

Iterating Over an Array. Example of Applying Parameterized Live Templates

As an example of applying a parameterized template, let us create a construct that iterates over an array.

To iterate over an array

1. Open the desired class for editing.
2. Type the itar abbreviation:

```
int[] arrayOfInts = new int[] {7,9,3,17,11,23,37};
char[] arrayOfChars = new char[] {'a', 'b', 'c'};
itar|
```

3. Press the template invocation key. The abbreviation expands into a code construct, with the suggested values for the template's variables already inserted:

```
int[] arrayOfInts = new int[] {7,9,3,17,11,23,37};
char[] arrayOfChars = new char[] {'a', 'b', 'c'};
for (int i = 0; i < arrayOfChars.length; i++) {
    char aChar = arrayOfChars[i];
}
}
```

4. The Editor is now in its special template editing mode. The cue is the red border around the index variable, which is the first input position in this template.

IntelliJ IDEA automatically suggests `i` as the value for the index variable. If you want to change the suggested name, type a new one. All references to the variable within the expanded template will be automatically changed as you type the new name.

In figure below, `i` is changed to `idx`:

```
int[] arrayOfInts = new int[] {7,9,3,17,11,23,37};
char[] arrayOfChars = new char[] {'a', 'b', 'c'};
for (int idx = 0; idx < arrayOfChars.length; idx++) {
    char aChar = arrayOfChars[idx];
}
}
```

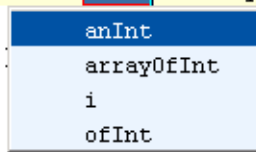
5. Press Tab or Enter to move to the next input position defined by the template:

```
int[] arrayOfInts = new int[] {7,9,3,17,11,23,37};
char[] arrayOfChars = new char[] {'a', 'b', 'c'};
for (int idx = 0; idx < arrayOfChars.length; idx++) {
    char aChar = arr
    arrayOfChars char[]
    arrayOfInts int[]
}
}
```

As you can see on the image above, IntelliJ IDEA automatically detects all array elements in the scope, and suggests selecting the correct one. If there is only one array element in the scope, IntelliJ IDEA inserts its name automatically without displaying a pop-up list.

6. IntelliJ IDEA detects whether the type of the array elements changes. It automatically updates the type of the variable that holds the current array element, and suggests a list of reasonable names for it:

```
int[] arrayOfInts = new int[] {7,9,3,17,11,23,37};
char[] arrayOfChars = new char[] {'a', 'b', 'c'};
for (int idx = 0; idx < arrayOfInts.length; idx++) {
    int anInt = arrayOfInts[idx];
```



7. Press Tab or Enter to apply the selected/specified variant. This will move you to the next input position which in our case is the END position of the template:

```
int[] arrayOfInts = new int[] {7,9,3,17,11,23,37};
char[] arrayOfChars = new char[] {'a', 'b', 'c'};
for (int idx = 0; idx < arrayOfInts.length; idx++) {
    int anInt = arrayOfInts[idx];
    |
}
```

See Also

Concepts:

- [Live Templates](#)

Procedures:

- [Creating Code Constructs by Live Templates](#)

Reference:

- [Live Templates](#)

Web Resources:

- [Developer Community](#) 