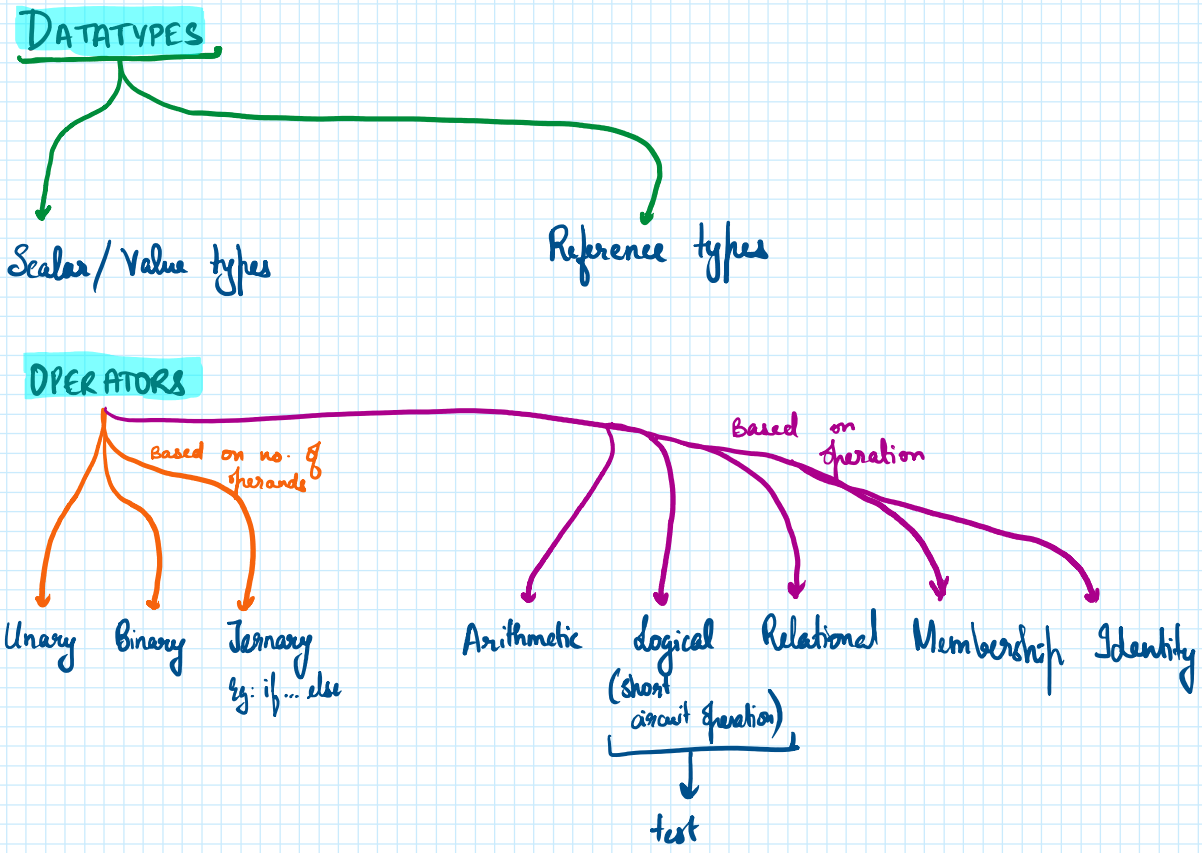
3. Datatypes, Operators, Control Statements

15 September 2023 11:43



HW: Why is id of variables different when assigned same no.?
B/w -5 and 255 values, ID remains the same
Outside this range, ID is different.. WHY?

- What is hash?

$3 > 6 > 1 = 3 > 6$ and $6 > 1$

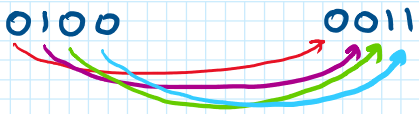
Precedence: Certain order of preference operators are evaluated in

Associativity: Order in which operators are executed when there are multiple of the same operator in one expression

BITWISE OPERATOR

Converts operands to binary and then performs an operation

4 & 3

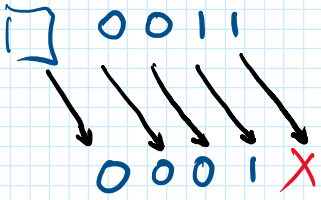


0000

0

Eg: & (and), | (or), >> (right shift), << (left shift), ~ (negation)

3 >> 1 → no of bits to shift by



NOTE: Right & left shift

$3 >> 1 = 3 \div 2^1$

$15 >> 3 = 15 \div 2^3$

$16 << 2 = 16 \times 2^2$

$11 << 1 = 11 \times 2^1$

NOTE:
Bitwise operators work only for integers

Negation

SHORT FORMULA

$\sim a = -(a+1)$

Negation returns 2's complement of given value.

BACK END

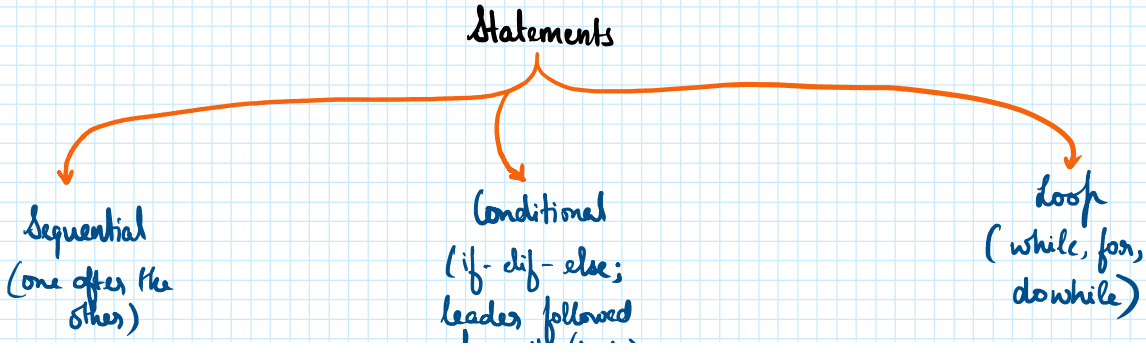
3

Binary: 00000011

1's complement: 11111100

2's complement: (1's comp. +1) 11111101 (+1)

CONTROL STATEMENTS



(one after the other)

(if- elif- else; leader followed by suite/body)

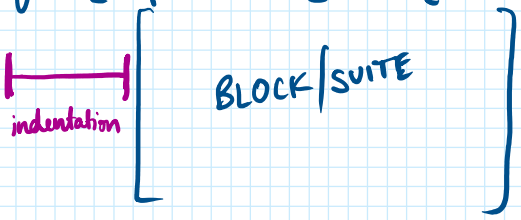
(while, for, dowhile)

If- else

- only takes expressions as conditions, not statements
- pass: used to fill empty suites. only for syntax purpose, lines after pass are still execute
- leader must have a body suite ; error without it
- only else is not allowed

For loop

- for [loop variable] in [iterable]:



inter() function ?

- loop variable is a copy of each element per iteration in given iterable
- able to handle end of iterable exception by itself without needing to specify
- range() function is useful to set a particular number of iterations, get index values etc. gives output as "range" datatype
- range [start, stop, step] where stop value is not included in the result
- output of range has index values to access specific elements