



Guía de ejercicios 2 - Condiciones y operadores lógicos



¡Hola! Te damos la bienvenida a esta nueva guía de ejercicios.

¿En qué consiste esta guía?

En la siguiente guía podrás trabajar los siguientes aprendizajes:

- Introducción a comparaciones con if.
- Ejecución de códigos si se cumple la función.
- Anatomía de un if.
- Uso de condiciones en el código.
- Operadores de comparación.
- Operadores lógicos.
- Manipulación con DOM.
- Condiciones else if.
- if - else if - else.
- Múltiples if vs else if.

¡Vamos con todo!





Tabla de contenidos

Introducción a la instrucción if	3
Anatomía de una condición	3
Actividad 1: Hace calor	4
Operadores de comparación	5
Combinando la instrucción if y manipulación del DOM	6
En caso contrario	7
Ejemplo if else con DOM	8
Actividad 2: Torta para todos con if/else en DOM	8
Else if	9
Actividad 3: Probando el else if	9
Actividad 4: Determinando el tamaño del escritorio.	10
Actividad 5: Múltiples ifs vs else if.	11
Actividad 6: Índice de Masa Corporal (IMC)	11
Comparaciones y tipos de datos	12
Operadores Lógicos	12
Probando las comparaciones en el inspector de elementos	13
Probando el operador "y" (&&)	13
Probando el operador "o" ()	14
Probando con el operador "no" (!)	14
Ejercicio con el operador &&	15
Actividad 7: Solicitud de email y password	15
Ejercicio con el operador	15
Operadores lógicos y checkbox	16
Actividad 8: Practica el uso del DOM	17
Sintaxis alternativas de la instrucción if	17
if inline	17
Operadores ternarios	18
Actividad 9: Transformación de condicionales	19
Otro uso para el operador	19
Preguntas de entrevista laboral	20



¡Comencemos!



Introducción a la instrucción if

La instrucción if nos permite ejecutar código en caso de que alguna variable sea menor, igual o mayor a un valor determinado, en otras palabras en caso de que se cumpla una condición.

Veamos un ejemplo:

```
edad = 14

if (edad < 18) {
  console.log("Lo siento, No puede pasar al club 🚫 ")
}
```

Este código lo podemos probar copiando y pegando en el tab "Console" del inspector de elementos, o creando un script como aprendimos en la unidad anterior.

En este ejemplo la condición se lee como "si la edad es menor a 18...", entonces mostrar "no puedes entrar al club". El valor también lo puede ingresar el usuario a través de un input o un prompt. Veamos un ejemplo sencillo con prompt.

```
edad = prompt("Ingresa tu edad")

if (edad < 18) {
  console.log("Lo siento, No puede pasar al club 🚫 ");
}
```

Anatomía de una condición

Una condición contiene las siguientes características:

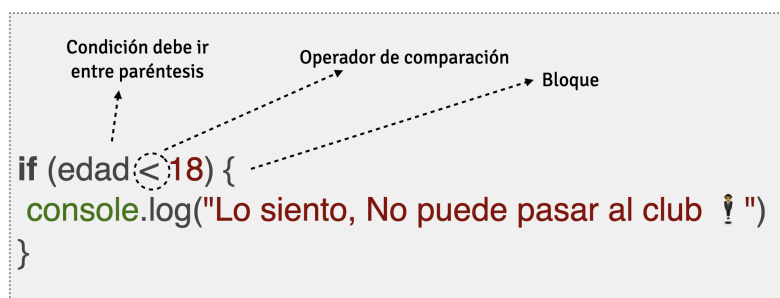


Imagen 1. Anatomía de una condición
Fuente: Desafío Latam.



Acompañado al `if` hay un bloque, que se observa en la imagen anterior, y corresponde al código que está entre las llaves `{}`.

Los bloques nos permiten agrupar instrucciones, y de esta forma ejecutarlas todas en caso de que se cumpla el `if`.

```
edad = 4

if (edad < 18) {
  console.log("1") // Instrucción 1
  console.log("2") // Instrucción 2
  console.log("3") // Instrucción 3
  a = 2 + 2 // Instrucción 4
  console.log(a) // Instrucción 5
}
```



Observamos que si se cambia el valor de `edad` a 20, no se ejecuta ninguna instrucción.

