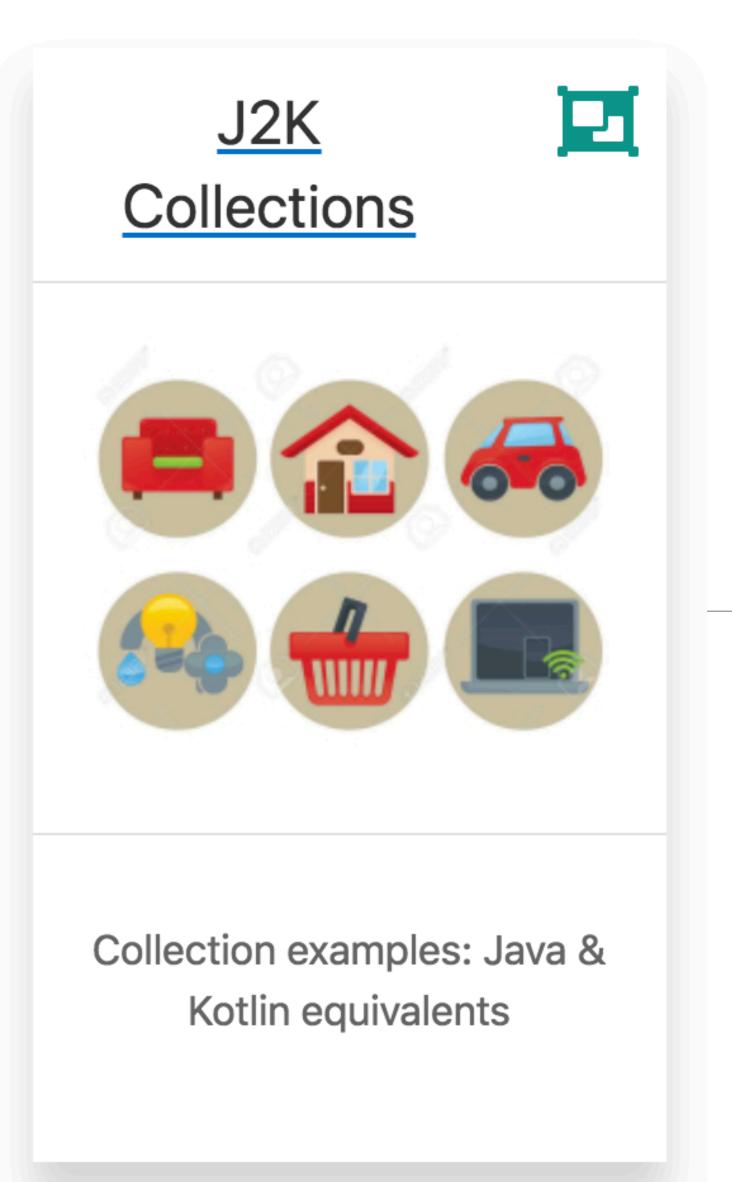
J2K: Collections



```
final List<Integer> listOfNumber = Arrays.asList(1, 2, 3, 4);

final Map<Integer, String> keyValue = new HashMap<Integer, String>();
map.put(1, "Amit");
map.put(2, "Ali");
map.put(3, "Mindorks");
```

```
final List<Integer> listOfNumber = Arrays.asList(1, 2, 3, 4);

final Map<Integer, String> keyValue = new HashMap<Integer, String>();

map.put(1, "Amit");

map.put(2, "Ali");

map.put(3, "Mindorks");
```

```
// Java 7 and below
for (Car car : cars) {
 System.out.println(car.speed);
// Java 8+
cars.forEach(car -> System.out.println(car.speed));
// Java 7 and below
for (Car car : cars) {
 if (car.speed > 100) {
   System.out.println(car.speed);
// Java 8+
cars.stream().filter(car -> car.speed > 100).forEach(car -> System.out.println(car.speed));
```

```
cars.forEach {
    println(it.speed)
}

cars.filter { it.speed > 100 }
    .forEach { println(it.speed)}

// kotlin 1.1+

cars.stream().filter { it.speed > 100 }.forEach { println(it.speed)}

cars.parallelStream().filter { it.speed > 100 }.forEach { println(it.speed)}
```

```
// Java 7 and below
for (Car car : cars) {
 System.out.println(car.speed);
// Java 8+
cars.forEach(car -> System.out.println(car.speed));
// Java 7 and below
for (Car car : cars) {
 if (car.speed > 100) {
    System.out.println(car.speed);
// Java 8+
cars.stream().filter(car -> car.speed > 100).forEach(car -> System.out.println(car.speed));
```

```
cars.forEach {
    println(it.speed)
}

cars.filter { it.speed > 100 }
    .forEach { println(it.speed)}

// kotlin 1.1+
cars.stream().filter { it.speed > 100 }.forEach { println(it.speed)}
cars.parallelStream().filter { it.speed > 100 }.forEach { println(it.speed)}
```