

# Javascript Commands Without Commentary

(for use as reference on exams and quizzes)

---

## Embedding Javascript

```
1 <html>
2   <head>
3     ...
4     <script type="text/javascript" src="path/to/script1.js"
5       defer></script>
6     <script type="text/javascript" src="path/to/script2.js"
7       defer></script>
8   </head>
9 <body>
10  ...
11  <script>
12    // Javascript code goes here
13  </script>
14  ...
15 </body>
16 </html>
```

---

## Variables, Types, Comments, and Semicolons

```
1 /* When we declare variables we don't give
2    them a type. We only specify a name. */
3
4 let value, name, isHot, fun
5
6 value = 1 // type is Number
7 name = 'Crazy' // type is String
8 isHot = true // type is Boolean
```

value

1
---

name

'Crazy'
---------

isHot

true
------

fun

undefined
-----------

---

## Displaying Values

```
1 let name
2 name = 'Crazy'
3 console.log(name); // write to browser console
4 alert(name); // present in annoying pop-up box
5 document.write(name); // place on page in current location
```

Output of all three statements  
(but in 3 different locations):

Crazy

---

## Getting Input

```
1 let doMore, num
2 doMore = confirm("Continue?")
3 num = prompt("Enter fav number: ")
4 console.log(doMore)
5 console.log(typeof(doMore))
6 console.log(num)
7 console.log(typeof(num))
```

Of course the output depends on what the user does. Suppose the user clicks OK at the first prompt generated by line 2 and then types 10.2 followed by OK at the second prompt generated by line 3. Expected output (in the console):

```
true
boolean
10.2
string
```

---

## Arithmetic

```
1 let a, b, c, d, e, f, g, h;
2 a = "5.5"
3 b = 3
4 c = a + b
5 d = b + Number(a)
6 b++
7 e = b / 3
8 f = Math.floor(b / 3)
9 g = b % 2
10 h = Math.random()
```

a	"5.5"
b	4
c	"5.53"
d	8.5
e	1.333333
f	1
g	0
h	0.7697511504757932

---

## If Statements

Expected output (in console.log):

A is true

B is false

C is true

D is false

```
1 let x, y
2 x = "5.5"
3 y = 0
4 if (x == 5.5) {
5   console.log("A is true")
6 }
7 else {
8   console.log("A is false")
9 }
10 if (x === 5.5) {
11   console.log("B is true")
12 }
13 else {
14   console.log("B is false")
15 }
16 if (x) {
17   console.log("C is true")
18 }
19 else {
20   console.log("C is false")
21 }
22 if (y) {
23   console.log("D is true")
24 }
25 else {
26   console.log("D is false")
27 }
```

---

## Boolean Expressions & Relational Operators

```
1 let isCool, a, b, c, d, e, f, g, h
2 isCool = false
3 a = 7
4 b = -1
5 c = true
6 d = a < b
7 e = a < b || a > 0
8 f = a <= b && a != 0
9 g = !isCool && a == 7
10 h = 0 <= b <= 100 // not what you expect!
```

a	7
b	-1
c	true
d	false
e	true
f	false
g	true
h	true

---

## Strings

```
1 let one, two, num
2 let a, b, c, d, e, f, g, h
3 one = "The quick brown fox"
4 two = ' jumps.'
5 num = two.length // 7
6 console.log(one + two)
7 if (one > "ABC") {
8   console.log("bigger")
9 }
10 a = one[4]
11 b = one.substr(4,7) // 7 chars starting at 4
12 c = one.substr(15) // skip first 15
13 d = one.substr(-2) // keep last 2
14 e = two.replace("jump", "leap")
15 f = two.replace(/\w+/, "xxx")
16 g = two.trim() // remove leading/trailing whitespace
17 h = two.toUpperCase()
```

num	7
a	"q"
b	"quick b"
c	"fox"
d	"ox"
e	" leaps."
f	"xxx."
g	"jumps."
h	"JUMPS."

