Iterators (ArrayList)

```
ArrayList<String> sl = new ArrayList<String>();

Iterator<String> it = sl.iterator();
while (it.hasNext()) {
  String str = it.next();
  println(str);
}
```

- **Template: specify *type* of elements in iterator**
- **Iterator: used to go through all elements in a set**

Iterators (ArrayList)

```
ArrayList<String> sl = new ArrayList<String>();

Iterator<String> it = sl.iterator();
while (it.hasNext()) {
  String str = it.next();
  println(str);
}
```

Same as:

```
for (int i = 0; i < sl.size(); i++) {
  String str = sl.get(i);
  println(str);
}
```

- **Iterator over ArrayList provides elements in order**

Iterators (ArrayList)

```
ArrayList<String> sl = new ArrayList<String>();
```

```
Iterator<String> it = sl.iterator();
while (it.hasNext()) {
  String str = it.next();
  println(str);
}
```

Java 5.0+ iterator shorthand:

```
for (String str: sl) {
  println(str);
}
```

- **Iterator over ArrayList provides elements in order**

“For each”

Java 5.0+ iterator shorthand:

```
for (String str: sl) {
  println(str);
}
```

General form:

```
for (type variable: collection) {
  ... body ...
}
```

Iterators (Map keySet)

```
Map<String,Integer> phonebook =
               new HashMap<String,Integer>();

Iterator<String> it =
               phonebook.keySet().iterator();

while (it.hasNext()) {
  String key = it.next();
  println(key + ":" + phonebook.get(key));
}
```

- **Iterator over Map keySet provides elements in <u>random order!</u>**