El código anterior se lee como:

1. Si `edad` es menor de 18 entonces
2.
 - a. Ejecutar instrucción 1
 - b. Ejecutar instrucción 2
 - c. Ejecutar instrucción 3
 - d. Ejecutar instrucción 4
 - e. Ejecutar instrucción 5



Actividad 1: Hace calor

Pongamos en práctica lo aprendido con un ejercicio donde debemos mostrar un mensaje dependiendo si se cumple una condición.

1. Crea una variable *temperatura* que almacene un número entero.
2. Agrega la instrucción `if` con la condición de si la variable es mayor a 30.
3. Dentro del bloque debe mostrar un mensaje con `console.log` o `alert` que diga: "Ya está haciendo calor 🥵".



Operadores de comparación

Existen varios operadores de comparación que nos sirven para diferentes situaciones.

Símbolo	Nombre
==	Igual
!=	Distinto
>	Mayor
<	Menor
>=	Mayor o igual
<=	Menor o igual

Tabla 1. Operadores de comparación
Fuente: Desafío Latam

Existen varios operadores para comparar, hasta el momento solo hemos ocupado el > y ahora aprenderemos a ocupar el operador == que nos sirve para comparar que dos valores sean iguales.

Veamos un ejemplo donde crearemos un pequeño script donde el usuario debe introducir un password y solo, si introduce el password correcto, se muestra un mensaje.

```
password = 'secreto'
if (password == 'secreto'){
  alert('Conoces el password')
}
```

El operador == nos permite comparar dos valores, si ambos valores son iguales la expresión se resolverá como cierto (true) y si los valores son distintos se resolverá como falso (false).

```
2 == 3 // false
'hola' == 'hola' // true
a = 'password'
b = 'p231mfas'
a == b // false
```

Cada operador de comparación resuelve de forma similar.



```
3 > 5 // false
5 > 3 // true
a = 5 // El igual no es un comparador, estamos asignando el valor 5
b = 8 // Asignamos el valor 8 a la variable b
a < b // true
```

Incluso podemos asignar cualquiera de esas expresiones directamente a una variable.

```
a = 3 > 5
alert(a) // false
```

Combinando la instrucción if y manipulación del DOM

Ahora veamos qué sucede cuando combinamos la instrucción if y la manipulación del DOM:

```
<p id="secret-result"></p>
<input id="input" type="text">
<button onclick="
  value = document.querySelector('#input').value
  if (value == 'password-secreto'){
    par = document.querySelector('#secret-result')
    par.innerHTML = 'lo lograste'
  }
"> Verificar </button>
```

1. En el código anterior tomamos el valor del input.
2. Si el valor del input es igual a 'password-secreto' Entonces
 - a. Buscamos el párrafo con id #secret-result
 - b. Modificamos el contenido interno.



En caso contrario

Es posible que cuando creamos una instrucción if también queremos ejecutar algún código específicamente si no cumple la condición, para esto nos sirve la instrucción else. Veamos un ejemplo:

```
password = 'secreto'
if (password == 'secreto'){
  alert('Conoces el password')
}
else {
  alert('No conoces el password')
}
```

Veamos el mismo ejercicio de adivinar el password de if/else en conjunto con modificar el DOM.

```
<p id="secret-result"></p>
<input id="input" type="text">
<button onclick="
  value = document.querySelector('#input').value
  if (value == 'password-secreto'){
    par = document.querySelector('#secret-result')
    par.innerHTML = 'lo lograste'
  }
  else {
    par = document.querySelector('#secret-result')
    par.innerHTML = 'No lo lograste'
  }
"> Verificar </button>
```

El código anterior se lee como:

- Cuando se haga clic en el botón:
 1. Tomamos el valor del input llamado input
 2. Si el valor es password-secreto entonces:
 - a. Seleccionamos el elemento con id secret-result
 - b. Modificamos el contenido de ese elemento a 'lo lograste'



3. En caso contrario (o sea el password es cualquier otro valor distinto a password-secreto)
 - a. Seleccionamos el elemento con id secret-result
 - b. Modificamos el contenido de ese elemento a 'lo lograste'

