert content after the element*. (as sibling)* |
| element.before() | Insert content before the element. *(as sibling)* |
| element.remove() | Remove the element from the DOM. |
| element.innerHTML = '' | Remove all children from the DOM. |

**events**

| DOMContentLoaded | Fires when the initial HTML document has been fully loaded and parsed, without waiting for stylesheets, images, frames, etc to finish loading. |
| --- | --- |
| click | Fires when a pointing device is clicked inside an element. [(See full explanation)](https://developer.mozilla.org/en-US/docs/Web/API/Element/click_event) |
| focus | Fires when an element receives focus. |
| blur | Fires when an element has lost focus. |
| mouseenter | Fires when a pointing device enters the border of an element. |
| mouseleave | Fires when a pointing device leaves the border of an element. |
| keydown | Fires when a key is pressed on the keyboard. |

**event object**

| new Event(eventName) | Construct a new event object. *Used with dispatchEvent().* |
| --- | --- |
| event.currentTarget | The element that triggered this event. |
| event.preventDefault() | Prevent the default behaviour for this event, |

Find the full Event API at <https://developer.mozilla.org/en-US/docs/Web/API/Event>

## jQuery API Quick Reference

Find the full docs at <https://api.jquery.com/>

| $(() => { /\* dom loaded \*/ }); | Execute a function when the DOM is fully loaded.  *Formerly known as .ready()* |
| --- | --- |
| $(selector) | Select DOM elements using a CSS style selector.  *Will select all elements that match the selector.* |
| $(element) | Wrap a DOM element in a query object.  *Allows you to call query methods on it.* |
| .find(selector) | *For each element in the result set...*  find all descendents that match the selector. |
| .closest(selector) | *For each element in the result set...*  find the first ancestor which matches the selector. |
| .parent() | *For each element in the result set…*  get the parent element. |
| .children() | *For each element in the result set…*  get all of its direct children. |
| .next() | *For each element in the result set…*  get the next element. |
| .prev() | *For each element in the result set…*  get the previous element. |
| .html() | Get the HTML contents of the **first** element in the result set. |
| .html(value) | Set the HTML contents for each element in the result set. |
| .text() | Get the **combined** text contents of all elements in the result set. |
| .text(value) | Set the text contents for each element in the result set. |
| .val() | Get the current value of the **first** element in the result set. |
| .val(value) | Set the current value for each element in the result set. |
| .prop(name) | Get a property value of the **first** element in the result set. |
| .prop(name, value) | Set a property value for each element in the result set. |
| .attr(name) | Get an attribute value of the **first** element in the result set. |
| .attr(name, value) | Set an attribute value for each element in the result set. |
| .removeAttr(name) | Remove an attribute from each element in the result set. |
| .hasClass(className) | Returns true if **any** element in the result set has the given class. |
| .addClass(className) | Adds the specified class(es) to each element in the result set. |
| .removeClass(className) | Removes the specified class(es) from each element in the result set. |
| .data(name) | Gets arbitrary data associated with the **first** element in the result set. |
| .data(name, value) | Sets arbitrary data for each element in the result set. |
| .on(eventName, handler) | Attach an event handler to the selected elements. |
| .one(eventName, handler) | Attach a **one-time** event handler to the selected elements. |
| .off(eventName, handler) | Detach an event handler from the selected elements. |
| .trigger(eventName) | Trigger an event on the selected elements. |
| .append(content) | Insert content at the end of each selected element. *(as child)* |
| .prepend(content) | Insert content at the beginning of each selected element. *(as child)* |
| .after(content) | Insert content after each selected element*. (as sibling)* |
| .before(content) | Insert content before each selected element. *(as sibling)* |
| .empty() | Remove all children of the selected elements from the DOM. |
| .remove() | Remove the selected elements from the DOM. |

**The above methods cover 90% of jQuery's use cases. Please memorize them!**

**TODO: each(), focus(), first(), length**