

Looping Over Real Test Data

A counter is useful; the everyday job is walking passwords, browsers, and rows. Start at zero, stop before .length, and use for-each when you do not need the index.

Automation Foundations · ashishautomationlabs.store

A counter is useful, but the everyday job is looping over a set of values -- passwords, browsers, rows. You hold many values in an array (you met `String[] args` in Chapter 1). Here it does real work:

```
public class L2 {
    public static void main(String[] args) {
        String[] passwords = {"", "123", "password", "admin", "P@ss!"};
        for (int i = 0; i < passwords.length; i++) {
            System.out.println("Attempt " + (i + 1) + ": trying '" + passwords[i] +
                "'");
        }
        System.out.println("Total attempts: " + passwords.length);
    }
}
```

```
Attempt 1: trying ''
Attempt 2: trying '123'
Attempt 3: trying 'password'
Attempt 4: trying 'admin'
Attempt 5: trying 'P@ss!'
Total attempts: 5
```

Three details worth noticing:

- The counter starts at **0**, not 1 -- array positions in Java are numbered from zero: the first item is `passwords[0]`, the fifth is `passwords[4]`.
- The condition uses `passwords.length`, the count of items -- so the same loop works for five passwords or five hundred, with no edit.
- `passwords[i]` reads the item at position `i`. You wrote `i + 1` only to print a human-friendly "Attempt 1" instead of "Attempt 0".

You are using arrays here before they are fully explained. That is on purpose -- loops and arrays are easier to learn together. The full treatment is Chapter 5. Until then: an array is a numbered row of boxes, counted from zero, and `.length` tells you how many.

For each: when you do not need the index

There is a cleaner way to say "do this for every item" when you do not need the position number -- the **enhanced for**, often read aloud as "for each":

```
public class L3 {  
    public static void main(String[] args) {  
        String[] browsers = {"chrome", "firefox", "edge"};  
        for (String browser : browsers) {  
            System.out.println("Launching " + browser);  
        }  
    }  
}
```

```
Launching chrome  
Launching firefox  
Launching edge
```

Read the header as "for each browser in browsers." No counter, no condition, no [i] -- Java walks the array and hands you one item each pass. Use this form whenever you want every item and do not care about its index; use the counted form when you need the position, or when you are not walking an array at all.

You can drive a password list and a browser list through one loop shape. The next lesson is for when you cannot answer "how many times?" -- only "until what?": while and do-while, the shapes behind retries and polls.