Expected output:

The quick brown fox jumps.  
bigger

---

## Regular Expressions

```
1 let str, a, b, c, d, e, f, g
2 str = "There are 53 people with SSN 123-12-1234."
3 a = str.search(/fun/)
4 b = str.search(/re/)
5 c = str.search(/[aeiou]/)
6 d = str.search(/\d/)
7 e = str.search(/\d{3}-\d{2}-\d{4}/)
8 f = str.search(/\s/)
9 g = str.search(/^SSN/)
```

a	-1
b	3
c	2
d	10
e	29
f	5
g	-1

---

## Loops

```
1 for (let i = 0; i < 3; i++) {
2   console.log(i)
3 }
4
5 let val = 40
6 while (val > 0) {
7   console.log(val)
8   val -= 10
9 }
10
11 let count = 100
12 do {
13   console.log(count)
14   count += 100
15 } while (count <= 400)
```

Expected output:

0  
1  
2  
40  
30  
20  
10  
100  
200  
300  
400

---

## Arrays

```
1 let a, b, c, val, str
2 a = ["zebra", 12, true, 17.3]
3 a.push("cool");
4 a[0] = 70
5 console.log(a)
6 console.log(a.length)
7 console.log(a[3])
8
9 a.sort();
10 console.log(a)
11 val = a.pop()
12 console.log(val)
13
14 b = [20, 30, 40]
15 for (let elem of b) {
16   console.log(elem)
17 }
18 str= b.join(",");
19 console.log(str);
20 str = "Hi, how are you?"
21 c = str.split(" ")
22 console.log(c)
```

Expected output:

```
5
17.3
[12, 17.3, 70, "cool", true]
true
20
30
40
20,30,40
["Hi,", "how", "are", "you?"]
```

---

## Spread Operator

```
1 // Spread operator on arrays in regular code
2 const nums = [12, 4, 32, 31, 19, 7]
3 console.log(nums)
4 console.log(...nums)
5
6 // Use spread operator to convert array to individual values
  for passing
7 // parameters to functions that require individual values.
8 let ans = Math.min(...nums)
9 console.log("Min is: " + ans)
10
11 // Spread operator on objects in regular code
12 const person = {
13   name: 'Max',
14   age: 29,
15   greet() {
16     console.log('Hi, I am ' + this.name)
17   }
18 }
19
20 // Use spread operator to copy an object (not just its
  reference)
21 const copyOfPerson = {...person}
22 console.log(copyOfPerson)
23
24 // Use of spread operator in parameter list
25 const myMin = (...args) => {
26   let min = args[0]
27   for (let val of args) {
28     if (val < min) {
29       min = val
30     }
31   }
32   return min
33 }
34
35 let ans2 = myMin(6, 20, 9, 34, 3, 22)
36 console.log("Ans: " + ans2)
```

Expected output (to console log):

```
[ 12, 4, 32, 31, 19, 7 ]
12 4 32 31 19 7
Min is: 4
{ name: 'Max', age: 29, greet: [Function] }
Ans: 3
```

---

## Functions

Expected output:

```
1 function showSum(a, b) {
2   let c = a + b
3   console.log(c)
4 }
5 function calcSum(a, b) {
6   let c = a + b
7   return c
8 }
9
10 // To use these functions
11 let x = 7.5
12 showSum(x, 1.0)
13 showSum(x * 2.0, x - 2.0)
14 x = calcSum(1.0, 2.2)
15 console.log(x)
```

8.5  
20.5  
3.2

---

## Arrow Functions

Expected output (in console log):

```
1 // Old fashioned JS notation
2 function doubleIt0(val) {
3   return 2 * val
4 }
5
6 // New-fangled JS notation
7 const doubleIt1 = (val) => {
8   return 2 * val
9 }
10
11 // If the function takes only 1 argument then we can leave off
12 // If the function only has a return statement we can leave
13 // off {}'s and return.
14 const doubleIt2 = val => 2 * val
15 console.log(doubleIt0(14.5))
16 console.log(doubleIt1(14.5))
17 console.log(doubleIt2(14.5))
```

29  
29  
29

---

## Functions as Parameters

```
1 function doubleIt(x) {
2   return 2 * x
3 }
4
5 let a, b, c, d, e, f, g
6 a = [25, 14, 75, 50, 22]
7 b = a.map(doubleIt)
8 console.log(b)
9 c = a.map(function doubleIt(x) {
10   return 2 * x
11 })
12 d = a.map(function (x) {
13   return 2 * x
14 })
15 e = a.map((x) => { return 2 * x })
16 f = a.map((x) => 2 * x )
17 g = a.map(x => 2 * x)
18 a.forEach((val, i) => console.log(i + " --> " +val))
```

