

```
// ATTENTION: THIS IS CODE FROM THE YOUTUBE CRASH COURSE. IT IS NOT MEANT TO  
RUN, IT IS JUST FOR LEARNING PURPOSES //
```

```
// LOGGING OUTPUT
```

```
alert('Hello World'); // Do not use for debugging. Stops script and only  
strings  
console.log('Hello World');  
console.error('This is an error');  
console.warn('This is a warning');
```

```
// VARIABLES - var, let, const
```

```
let age = 30;
```

```
// let can be re-assigned, const can not
```

```
age = 31;
```

```
// DATA TYPES - String, Number, Boolean, null, undefined
```

```
const name = 'Brad';  
const age = 37;  
const rating = 3.5;  
const isCool = true;  
const x = null;  
const y = undefined;  
let z; // undefined
```

```
// Check type
```

```
console.log(typeof z);
```

```
// STRINGS
```

```
// Concatenation
```

```
console.log('My name is ' + name + ' and I am ' + age);
```

```
// Template literal (better)
```

```
console.log(`My name is ${name} and I am ${age}`);
```

```
// String methods & properties
```

```
const s = 'Hello World';
```

```
let val;
```

```
// Get length
```

```
val = s.length;
```

```
// Change case
```

```
val = s.toUpperCase();
```

```
val = s.toLowerCase();
```

```
// Get sub string
```

```
val = s.substring(0, 5);
```

```
// Split into array
```

```
val = s.split('');
```

```
// ARRAYS - Store multiple values in a variable
```

```
const numbers = [1,2,3,4,5];
```

```
const fruits = ['apples', 'oranges', 'pears', 'grapes'];
```

```
console.log(numbers, fruit);
```

```
// Get one value - Arrays start at 0
```

```
console.log(fruits[1]);
```

```
59 |
60 // Add value
61 fruits[4] = 'blueberries';
62
63 // Add value using push()
64 fruits.push('strawberries');
65
66 // Add to beginning
67 fruits.unshift('mangos');
68
69 // Remove last value
70 fruits.pop();
71
72 // // Check if array
73 console.log(Array.isArray(fruits));
74
75 // // Get index
76 console.log(fruits.indexOf('oranges'));
77
78
79
80 // OBJECT LITERALS
81 const person = {
82   firstName: 'John',
83   age: 30,
84   hobbies: ['music', 'movies', 'sports'],
85   address: {
86     street: '50 Main st',
87     city: 'Boston',
88     state: 'MA'
89   }
90 }
91
92 // Get single value
93 console.log(person.name)
94
95 // Get array value
96 console.log(person.hobbies[1]);
97
98 // Get embedded object
99 console.log(person.address.city);
100
101 // Add property
102 person.email = 'jdoe@gmail.com';
103
104 // Array of objects
105 const todos = [
106   {
107     id: 1,
108     text: 'Take out trash',
109     isComplete: false
110   },
111   {
112     id: 2,
113     text: 'Dinner with wife',
114     isComplete: false
115   },
116   {
117     id: 3,
118     text: 'Meeting with boss',
```

```
119     isComplete: true
120   }
121 ];
122
123 // Get specific object value
124 console.log(todos[1].text);
125
126 // Format as JSON
127 console.log(JSON.stringify(todos));
128
129
130 // LOOPS
131
132 // For
133 for(let i = 0; i <= 10; i++){
134   console.log(`For Loop Number: ${i}`);
135 }
136
137 // While
138 let i = 0
139 while(i <= 10) {
140   console.log(`While Loop Number: ${i}`);
141   i++;
142 }
143
144 // Loop Through Arrays
145 // For Loop
146 for(let i = 0; i < todos.length; i++){
147   console.log(` Todo ${i + 1}: ${todos[i].text}`);
148 }
149
150 // For...of Loop
151 for(let todo of todos) {
152   console.log(todo.text);
153 }
154
155
156 // HIGH ORDER ARRAY METHODS (show prototype)
157
158 // forEach() - Loops through array
159 todos.forEach(function(todo, i, myTodos) {
160   console.log(`${i + 1}: ${todo.text}`);
161   console.log(myTodos);
162 });
163
164 // map() - Loop through and create new array
165 const todoTextArray = todos.map(function(todo) {
166   return todo.text;
167 });
168
169 console.log(todoTextArray);
170
171 // filter() - Returns array based on condition
172 const todo1 = todos.filter(function(todo) {
173   // Return only todos where id is 1
174   return todo.id === 1;
175 });
176
177
178 // CONDITIONALS
```

```
179
180 // Simple If/Else Statement
181 const x = 30;
182
183 if(x === 10) {
184   console.log('x is 10');
185 } else if(x > 10) {
186   console.log('x is greater than 10');
187 } else {
188   console.log('x is less than 10')
189 }
190
191 // Switch
192 color = 'blue';
193
194 switch(color) {
195   case 'red':
196     console.log('color is red');
197   case 'blue':
198     console.log('color is blue');
199   default:
200     console.log('color is not red or blue')
201 }
202
203 // Ternary operator / Shorthand if
204 const z = color === 'red' ? 10 : 20;
205
206
207
208 // FUNCTIONS
209 function greet(greeting = 'Hello', name) {
210   if(!name) {
211     // console.log(greeting);
212     return greeting;
213   } else {
214     // console.log(`${greeting} ${name}`);