Ejemplo if else con DOM

Queremos hacer una página web donde se ingrese la cantidad de personas en un cumpleaños y la cantidad de pedazos de torta, la aplicación realizará el cálculo para determinar si alcanza al menos un pedazo para cada uno.

```
pedazosDeTorta = 8
invitados = 4

cantidadDePedazosPorInvitado = pedazosDeTorta / invitados

if (cantidadDePedazosPorInvitado >= 1) {
  alert("Hay suficientes pedazos de torta para todos 🎉");
} else {
  alert("Hay más invitados que pedazos de torta 😞");
}
```



Actividad 2: Torta para todos con if/else en DOM

A partir del ejercicio anterior se debe crear una página web con dos inputs y un botón para realizar el cálculo.

1. Crear el index.html con la base de código
2. Agregar los inputs
3. Agregar el botón
4. Agregar el script sobre el botón en el evento click
 - a. Tomar los valores de los inputs
 - b. Realizar la división y guardarla en una variable
 - c. Si el resultado es mayor o igual a 1 mostrar en el mensaje de que alcanza
 - d. En caso contrario mostrar que no alcanza.



Else if

Es posible evaluar múltiples situaciones relacionadas, algo así como:

- Si pasa esto haz x.
- Si no pasa lo anterior, pero pasa esto otro, entonces haz y.
- Si no pasa lo anterior, pero pasa esto otro, entonces haz z.
- En cualquier otro caso, haz esto otro.

Para lograr esto ocuparemos la instrucción else if.

```
if (false) {  
  console.log("Este mensaje no se mostrará");  
} else if (false) {  
  console.log("Este mensaje tampoco");  
} else if (true) {  
  console.log("Este mensaje SI se mostrará 🙌😄");  
} else {  
  console.log("A mi no me verás 😞😞");  
}
```



Actividad 3: Probando el else if

Prueba el código anterior cambiando las condiciones, intenta adivinar la respuesta correcta previo a intentarlo.

1. ¿Qué se muestra si todas las expresiones son falsas?
2. ¿Qué se muestra si todas las expresiones son verdaderas?
3. ¿Qué sucede si hay múltiples verdaderas?



Actividad 4: Determinando el tamaño del escritorio.

Nos piden crear una página web sencilla con un input y un botón. El input es para que una persona ingrese el ancho de un viewport en pixeles. (Por ejemplo: 720).

Al presionar el botón, se mostrará un mensaje indicando si la pantalla es xs, sm, md, o xl.

Para hacer el cálculo se deben ocupar las siguientes medidas:

- Si el número es menor a 576 mostrar xs
- Desde 576 a 768 mostrar sm
- Desde 768 a 962 mostrar md
- Mayor a 962 mostrar xl

Partiremos creando un script que podamos probar en el inspector de elementos.

```
a = 758
if (a < 576){
  console.log("xs")
}
else if(a < 768){
  console.log("sm")
}
else if(a < 962){
  console.log("md")
}
else{
  console.log("xl")
}
```

1. Probemos cambiando el valor dentro de a y probando justo en los puntos de comparación, 576, 768, 962. ¿Cumple las expectativas? Prueba con valores decimales, ejemplo 575.99 ¿Qué sucede?
2. Modifica el código anterior para que funcione dentro de una página web con un input y un botón.
3. Agrega la capacidad de diferenciar tamaños xl de xxl, sobre 1200 sería xxl.



Actividad 5: Múltiples ifs vs else if.

Compara los siguientes códigos, ejecuta ambos y observa si hay alguna diferencia, prueba cambiando el valor de la variable a.

```
a = 950
if (a < 576){
  console.log("xs")
}
else if(a < 768){
  console.log("sm")
}
else if(a < 962){
  console.log("md")
}
else{
  console.log("xl")
}
```

```
a = 950
if (a < 576){
  console.log("xs")
}
if(a < 768){
  console.log("sm")
}
if(a < 962){
  console.log("md")
}
else{
  console.log("xl")
}
```



Actividad 6: Índice de Masa Corporal (IMC)

Realicemos el siguiente ejercicio en donde se calculará el IMC de una persona. Para esto, será necesario que el usuario ingrese dos valores: la altura en metros y el peso en kilos.