Since we expect b, c, d, e, f, and g to end with the same contents we only display the array b.  
Expected output:

```
[50, 28, 150, 100, 44]
0 --> 25
1 --> 14
2 --> 75
3 --> 50
4 --> 22
```

---

## Objects

```
1 let obj, obj2, str
2 obj = { name: "Fred", age: 10 }
3 obj2 = {
4   name: "Alice",
5   age: 12,
6   display() {
7     return this.name + " is " + this.age + " years old"
8   }
9 }
10
11 console.log(obj.name)
12 console.log(obj['age'])
13 obj.name = "George"
14 console.log(obj.name)
15 str = obj2.display()
16 console.log(str)
```

Expected output:

```
Fred
10
George
Alice is 12 years old
```

---

## Classes

Expected output:

Pyro is 3 years old  
Polly is owned by B.Jones

```
1 class Animal {
2   constructor(name, age) {
3     this.name = name
4     this.age = age
5   }
6   display() {
7     console.log(this.name + " is " + this.age + " years old"
8     )
9   }
10 }
11 class Pet extends Animal {
12   constructor(name, age, owner) {
13     super(name, age)
14     this.owner = owner
15   }
16   display() {
17     console.log(this.name + " is owned by " + this.owner);
18   }
19 }
20 let wild, pet
21 wild = new Animal("Pyro", 3)
22 wild.display()
23
24 pet = new Pet("Polly", 10, "B.Jones")
25 pet.display()
```

---

## DOM Intro

The code modifies the screen to set the element whose id is “fun” to have a gray background and a border. Also, its contents are changed to hold “New paragraph.”. Output to the console is:

cool neat

```
1 <html>
2   <head></head>
3   <body>
4     <h1>Greetings</h1>
5     <div id="mydiv">
6       <p>Welcome to this amazing page!</p>
7       <p>I hope you like it.</p>
8     </div>
9     <p id="fun" class="cool neat">This is fun.</p>
10    <script>
11      // Here we use .innerHTML
12      let elem
13      elem = document.getElementById("fun")
14      elem.style.backgroundColor = "#ccc"
15      elem.style.border = "solid 1px black"
16      elem.innerHTML = "New paragraph."
17      console.log(elem.getAttribute("class"))
18
19      // Here we use .insertAdjacentHTML
20      let str = '<p>Great paragraph.</p>';
21      elem = document.getElementById("mydiv")
22      elem.insertAdjacentHTML('afterbegin', str)
23    </script>
24  </body>
25 </html>
```



The even numbered paragraphs will be given a grey background and the console will show these values:

Paragraph 2  
5  
Paragraph 4  
The end  
null

```
1 <html>
2   <head></head>
3   <body>
4     <h1>Greetings</h1>
5     <p>Paragraph 1</p>
6     <p class="even">Paragraph 2</p>
7     <p>Paragraph 3</p>
8     <p class="even">Paragraph 4</p>
9     <p id="last">Paragraph 5 <span>The end</span></p>
10    <script>
11      let elem, arr, elem2
12      elem = document.querySelector(".even")
13      console.log(elem.innerHTML)
14
15      arr = document.querySelectorAll("p")
16      console.log(arr.length)
17
18      // give all even paragraphs a grey background
19      arr = document.querySelectorAll(".even")
20      console.log(arr[1].innerHTML)
21      for (elem of arr) {
22        elem.style.backgroundColor = "#eee"
23      }
24
25      // Alternate way to make a change to all elements
26      // that have been selected using querySelectorAll()
27      [...document.querySelectorAll('.even')].forEach((
28        element) => {
29        element.style.color = 'green';
30      });
31
32      // search inside of an element
33      elem = document.querySelector("#last")
34      elem2 = elem.querySelector("span")
35      console.log(elem2.innerHTML)
36
37      // there is no span in first paragraph
38      elem = document.querySelector("p")
39      elem2 = elem.querySelector("span")
40      console.log(elem2)
41    </script>
42  </body>
43 </html>
```

---

## setInterval() and clearInterval()

```
1 <html>
2   <head></head>
3   <body>
4     <h1>Behold setInterval()</h1>
5     <script>
6       let colorChanger
7       let green = 0
8
9       function changeColor() {
10        green = (green + 1) % 256
11        document.querySelector("h1").style.color = "rgba
12          (0, " + green + ", 0, 1.0)"
13        if (green == 0) {
14          clearInterval(colorChanger)
15        }
16      }
17      colorChanger = setInterval(changeColor, 10)
18    </script>
19  </body>
20 </html>
```