215     return `${greeting} ${name}`;
216   }
217 }
218
219
220 // ARROW FUNCTIONS
221 const greet = (greeting = 'Hello', name = 'There') => `${greeting} ${name}`;
222 console.log(greet('Hi'));
223
224
225 // OOP
226
227 // Constructor Function
228 function Person(firstName, lastName, dob) {
229   // Set object properties
230   this.firstName = firstName;
231   this.lastName = lastName;
232   this.dob = new Date(dob); // Set to actual date object using Date
233   constructor
234   // this.getBirthYear = function(){
235   //   return this.dob.getFullYear();
236   // }
237   // this.getFullName = function() {
238   //   return `${this.firstName} ${this.lastName}`
```

```
238 // }
239 }
240
241 // Get Birth Year
242 Person.prototype.getBirthYear = function () {
243     return this.dob.getFullYear();
244 }
245
246 // Get Full Name
247 Person.prototype.getFullName = function() {
248     return `${this.firstName} ${this.lastName}`
249 }
250
251
252 // Instantiate an object from the class
253 const person1 = new Person('John', 'Doe', '7-8-80');
254 const person2 = new Person('Steve', 'Smith', '8-2-90');
255
256 console.log(person2);
257
258 // console.log(person1.getBirthYear());
259 // console.log(person1.getFullName());
260
261
262
263 // Built in constructors
264 const name = new String('Kevin');
265 console.log(typeof name); // Shows 'Object'
266 const num = new Number(5);
267 console.log(typeof num); // Shows 'Object'
268
269
270 // ES6 CLASSES
271 class Person {
272     constructor(firstName, lastName, dob) {
273         this.firstName = firstName;
274         this.lastName = lastName;
275         this.dob = new Date(dob);
276     }
277
278     // Get Birth Year
279     getBirthYear() {
280         return this.dob.getFullYear();
281     }
282
283     // Get Full Name
284     getFullName() {
285         return `${this.firstName} ${this.lastName}`
286     }
287 }
288
289 const person1 = new Person('John', 'Doe', '7-8-80');
290 console.log(person1.getBirthYear());
291
292
293 // ELEMENT SELECTORS
294
295 // Single Element Selectors
296 console.log(document.getElementById('my-form'));
297 console.log(document.querySelector('.container'));
```

```
298 // Multiple Element Selectors
299 console.log(document.querySelectorAll('.item'));
300 console.log(document.getElementsByTagName('li'));
301 console.log(document.getElementsByClassName('item'));
302
303 const items = document.querySelectorAll('.item');
304 items.forEach((item) => console.log(item));
305
306
307 // MANIPULATING THE DOM
308 const ul = document.querySelector('.items');
309 // ul.remove();
310 // ul.lastElementChild.remove();
311 ul.firstElementChild.textContent = 'Hello';
312 ul.children[1].innerText = 'Brad';
313 ul.lastElementChild.innerHTML = '<h1>Hello</h1>';
314
315 const btn = document.querySelector('.btn');
316 // btn.style.background = 'red';
317
318
319 // EVENTS
320
321 // Mouse Event
322 btn.addEventListener('click', e => {
323   e.preventDefault();
324   console.log(e.target.className);
325   document.getElementById('my-form').style.background = '#ccc';
326   document.querySelector('body').classList.add('bg-dark');
327   ul.lastElementChild.innerHTML = '<h1>Changed</h1>';
328 });
329
330 // Keyboard Event
331 const nameInput = document.querySelector('#name');
332 nameInput.addEventListener('input', e => {
333   document.querySelector('.container').append(nameInput.value);
334 });
335
336
337 // USER FORM SCRIPT
338
339 // Put DOM elements into variables
340 const myForm = document.querySelector('#my-form');
341 const nameInput = document.querySelector('#name');
342 const emailInput = document.querySelector('#email');
343 const msg = document.querySelector('.msg');
344 const userList = document.querySelector('#users');
345
346 // Listen for form submit
347 myForm.addEventListener('submit', onSubmit);
348
349 function onSubmit(e) {
350   e.preventDefault();
351
352   if(nameInput.value === '' || emailInput.value === '') {
353     // alert('Please enter all fields');
354     msg.classList.add('error');
355     msg.innerHTML = 'Please enter all fields';
356
357     // Remove error after 3 seconds
```

```
358     setTimeout(() => msg.remove(), 3000);
359   } else {
360     // Create new list item with user
361     const li = document.createElement('li');
362
363     // Add text node with input values
364     li.appendChild(document.createTextNode(`${nameInput.value}:
    ${emailInput.value}`));
365
366     // Add HTML
367     // li.innerHTML = `<strong>${nameInput.value}</strong>e:
    ${emailInput.value}`;
368
369     // Append to ul
370     userList.appendChild(li);
371
372     // Clear fields
373     nameInput.value = '';
374     emailInput.value = '';
375   }
376 }
```