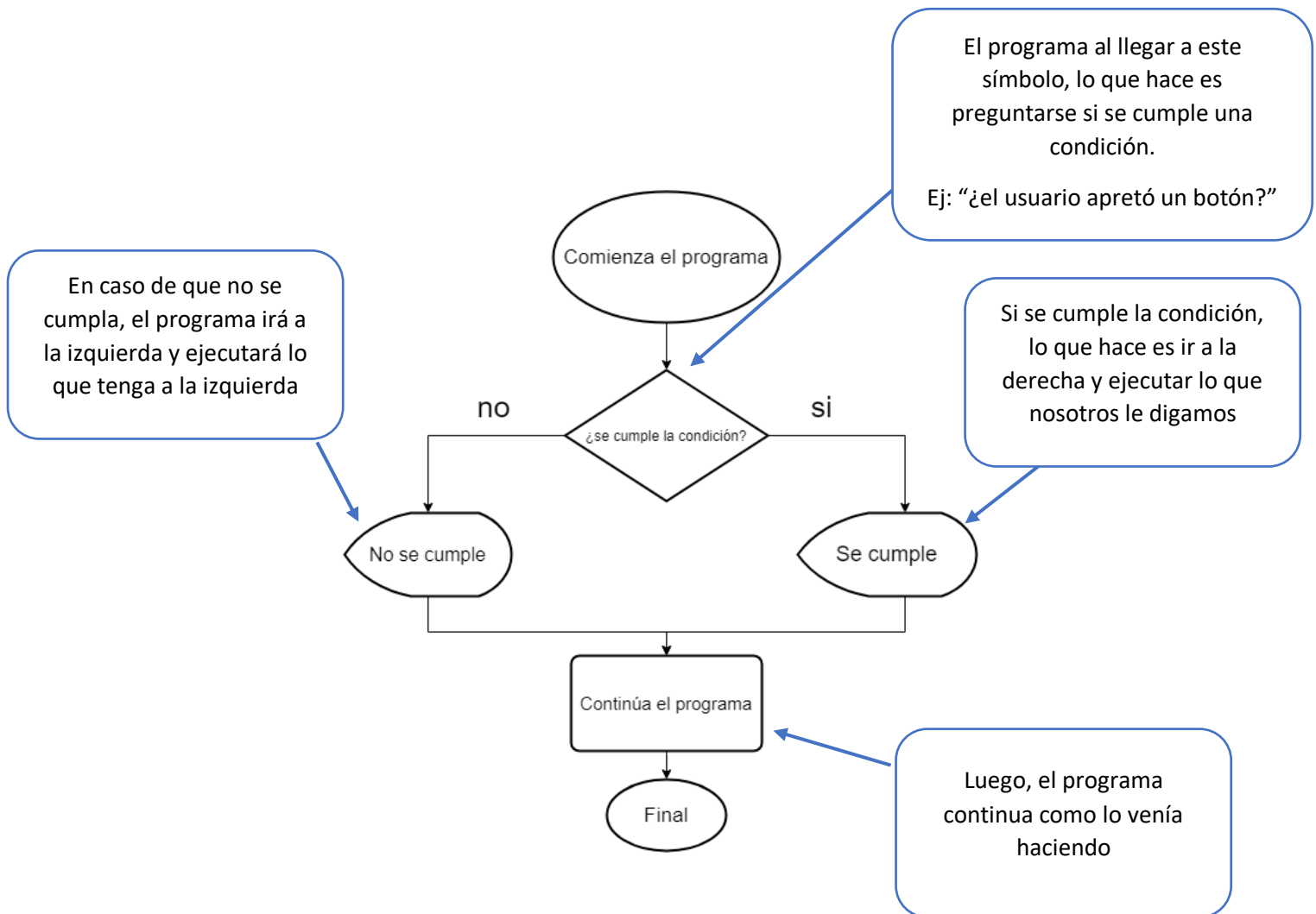


Condicionales

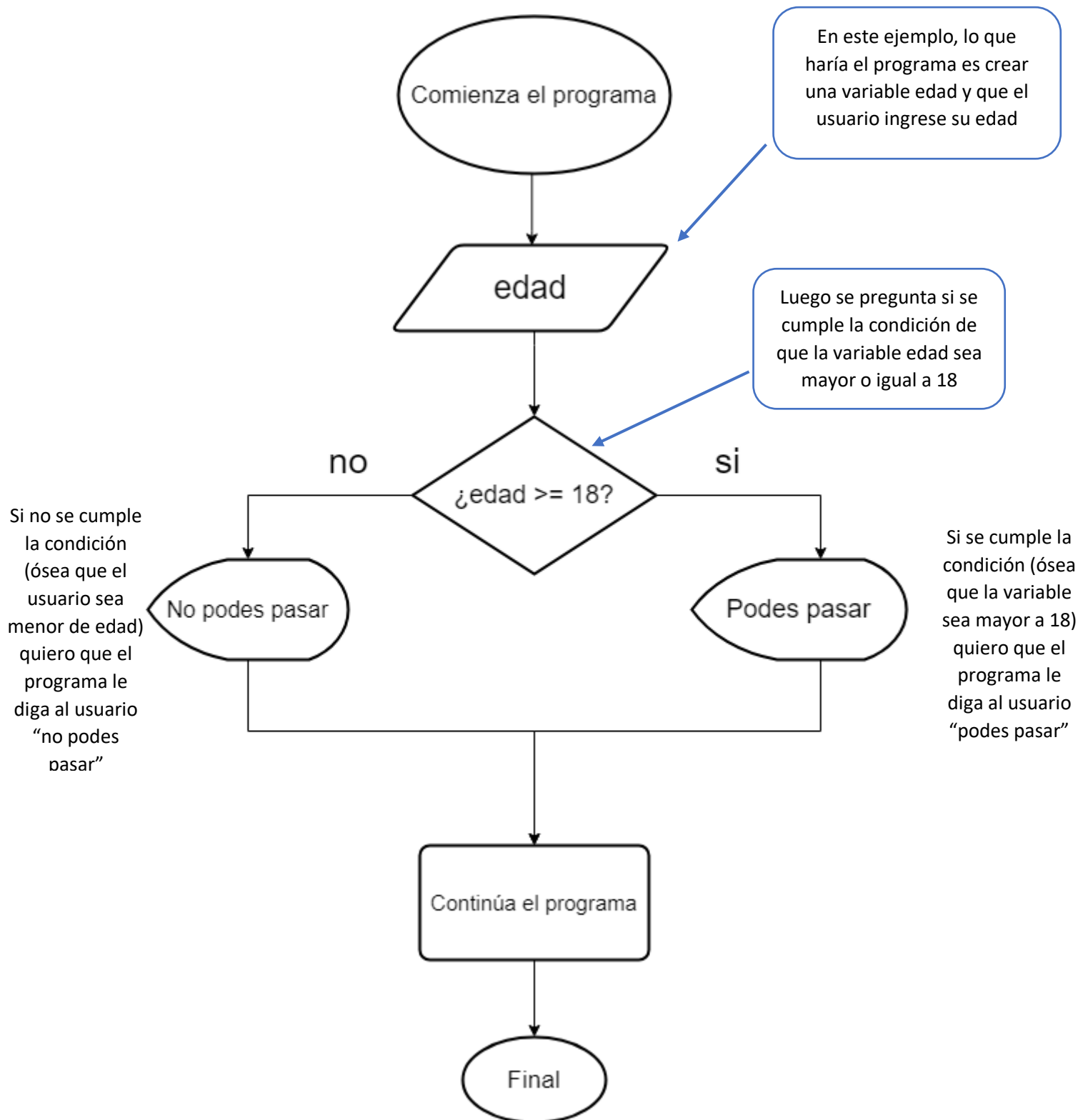
Sirven para que el programa tome decisiones. En base a una condición el programa decide qué bloque de código va a ejecutar.

