

Loop Notes – Python, Java, and JavaScript

■ Python – Key Facts

- 1 Python's for loop iterates directly over sequences (no manual init/condition/increment).
- 2 range(start, stop, step) handles initialization, condition, and increment implicitly.
- 3 Both for and while loops can have an else clause, executed only if no break occurs.
- 4 Supports backward iteration using range with negative step.
- 5 for loops can iterate over strings, lists, tuples, sets, dicts, and files.
- 6 List comprehensions allow one-line loops for building lists.
- 7 Use while loop if manual increment or conditional looping is needed.

■ Java – Key Facts

- 1 Classic C-style for loop uses (initialization; condition; increment).
- 2 All three parts of for loop header are optional (e.g., for(;;) creates an infinite loop).
- 3 Enhanced for-each loop allows iterating collections or arrays directly (for(Type var : collection)).
- 4 Supports labeled break and continue for nested loops.
- 5 For loops and while loops are block-based, requiring curly braces for multiple statements.

■ JavaScript – Key Facts

- 1 C-style for loop syntax: for(initialization; condition; increment), with all parts optional.
- 2 for(;;) creates an infinite loop.
- 3 for...in iterates over keys or indexes of objects/arrays.
- 4 for...of iterates over actual values (Python-style).
- 5 Array methods like forEach, map, filter can replace loops for cleaner functional code.
- 6 Supports labeled break and continue similar to Java.
- 7 Can loop directly through strings and iterable objects.

■ Cross-Language Summary

- 1 Python uses indentation for loop bodies; Java and JavaScript use curly braces.
- 2 Python's for is iterator-based; Java/JS for are condition-based unless using for-each or for...of.
- 3 Infinite loops: while True (Python), for(;;) (Java/JS).
- 4 Break and continue work in all three languages.
- 5 Python uniquely supports else with loops; Java and JS do not.