**[NOTES] Lambda are cool why?**

Lambda expressions. These little guys are like ninjas - small, efficient, and they get the job done without leaving a trace (or a name).

Sorting a list

1. // Before lambda
2. Collections.sort(names, new Comparator<String>() {
3. @Override
4. public int compare(String a, String b) {
5. return a.compareTo(b);
6. }
7. });

After Java 8:

1. Collections.sort(names, (a, b) -> a.compareTo(b));

It's like Java said, "Why use many word when few word do trick?" 😉

Event handling in JavaFX

1. // Before lambdas
2. button.setOnAction(new EventHandler<ActionEvent>() {
3. @Override
4. public void handle(ActionEvent event) {
5. System.out.println("Button clicked!");
6. }
7. });
9. // With lambdas
10. button.setOnAction(event -> System.out.println("Button clicked!"));

We just turned 5 lines into 1. That's an 80% code reduction! If only we could do that with our waistlines...

Creating threads

Starting a new thread? Lambdas make it look like child's play.

1. // Before lambdas
2. new Thread(new Runnable() {
3. @Override
4. public void run() {
5. System.out.println("Running in a separate thread");
6. }
7. }).start();
9. // With lambdas
10. new Thread(() -> System.out.println("Running in a separate thread")).start();

It's so simple now, you might find yourself creating threads just for fun!