Diagrama de una condicional if

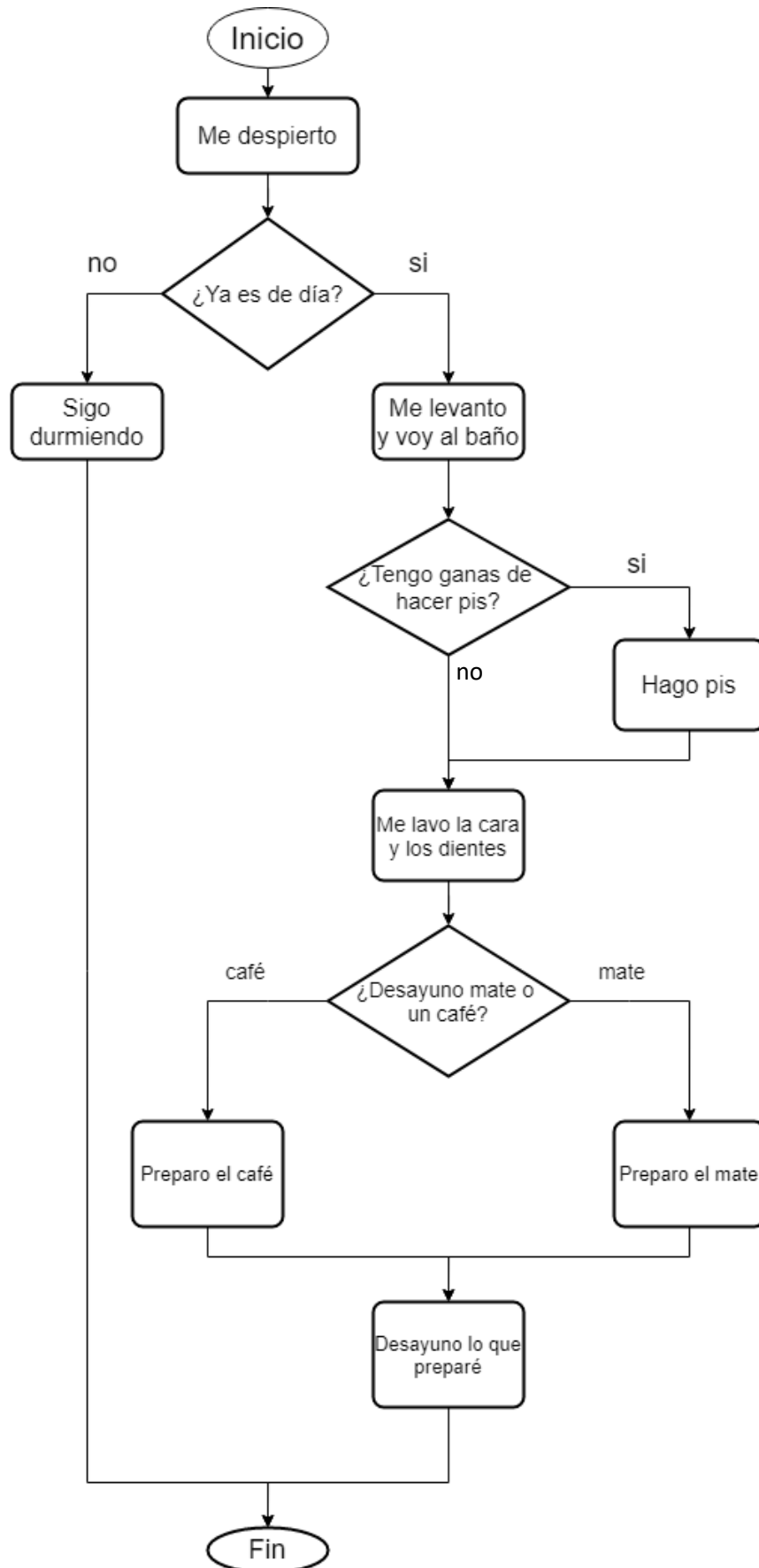
Lo siguiente es un diagrama de flujo para entender el funcionamiento de una toma de decisiones en un programa.



