Taller de Física Computacional

Operaciones Lógicas y comparaciones

Cristián G. Sánchez y Carlos J. Ruestes

2020

Booleanos

Booleanos

Python implementa el tipo `bool` que puede tomar los valores `True` (verdadero) o `False` (falso).

Operaciones Lógicas

Operaciones Lógicas

Las siguientes operaciones lógicas están definidas para este tipo:

Operación	Tipo	Definición
`and`	Binaria	"y" lógico
`or`	Binaria	"o" lógico
`not`	Unaria	negación

Tablas de verdad

"y" lógico

Esta tabla muestra el resultado de la operación `A and B` para todas las posibles combinaciones de valores de los operandos:

Operando A	Operando B	Resultado
`True`	`True`	`True`
`False`	`False`	`True`
`True`	`False`	`False`
`False`	`True`	`False`

Tablas de verdad

"o" lógico

Esta tabla muestra el resultado de la operación `A or B` para todas las posibles combinaciones de valores de los operandos:

Operando A	Operando B	Resultado
`True`	`True`	`True`
`False`	`False`	`False`
`True`	`False`	`True`
`False`	`True`	`True`

Tablas de verdad

negación lógica

Esta tabla muestra el resultado de la operación `not A` para todos los posibles de valores del operando:

Operando	Resultado
`True`	`False`
`False`	`True`

Comparaciones de valor

Operadores de comparación de valor

Los operadores lógicos en general se combinan con los siguientes operadores de comparación:

Operador	Definición
`A == B`	`True` si A es igual a B
`A != B`	`True` si A es distinto de B
`A > B`	`True` si A es mayor a B
`A < B`	`True` si A es menor a B
`A >= B`	`True` si A es mayor o igual a B
`A <= B`	`True` si A es menor o igual a B

Las comparaciones se pueden encadenar, por jemplo `A > B > C` es una expresión válida equivalente a `(A > B) and (B > C)`.

Otros operadores que devuelven Booleanos

Otros operadores que devuelven Booleanos

Los siguiente operadores binarios también devuelven valores Booleanos. Los vamos a utilizar más adelante cuando nos adentremos en el modelo de memoria y los conjuntos y secuencias, para las de equivalencia y pertenencia respectivamente.

Operador	Definición
`A is B`	`True` si A es *equivalente*† a B
`A is not B`	`True` si A es *no equivalente* de B
`A in B`	`True` si A está contenido en B
`A not in`	`True` si A no está contenido en B

(†) El uso de *equivalente* en este contexto es inexacto, pero su significado será aclarado oportunamente.

Igualdad y desigualdad entre números de punto flotante

Teniendo en cuenta lo que hasta ahora vimos sobre ejemplos en los que la representación de punto flotante de un número real en general es inexacta, tenemos que, por ejemplo:

`0.3 == 0.1 + 0.2` es `False`.

Para comparar "igualdad" entre números reales una primera idea sería hacer algo así:

```
abs(X-Y) <= eps,
```

donde $\epsilon > 0$ es una variable que depende de cuan cerca necesitemos que X esté de Y para considerarlos "iguales".

Igualdad y desigualdad entre números de punto flotante

Pero la cosa no es tan simple. Encontrar la mejor forma de decidir cuan cerca están dos números de punto flotante es un problema no trivial. La recomendación moderada es usar la comparación relativa:

$$|X - Y| <= \text{máx}(|X|, |Y|) * \sqrt{\epsilon}$$

donde ϵ sea lo que se llama *machine epsilon*.

Por ahora podemos decir que ϵ en NumPy está en `np.finfo(float).eps` y vale `2.220446049250313e-16`.

Magia Negra Pytónica

Ojo!

```
>>> print("aa" > "a")
    True
>>> print("ab" > "a")
    True
>>> print("ab" > "b")
    False
>>> print("b" > "a")
    True
```

Notar los resultados de estas operaciones de comparación entre cadenas. En mi (limitada) experiencia es fácil terminar comparando cosas que uno no quiere comparar y aún así esa comparación puede terminar teniendo un valor.

Magia Negra Pytónica

La "magia negra" no es tal, el manual de referencia es muy explícito:

"The value of an object is a rather abstract notion in Python: For example, there is no canonical access method for an object's value. Also, there is no requirement that the value of an object should be constructed in a particular way, e.g. comprised of all its data attributes. Comparison operators implement a particular notion of what the value of an object is. One can think of them as defining the value of an object indirectly, **by means of their comparison implementation**."

Síntesis y recursos:

- Documentación sobre comparaciones en el manual de referencia.
- Más sobre expresiones condicionales.
- Artículo sobre comparación de números de punto flotante.

Para los ejercicios:

- Puerta lógica
- Lógica NAND
- Sumador