Luego, se utilizará la siguiente fórmula para obtener el resultado:

$$IMC = \frac{peso}{(altura * altura)}$$

Para esta actividad:

1. Crea la variable peso y asigna el valor 80.
2. Crea la variable altura y asigna el valor 1.70 .
3. Si el IMC está entre 18.5 y 24.9, entonces mostrar con console.log "Tiene un IMC normal".
4. Crea un condicional que evalúe si el IMC es mayor a 25, entonces imprimir por consola: "Tiene un IMC alto".



5. Crea un condicional que evalúe si el IMC es menor a 18.5, entonces imprimir por consola: "Tiene un IMC bajo".



Tip: Ordena las condiciones de menor a mayor como en el ejercicio anterior.

Comparaciones y tipos de datos

Símbolo	Nombre
===	Estrictamente Igual
!==	Estrictamente Distinto

Tabla 2: Símbolos de comparaciones

Fuente: Desafío Latam.

El operador `===` nos permite comparar tanto el dato como el tipo de dato. Por ejemplo, si comparamos números con strings siempre obtendremos resultado falso, puesto que aunque ambos tengan el mismo contenido, tienen distintos tipos de datos.

```
console.log("10" == 10) // true
console.log("10" === 10) // false
console.log("10" != 10) // false
console.log("10" !== 10) // true
```



Se considera una buena práctica comparar utilizando el operador estrictamente igual o estrictamente distinto debido a que puede ayudar a prevenir errores.

Operadores Lógicos

Así como los operadores aritméticos nos permiten resolver expresiones como `3 + 3`, los operadores lógicos nos permiten resolver expresiones lógicas como cierto y falso (`true` y `false`). Gracias a este tipo de operaciones podemos validar que un usuario y su password corresponda. Por ejemplo, podríamos pedir que el `input1` sea igual a juan@desafiolatam.com y que el `input2` sea igual a `password-secreto` y verificar que



ambas expresiones sean ciertas en conjunto, o sea que el usuario haya correctamente ingresado ambas.

Dentro de JavaScript, existen 3 operadores lógicos que debemos manejar. Y que se explican en la tabla de la siguiente página.

Nombre	Operador	¿Cómo se interpreta?
y (and)	&&	Si ambas condiciones son ciertas se resuelve como true . Si alguna de ellas es falsa se resuelve como false
o (or)		Si alguna de las expresiones es cierta se resuelve como true . si ambas son falsas se resuelve como false
no (not)	!	Invierte la expresión. No true es false , no false es true .

Tabla 3: Operadores Lógicos.
Fuente: Desafío Latam.

Probando las comparaciones en el inspector de elementos

Para probar las comparaciones, abre el inspector de elementos y escribe las combinaciones en el tab console utilizando true y false, como se muestra en la siguiente imagen.

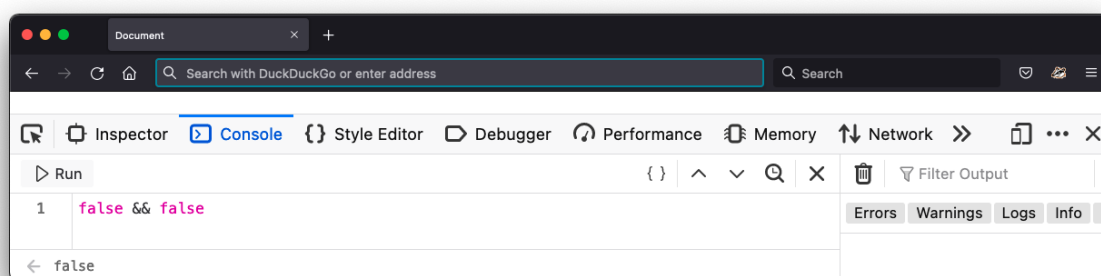


Imagen 2. Probando los operadores de comparación
Fuente: Desafío Latam

En la sección inferior del navegador podremos ver como se resuelve la expresión, en esta imagen vemos que `false && false` se resuelve como false. A continuación veremos cada uno de los operadores.