Ahora veamos un ejemplo más demostrativo



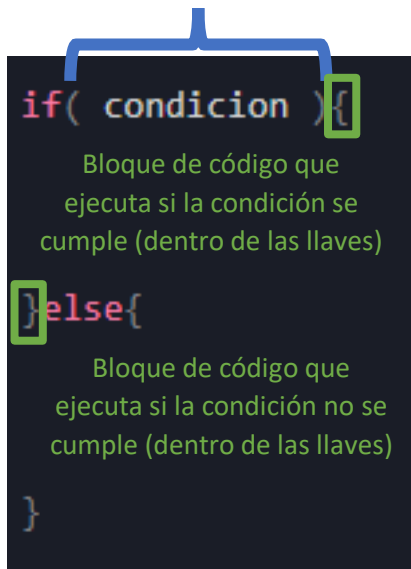
Otro ejemplo: Supongamos que hacemos el diagrama de una rutina a la mañana, desde despertarme, decidir si levantarme o seguir durmiendo, en caso de levantarme decidir si tengo ganas de hacer pis o directamente me lavo la cara y cuando termine, decidir qué desayunar. cuando termine, decidir qué desayunar.



Sintaxis del if en JavaScript

La **condición** a evaluar va dentro de unos **paréntesis**.

La condición puede ser una comparación entre variables por ejemplo (si una es mayor a otra, o si son iguales) o puede ser comprobar que una variable tenga cierto valor



```
if( condicion ){  
    Bloque de código que  
    ejecuta si la condición se  
    cumple (dentro de las llaves)  
} else {  
    Bloque de código que  
    ejecuta si la condición no se  
    cumple (dentro de las llaves)  
}
```

The diagram shows the syntax of an if-else statement. A blue bracket highlights the condition 'condicion' inside the parentheses of the 'if' statement. A green box highlights the opening curly brace '{' of the first code block. Another green box highlights the 'else' keyword. A final green box highlights the closing curly brace '}' of the second code block. Green text annotations explain that the code blocks execute when the condition is met or not met.

If -> si

Else -> sino

Lo que el programa piensa:

If (si) ocurre esto, ejecuto este código,
else (si no se cumple) ejecuto este otro código

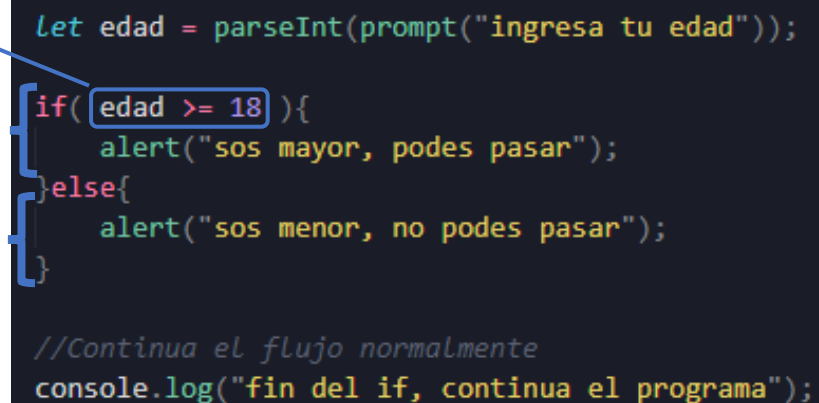
Ejemplos

Supongamos el ejemplo de que ingreso mi edad por teclado, eso se guarda en una variable y el programa decide si dejarme pasar (si soy mayor de edad) o no dejarme pasar (si soy menor de edad).

