



FORO

7

respuestas

duda en opcion a

Publicado hace 2 semanas, el 5/8/22

[Programación](#) [Lógica de Programación](#) [Lógica de programación parte 2: Conceptos primordiales](#)

Referente al curso [Lógica de programación parte 2: Conceptos primordiales](#), en el capítulo [Ejecute códigos diferentes dependiendo de la condición](#) y actividad [¿Será que entra cumpliendo las condiciones?](#)

[Seguir tópico](#)

por [Ronan Omar Carrero Sandoval](#) | 15.1k xp | 2 posts
asesor comercial

no entendi por que la opcion a tenia los simbolos && y por eso pense que era la opcion c

7 RESPUESTAS

hace 2 semanas



por **Alumno** | 74.9k xp | 154 posts

El operador AND es representado con dos ampersands &&. En la programación clásica, AND retorna true si ambos operandos son valores verdaderos y false en cualquier otro caso. Ejemplos:

```
alert(true && true); // true
alert(false && true); // false
alert(true && false); // false
alert(false && false); // false
```

[COPIA EL CÓDIGO](#)

por **David Augusto Soliz Rubin De Celis** | 9.3k xp | 18 posts

hace 1 semana

Significa "y", por lo tanto, establece dos o más condiciones a cumplirse.



por **Marcos Javier Chiappini** | 8.7k xp | 20 posts

hace 1 semana

Gracias por la explicacion, yo tambien seleccione la opcion C por que no sabia que significaba &&



por **Alumno** | 6.3k xp | 2 posts

hace 1 semana

gracias por explicar, como no aparecía en los videos tampoco aprete la opcion a.



por **Rodríguez Gabriel Alejandro** | 2.9k xp | 1 posts

hace 1 semana

me parece que esta mal, la respuesta A y B son iguales y dicen (si es menor o igual a 2, y tambien es mayor a 12) por lo cual no hay forma posible de que un numero sea menor a 2 y mayor a 12 a la vez..



por **Dante Demarchi** | 5.0k xp | 10 posts
Estudiante

hace 3 días

gracias por la explicación al compañero, no sabia el significado de &.



por **AUGUSTO FRANCO BERNARDO CAPUTO ORTEGA** | 12.9k xp | 1 posts

hace 3 días

Como ya te respondieron, el doble *ampersand* (`&&`) representa la operación lógica *AND*. Creo que también puedo ayudar a entender de donde viene dicho significado y sumar más claridad a dichos resultados.

Las dos operaciones de concatenación de condiciones (*AND* y *OR*) se pueden también expresar de forma algebraica como operaciones de multiplicación (*AND*) y suma (*OR*) en un conjunto de valores (en este caso 1 y 0 porque los ordenadores utilizan binario para representar números, donde 1 representa *TRUE* y 0 representa *FALSE*). Esto quizás puede sonar complejo al principio, pero es simplemente una cuestión de matemática básica.

El resultado de una operación de concatenación se lo conoce a partir de la construcción de una "tabla de verdad", en dicha tabla se representa cada una de las condiciones (llamadas proposición o premisa en el álgebra) y se efectúa la operación correspondiente. Así, en el caso de tener una única condición, el resultado de dicha tabla de verdad representa el valor de la premisa en sí misma, es decir, no se efectúa ninguna operación extra. Cuando concatenamos condiciones, el resultado de dicha tabla será la **operación combinada de las premisas**.

Por ejemplo:

Partiendo de la variable `numero = 13` debemos comprobar que:

- Premisa 1: `numero >= 2`
- Premisa 2: `numero < 12`

Si utilizamos el operador `AND` encontraremos que, convirtiendo el valor de cada proposición a un número algebraico válido (en este caso 0 (*FALSE*) o 1 (*TRUE*), y realizando una multiplicación obtendremos el valor resultante de la evaluación conjunta de las premisas:

- $13 \geq 2$? true
- $13 < 12$? false

Entonces:

$$1 * 0 = \mathbf{0 \text{ (false)}}$$

El conjunto de verdad de las premisas es falso.

Ahora bien, si $\text{numero} = 10$:

- $10 \geq 2$? true
- $10 < 12$? true

Entonces

$$1 * 1 = \mathbf{1 \text{ (true)}}$$

Si en lugar de utilizar la operación AND utilizáramos la operación OR los pasos serían los mismos, con la diferencia que en lugar de MULTIPLICAR deberíamos SUMAR los valores:

Para $\text{numero} = 13$:

- $13 \geq 2$? true
- $13 < 12$? false

Utilizando OR ($||$), el resultado sería:

$$1 + 0 = \mathbf{1 \text{ (true)}}$$

Y para `numero = 10` :

- `10 >= 2 ? true`
- `10 < 12 ? true`

Utilizando OR (`||`), el resultado sería:

`1 + 1 = 1 (true)`

Pequeña aclaración: el resultado **en el sistema DECIMAL** de `1+1` es `2` , sin embargo, hay que recordar que aquí trabajamos con el sistema **binario**, donde `1+1 = 10` , como el número máximo de dígitos que puedo tomar es 1 el valor `10` sobrepasa el límite (*overflow*), por lo que se trunca y el cero desaparece.

Si tuviera más de 2 condiciones el calculo se evaluaría de dos en dos condiciones.

Por ejemplo:

Para `numero = 13` evaluaremos con la operación *OR* las siguientes premisas:

- `numero >= 2`
- `numero > 10`
- `numero <= 15`

Es decir, la variable puede adoptar cualquier valor entre 2 y 15 (inclusive), la operación intermedia `> 10` no cumple, en una evaluación con *OR* ninguna función. Entonces:

- a. `13 >= 2 ? true`
- b. `13 > 10 ? true`

- C. $13 \leq 15$? true

La operación matemática a construir para dicha tabla sería:

$$(a + b) + c$$

Notese el paréntesis indicando que a y b se evalúan juntas y su valor en conjunto se evalúa contra c .

El resultado sería $(1 + 1) = 1$, es decir, el valor intermedio entre $a + b$, comparado contra $c = 1$, da como resultado 1 (true)

Si usamos AND:

$$(a * b) * c$$

El resultado sería $(1 * 1) = 1$, es decir, el valor intermedio entre $a * b$, comparado contra $c = 1$, da como resultado 1 (true)

Si $\text{numero} = 10$ el resultado sería distinto en la comparación con AND pero idéntico en la comparación con OR porque:

- a. $10 \geq 2$? true
- b. $10 > 10$? false (el 10 no está incluido como valor válido de la premisa)
- c. $10 \leq 15$? true

Con OR: $(1 + 0) + 1 = 1$

Con AND: $(1 * 0) * 1 = 0$ porque $(1 * 0 = 0$ (intermedio entre premisa a y b), y $0 * 1 = 0$, el valor intermedio contra c).

RESPONDER



INSTRUCTORES

BLOG

SOBRE NOSOTROS

PREGUNTAS FRECUENTES

SUGERENCIA DE CURSOS

DISCORD ALURA

SÍGUENOS EN NUESTRAS REDES SOCIALES

