

Condiciones

Las condiciones se pueden usar para flujos condicionales para determinar la siguiente tarea.

Una condición es una expresión booleana con una sintaxis similar a JavaScript. Permite comparar variables de una instancia de flujo de trabajo con otras variables o literales (por ejemplo, números, cadenas, etc.).

Se accede a las variables de una instancia de flujo de trabajo por su nombre. Si una variable tiene un valor de documento, se puede acceder a las propiedades anidadas mediante la notación de puntos. Vea la sección [Variables](#) para más detalles.

Ejemplos:

```
totalPrice > 100

owner == "Paul"

order.count >= 5 && order.count < 15
```



Literales

Literal	Ejemplos
Variable	totalPrice, order.id
Número	25, 4.5, -3, -5.5
Cuerda	"Paul", 'Jonny'
Booleano	verdadero Falso
Valor nulo	nulo

Se puede usar un valor nulo para verificar si se establece una propiedad variable o anidada (por ejemplo, `owner == null`). Si no existe una propiedad variable o anidada, entonces se resuelve `null` y se puede comparar como tal.

Operadores de comparación

Operador	Descripción	Ejemplo
==	igual a	propietario == "Paul"
!=	no igual a	dueño! = "Paul"
<	menos que	precio total <25

<=	Menos que o igual a	precio total <= 25
>	mas grande que	precio total > 25
> =	Mayor qué o igual a	precio total > = 25

Los operadores `<`, `<=`, `>` y `>=` sólo pueden utilizarse para los números.

Si los valores de un operador tienen diferentes tipos, la evaluación falla. Comparar propiedades nulas o faltantes con un número se considera como comparar diferentes tipos.

Operadores logicos

Operador	Descripción	Ejemplo
&&	y	orderCount >= 5 && orderCount < 15
	o	orderCount > 15 precio total > 50

También es posible usar paréntesis entre los operadores para cambiar la precedencia (por ejemplo, `(owner == "Paul" || owner == "Jonny") && totalPrice > 25`).