En este caso, la condición es una comprobación de si es mayor o igual a 18

En caso de que se cumpla la condición, ejecuta el primer bloque

En caso de que no se cumpla la condición, ejecuta el segundo bloque (ejecuta el else)



```
let edad = parseInt(prompt("ingresa tu edad"));  
  
if( edad >= 18 ){  
    alert("sos mayor, podes pasar");  
} else {  
    alert("sos menor, no podes pasar");  
}  
  
//Continua el flujo normalmente  
console.log("fin del if, continua el programa");
```

The code example shows a variable 'edad' assigned the value from a prompt. An if-else statement checks if 'edad' is greater than or equal to 18. A blue box highlights the condition 'edad >= 18'. A blue arrow points from the text 'En este caso, la condición es una comprobación de si es mayor o igual a 18' to this box. The if block shows an alert message for being older, and the else block shows an alert for being younger. The code continues with a console log message.

Else if

Hasta ahora vimos que la estructura es *if-else*, pero van a haber casos en donde la condición del *if* no se cumple y queremos comprobar si se cumple otra condición. Ahí la estructura cambia, usamos el *else if*.

Lo que hace el programa es primero comprobar la primera condición

En caso de que la primera condición no se cumpla, salta a la siguiente condición y comprueba si se cumple

En este ejemplo puse cuatro condiciones, lo hice para que se entienda que se puede poner un else if, o puedo poner 2, o 3 o 70.

Es decir, puedo poner la cantidad que quiera y que necesite para hacer comprobaciones

Cuando ninguna se cumple, salta a else y ejecuta el código dentro del else

```
if( "condición 1" ){  
    /*  
    Bloque de código  
    si se cumple la  
    primera condición  
    */  
}  
else if( "condición 2" ){  
    /*  
    Bloque de código  
    si se cumple la  
    segunda condición  
    */  
}  
else if( "condición 3" ){  
    /*  
    Bloque de código  
    si se cumple la  
    tercera condición  
    */  
}  
else if( "condición 4" ){  
    /*  
    Bloque de código  
    si se cumple la  
    cuarta condición  
    */  
}  
else{  
    /*  
    Bloque de código  
    si al final  
    ninguna se cumple  
    */  
}
```

Ejemplo con else if

```
if(num > 5){  
    //Ejecuta esto cuando se cumpla la condición num > 5  
    alert("mayor que 5");  
}  
else if(num < 5){  
    //Ejecuta esto cuando se cumpla la condición num < 5  
    alert("menor que 5");  
}  
else{  
    //Ejecuta esto cuando no se cumpla que num sea < 5 ni num > 5  
    //La única posibilidad que queda cuando no se cumplan las dos anteriores, es  
    //que num sea igual a 5  
    alert("igual a 5");  
}
```

Condiciones

Para las condicionales hacemos uso de operadores de comparación (también llamados operadores relacionales) que nos indican la relación entre dos cosas, ya sea si una variable es mayor a otra, o si es igual a un valor, o si es distinto, etc.

Estas operaciones son llamadas operaciones booleanas, porque (al igual que las operaciones aritméticas matemáticas) dejan un resultado, que va a ser booleano (es decir, va el resultado va a ser uno de dos valores: true o false).

2 + 7 → 9

Esta sería una operación aritmética, que deja como resultado 9, porque $2+7 = 9$

2 < 7 → true

Esta sería una operación booleana, que usa el operador < para decir si la relación entre 2 y 7. Esto deja como resultado true porque 2 es menor a 7

Operadores
relacionales

9 < 7 → false

Esta otra operación deja como resultado un false, porque **no es verdad** que 9 sea menor a 7.