---

## Events and Listeners

```
1 <html>
2   <body>
3     Username: <input type="text" name="username"> <br>
4     Password: <input type="password" name="password"
5               onchange="bewareOfChange()"> <br>
6     <button>Done</button>
7     <script>
8       let textbox, button
9       function bewareOfChange() {
10        alert("A change was made!")
11      }
12      button = document.querySelector('button')
13      button.onclick = () => alert("Done!")
14      textbox = document.querySelector('input[name="
15        username"]')
16      textbox.style.color = "#bbb"
17      textbox.value = "Enter username"
18      textbox.addEventListener("focus", function() {
19        textbox.style.color = "#000"
20        textbox.value = ""
21      })
22    </script>
23  </body>
24 </html>
```

---

## Event Handler for Multiple/Future Objects

```
1 <html>
2   <body>
3     <ul id="grocerylist">
4       <li><input type="text" value="Eggs" /></li>
5       <li><input type="text" value="Bread" /></li>
6       <li><input type="text" value="Milk" /></li>
7       <li><input type="text" value="Butter" /></li>
8       <li><input type="text" value="Cheese" /></li>
9     </ul>
10
11     <script>
12       var list = document.getElementById("grocerylist")
13       list.addEventListener("input", function (event) {
14         if (event.target.matches('input[type="text"]')) {
15           // call validation function and pass
16           // event.target as parameter
17         }
18       });
19     </script>
20   </body>
21 </html>
```

---

## Client-Side Form Validation

```
1 <html>
2   <body>
3     <form method="post" action="passed.html"
4       onsubmit="return formIsValid()">
5       Name: <input type="text" name="name">
6       <span>Name must have at least 1 character</span><br>
7       SSN: <input type="text" name="ssn">
8       <span>SSN must be ###-##-####</span><br>
9       <input type="submit" value="Done">
10    </form>
11
12    <script>
13      for (let elem of document.querySelectorAll('span')) {
14        elem.style.display="none"
15      }
16      function formIsValid() {
17        let name, ssn, valid
18        valid = true
19        name = document.querySelector('input[name="name"]')
20        if (name.value.search(/^s*$/) == 0) {
21          name.nextElementSibling.style.display="inline"
22          valid = false
23        }
24        else {
25          name.nextElementSibling.style.display="none"
26        }
27        ssn = document.querySelector('input[name="ssn"]')
28        if (ssn.value.search(/^\d{3}-\d{2}-\d{4}$/) != 0) {
29          ssn.nextElementSibling.style.display="inline"
30          valid = false
31        }
32        else {
33          ssn.nextElementSibling.style.display="none"
34        }
35        return valid
36      }
37    </script>
38  </body>
39 </html>
```

Expected output is that the even-numbered paragraphs will have a grey background and odd-numbered paragraphs will have a green background.

```
1 <html>
2   <head>
3     <script src="https://cdnjs.cloudflare.com/ajax/libs/
4       jquery/3.4.1/jquery.min.js"
5       integrity="sha256-
6         CSXorXvZcTkaix6Yvo6HppcZGetbYMGWSF1Bw8HfCJo="
7       crossorigin="anonymous"></script>
8   </head>
9   <body>
10    <h1>Greetings</h1>
11    <p>Paragraph 1</p>
12    <p class="even">Paragraph 2</p>
13    <p>Paragraph 3</p>
14    <p class="even">Paragraph 4</p>
15    <p id="last">Paragraph 5 <span>The end</span></p>
16    <script>
17      $(function() {
18        // the jQuery way
19        $("p:not(.even)").css("background-color", "green")
20        // the Javascript way
21        let arr = document.querySelectorAll(".even")
22        for (let elem of arr) {
23          elem.style.backgroundColor = "#ccc"
24        }
25      })
26    </script>
27  </body>
28 </html>
```