Probando el operador "y" (&&)



El operador lógico `&&` se resuelve como `true` si ambas condiciones son ciertas. Prueba en el navegador las siguientes expresiones y observa los resultados:

1. `true && true`
2. `true && false`
3. `false && true`
4. `false && false`

Lo anterior tiene mucho sentido considerando que la interpretación lógica que revisamos del operador “y” es:

Si ambas condiciones son ciertas se resuelve como **true**.
Si alguna de ellas es falsa se resuelve como **false**.

Probando el operador “o” (||)

El operador lógico `||` encuentra el primer valor `true`. Prueba las siguientes expresiones en la consola del inspector de elementos:

1. `true || true`
2. `true || false`
3. `false || true`
4. `false || false`

Lo anterior tiene sentido considerando que la interpretación lógica que revisamos del operador o es:

Si alguna de las expresiones es cierta se resuelve como **true**. Si ambas son falsas se resuelve como **false**

Probando con el operador “no” (!)

El operador acepta un solo argumento y retorna el valor contrario.

Prueba las siguientes expresiones:

1. `!true`
2. `!false`



Invierte la expresión. No true es **false**, no false es **true**.

Ejercicio con el operador &&

Creemos un script para pedir el input y password a un usuario:

```
a = prompt("email")
b = prompt("password")
if (email == "juan@desafiolatam.com" && password == "password-secreto")
{
    alert("Usuario correcto");
} else {
    alert("Usuario incorrecto");
}
```



Actividad 7: Solicitud de email y password

Según lo practicado anteriormente con el operador &&, crea una página web que tenga ambos inputs uno para pedir el email y otro para el password, un botón para verificar el resultado y un párrafo donde mostrar el resultado.

Ejercicio con el operador ||

En un formulario se tienen dos inputs. Uno para ingresar el salario actual y otro para ingresar el salario anterior. Nos piden que al menos uno de los dos debe ser llenado.

Primero agregamos todos los elementos a la página web, en este caso dos inputs y un botón. Adicionalmente agregaremos un texto al lado de cada input utilizando label y dentro del atributo for debemos indicar el nombre del input. Esto es para que al hacer clic en el texto nos lleve automáticamente al input.

Luego al hacer clic en el botón revisar, tomaremos los valores de cada input y verificaremos que no estén vacíos, como utilizamos el operador o, entonces si el primero es distinto a vacío (**true**) y el segundo es vacío (**false**) la expresión se resolverá como **true** puesto que **true || false** se resuelve como true.

```
<label for="salario"> Salario </label>
<input name="salario" id="salario">
```



```
<label for="salario-anterior"> Salario Anterior </label>
<input name="salario-anterior" id="salario-anterior">

<button onclick="
  value1 = document.querySelector('#salario').value
  value2 = document.querySelector('#salario-anterior').value
  if (value1 != '' || value2 != ''){
    alert('Correcto')
  } else {
    alert('Incorrecto')
  }
"> Revisar </button>
```

Operadores lógicos y checkbox

En HTML para hacer un checkbox tenemos que utilizar la etiqueta input con el type checkbox. Adicionalmente, agregaremos el texto al lado utilizando label y dentro del atributo **for** debemos indicar el nombre del input. Esto es para que al hacer clic en el texto sea equivalente a hacer click dentro del checkbox. Es una buena práctica de usabilidad.

```
<label for="opt1"> Opción 1</label>
<input type="checkbox" name="opt1" id="opt1">
```

Para revisar si el checkbox está seleccionado debemos seleccionar el elemento y utilizar `.checked`

```
document.querySelector('#opt1').checked
```

Prueba en una página web nueva escribir el html con el label e input. Luego abre la página con el navegador, marca la caja, abre el inspector de elementos y en el tab console escribe: `document.querySelector('#opt1').checked`, deberías obtener `true`. Luego desmarca la caja y ejecuta el código nuevamente, deberías obtener `false`.

Ahora que sabemos trabajar con checkboxes veamos un ejemplo sencillo con dos checkboxes y que sea necesario que ambos estén seleccionados para mostrar un mensaje en específico.

```
<button onclick="
  opt1 = document.querySelector('#opt1').checked
  opt2 = document.querySelector('#opt2').checked
```




```
if (opt1 == true && opt2 == true) {  
    alert('Los dos elementos están marcados')  
}  
else {  
    alert('Los dos elementos *NO* están marcados')  
}  
"> Revisar </button>
```



Actividad 8: Practica el uso del DOM

A continuación seguiremos practicando la manipulación del DOM con JavaScript, utilizando el ejemplo anterior.