Operadores relacionales

$A < B$ → True si se cumple que A sea menor a B

$A > B$ → True si se cumple que A sea mayor a B

$A == B$ → True si se cumple que A sea igual a B

$A != B$ → True si se cumple que A sea distinto a B

$A >= B$ → True si se cumple que A sea mayor o igual a B

$A <= B$ → True si se cumple que A sea menor o igual a B

Algunos ejemplos

Ejemplo donde en un boliche solo
se deje pasar a gente adulta

```
if(edad >= 18){  
    alert("te dejo pasar");  
}
```

Ejemplo donde en un boliche Juan
tenga acceso a la zona VIP

```
if(nombre == "Juan"){  
    alert("te dejo pasar a la zona VIP porque sos Juan");  
}
```

Ejemplo donde en un boliche puede entrar
cualquier persona, menos Matías porque
tiene prohibida la entrada

```
if(nombre != "Matias"){  
    alert("te dejo pasar porque no sos matías");  
}else{  
    alert("no te dejo pasar porque sos Matías");  
}
```

El primer bloque se ejecuta si
nombre es distinto a Matías

El segundo bloque se ejecuta
si nombre es igual a Matías

Operadores lógicos

Va a haber veces en donde en una misma condición, tenemos que comprobar si se cumplen dos al mismo tiempo (o tres, o cuatro) o si se cumple una sola. Para eso hacemos uso de dos operadores lógicos: el operador AND y el operador OR.

AND

El operador AND básicamente dice “ejecuta este bloque de código si se cumple **esta condición** y **esta otra** al mismo tiempo” (es decir, va a ejecutar un código si se cumplen dos condiciones). Vamos a verlo con un ejemplo.

A && B



En JavaScript, el operador AND se escribe con &&

En este ejemplo, A y B son condiciones.

Toda esta condición compuesta va a dar como resultado true, **si y solo si** se cumple A (A = true) y se cumple B (B = true).

Vamos a ver un ejemplo

En este ejemplo pueden ocurrir cuatro casos

El primero, que $num = 15$

Es decir que está dentro del rango de 10 a 20

comprobar si un número está dentro de 10 y 20

$num > 10$	&&	$num < 20$	result
V	AND	V	V
F	AND	V	F
V	AND	F	F
F	AND	F	F

El segundo, que $num = 5$

Es decir que no está dentro del rango.

La primera condición no se cumple, pero la segunda si. Toda la condición compuesta va a dar false porque las dos no se cumplen al mismo tiempo, solo se cumple una sola

(lo mismo pasa con el tercer caso)

El cuarto no existe, es imposible que num se menor a 10 y mayor a 20. Por eso también deja como resultado false

```
let num = parseInt(prompt("ingresar num"));
```

```
if( num>10 && num<20){  
    alert("está dentro del rango");  
}else{  
    alert("no está dentro del rango");  
}
```

OR

El operador OR hace algo distinto al operador AND. Si el operador AND deja como resultado true **si y solo si** las dos condiciones se cumplen al mismo tiempo, el operador OR deja como resultado true si por lo menos una de las dos condiciones es true, o las dos al mismo tiempo.

Esto se ve bien en el siguiente ejemplo:

Supongamos que tenemos un programa en donde el programa tenga dos números en mente y nosotros tenemos que adivinar uno de esos dos.

comprobar si un número es
igual a 3 o a 7

En caso de haber
ingresado un 3 o
un 7, me deja
como resultado
true

num == 3		num == 7	result
V	OR	V	V
F	OR	V	V
V	OR	F	V
F	OR	F	F

```
let num = parseInt(prompt("ingresar num"));
```

```
if( num==3 || num==7){  
    alert("correcto");  
}else{  
    alert("incorrecto");  
}
```

En este ejemplo el programa piensa “voy a ejecutar el siguiente código si *num* es 3 o 7. Si no es ninguno de los dos, voy a ejecutar el else”

Estas tablas de valores no son para memorizarlas, son simplemente para entender como funcionan los operadores lógicos AND y OR.

En resumen

`if(num>10 && num<20)`

Si num es mayor a 10 y es menor a 20

`if(num==7 || num==3)`

Si num es 7 o es 3