---

## jQuery Events, Slides, and Fades

```
1 <div id="buttons">
2   <button id="slide-button">Slide</button>
3   <button>Fade</button>
4 </div>
5 <h1>Greetings!</h1>
6 <script>
7   $("#slide-button").on("click", function() {
8     let greeting = $("h1")
9     if (greeting.is(":visible")) {
10      greeting.slideUp()
11    }
12    else {
13      greeting.slideDown()
14    }
15  })
16  $("#buttons").on("click", "button:last", function() {
17    $("h1").fadeToggle(2000) // animation speed = 2000 ms
18  })
19  $("button").on("mouseenter", function () {
20    $(this).css("background-color", "green")
21  })
22  $("button").on("mouseleave", function () {
23    $(this).css("background-color", "buttonface")
24  })
25 </script>
```

Expected output to console log  
(assuming user has not changed  
initial settings):

```
SSN: 123-12-1234
Password: hello
First option: true
Last option: false
First radio: false
Second radio: false
First checkbox: false
Last checkbox: true
Button: Click Me
Submit: No Me!
Color: Choose One ...
Pick: no
```

```
1 SSN: <input type="text" name="ssn" value="123-12-1234"><br>
2 Password: <input type="password" name="passcode" value="hello"
   ><br>
3 Favorite:
4   <select name="color">
5     <option value="0">Choose One ...</option>
6     <option value="1">Red</option>
7     <option value="2">Green</option>
8     <option value="3">Blue</option>
9   </select><br>
10 Pick 1:
11   Yes <input type="radio" name="choose" value="yes">
12   No  <input type="radio" name="choose" value="no" checked="
      checked">
13   Maybe <input type="radio" name="choose" value="maybe"><br>
14 Send spam: <input type="checkbox" name="spam"><br>
15 Send money: <input type="checkbox" name="money" checked="
      checked"><br>
16 <button>Click Me</button>
17 <input type="submit" value="No Me!">
18 <script>
19   console.log("SSN: " + $("input[name='ssn']").val())
20   console.log("Password: " + $("input[name='passcode']").val()
   )
21   console.log("First option: " + $("option:first").prop("
      selected"))
22   console.log("Last option: " + $("option:last").prop("
      selected"))
23   console.log("First radio: " + $(":radio:first").prop("
      checked"))
24   console.log("Second radio: " + $(":radio:nth(1)").prop("
      checked"))
25   console.log("First checkbox: " + $(":checkbox:first").prop("
      checked"))
26   console.log("Last checkbox: " + $(":checkbox:last").prop("
      checked"))
27   console.log("Button: " + $("button").text())
28   console.log("Submit: " + $("input[type='submit']").val())
29   console.log("Color: " + $("select option:selected").text())
30   console.log("Pick: " + $(":radio:checked").val())
31 </script>
```

The behavior of the code is a bit weird. Each time the button is clicked a different element on the page takes on the `clickStyle` class. The first click highlights the button; second click highlights the first aside, the third click highlights the second aside. This continues in a circular fashion.

```

1 <html>
2   <head>
3     <script src="https://cdnjs.cloudflare.com/ajax/libs/
4       jquery/3.4.1/jquery.min.js"
5       integrity="sha256-
6         CSXorXvZcTkaix6Yvo6HppcZGetbYMGWSF1Bw8HfCJo="
7         crossorigin="anonymous"></script>
8     <style>
9       #panels {
10        display: flex;
11        flex-wrap: wrap;
12        width: 400px;
13      }
14      aside {
15        width: 46%;
16        background-color: #ccc;
17        border: 1px solid;
18        margin-right: 2%;
19        margin-bottom: 10px
20      }
21      button {
22        width: 100%;
23      }
24      .clickStyle {
25        background-color: pink;
26        text-align: center;
27        color: green;
28        border: none;
29      }
30    </style>
31  </head>
32  <body>
33    <div id="panels">
34      <aside>
35        <h1>Welcome!</h1>
36        <p>This is fun.</p>
37      </aside>
38      <aside>
39        <h1>Salutations!</h1>
40        <p>This is also fun.</p>
41      </aside>
42      <button>Please Do Click Me</button>
43    </div>
44    <script>
45      let count = 0
46      $("button").on("click", function() {
47        if (count % 3 == 0) {
48          $(this).addClass("clickStyle")
49          $("aside").removeClass("clickStyle")
50        } else if (count % 3 == 1) {
51          $("button").removeClass("clickStyle")
52          $("aside:first").toggleClass("clickStyle")
53        } else {
54          $("aside").toggleClass("clickStyle")
55        }
56        count++
57      })
58    </script>
59  </body>
60 </html>

```