1. Para ello, deberás mostrar un párrafo de HTML en vez de una alerta con JavaScript.
2. Ahora creemos un nuevo botón con el texto "Enviar", y que el estado del mismo se encuentre deshabilitado, esto se logra desde el HTML agregando el atributo `disabled=true`. La idea es que a través de un identificador único denominado `"btn-enviar"`, manipulemos desde JavaScript el estado del atributo `disabled` y lo llevemos a `false`. Esto lo puedes lograr de la siguiente manera `elemento_seleccionado.disabled = false`



Un botón desactivado se visualiza con un color más opaco y además no permite hacer click sobre él.

3. Crea el código desde cero, haciendo que para esta ocasión el botón de "Enviar" se habilite a partir de las siguientes condiciones:
 - a. Al presionar el botón "Revisar".
 - b. Que al menos uno de los dos checkboxes esté marcado, pueden ser ambos.

Sintaxis alternativas de la instrucción if

Hasta ahora hemos aprendido que la forma tradicional de escribir `if` es la acompañada de la condición y luego el bloque dentro de llaves `{}`. Existen otras formas alternativas para usar `if`.

if inline

Si necesitamos ejecutar una única instrucción en función de si se cumple una condición podemos ocupar.

```
if("100" === 100) alert("Son estrictamente iguales")
```



```
else alert("No son estrictamente iguales")
```

De esta manera podemos escribir al lado de la declaración del `if` y del `else` la instrucción que queremos ejecutar.

Operadores ternarios

En JavaScript también tenemos los operadores ternarios, que se expresan con el signo de interrogación (?); que representa el bloque `if`, y los dos puntos (:); que representa el bloque `else`.

Por ejemplo:

```
if( 1 + 1 === "2"){  
  console.log("Son igualitos")  
}else {  
  console.log("De nuevo, no es lo mismo un Number que un String")  
}
```

Se puede escribir de la siguiente manera:

```
1 + 1 === "2"  
? console.log("Son igualitos")  
: console.log("De nuevo, no es lo mismo un Number que un String");
```

En donde:

- Azul: Representa la expresión
- Rojo: Representa el bloque `if`
- Magenta: Representa el bloque `else`



En los operadores ternarios es obligatorio definir un bloque `else`.



Actividad 9: Transformación de condicionales

Basado en el siguiente script en donde se utiliza un condicional para validar si la cantidad de cuotas en una compra es mayor o igual a 6, se aplicarán intereses. Reescribe el condicional usando operador ternario.

```
cuotas = 5

if(cuotas >= 6){
  console.log("Serán aplicados intereses en esta compra")
}else {
  console.log("No serán aplicados intereses en esta compra")
}
```

Otro uso para el operador ||

Hay otro uso interesante del operador || sin necesidad del if

A veces queremos seleccionar un valor que ingresó el usuario pero si no ingresó ninguno, necesitaremos utilizar un valor por defecto. Esto lo podemos escribir como:

```
variable = valor_ingresado || valor_por_defecto
```

Por ejemplo si intentamos lo siguiente:

```
variable = "" || "hola" . Nuestra variable guardará el valor "hola".
```

Luego podemos llevar esta misma idea a tomar el valor de un input o un valor por defecto:

```
input = document.querySelector("#input-1").value || 10
console.log(10)
```

Para probarlo necesitas en el html un input con id "input-1", prueba primero ejecutando esta línea en el inspector de elementos cuando no haya ningún valor ingresado en el input y luego ingresa un valor y ejecuta la línea nuevamente.



Preguntas de entrevista laboral

Intenta contestar con tus palabras las siguientes preguntas, anota las respuestas y luego revisa en la guía las definiciones para evaluar si las aprendiste correctamente.

1. ¿Cuál es la diferencia entre el operador `==` y el operador `===`?
2. ¿Qué se obtiene al comparar `2 === '2'`?
3. ¿Qué muestra el siguiente código?

```
x = 5;  
y = x > 3 ? "El número es mayor a 3" : "El número es menor o igual a 3";  
console.log(y);
```