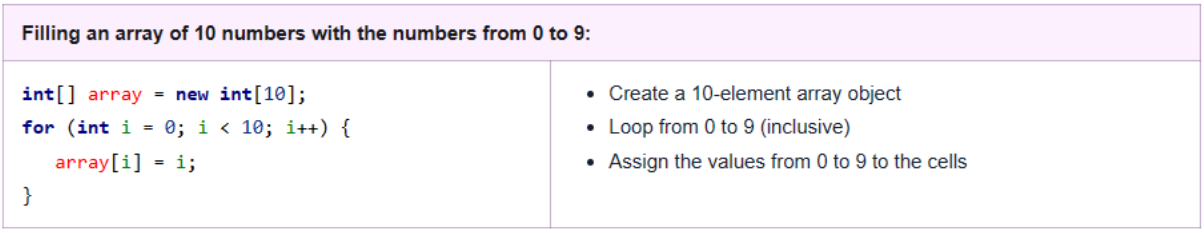
Examples of using arrays

1. Useful examples of working with arrays:

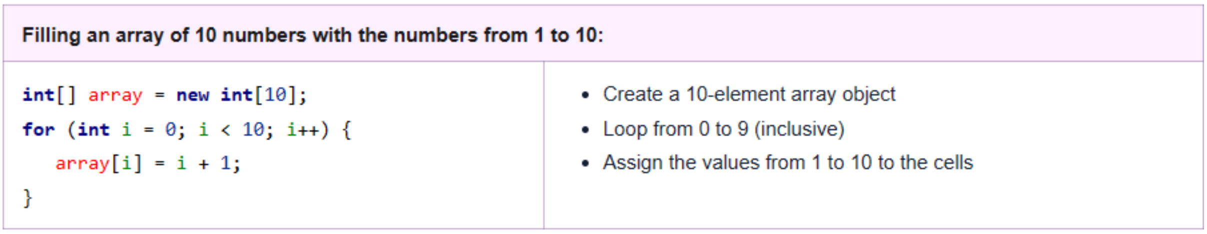
I think the time has come to move on to some practical tasks. We'll start with a few of the simplest:



Filling an array of 10 numbers with the numbers from 0 to 9:

```java
int[] array = new int[10];
for (int i = 0; i < 10; i++) {
   array[i] = i;
}
```

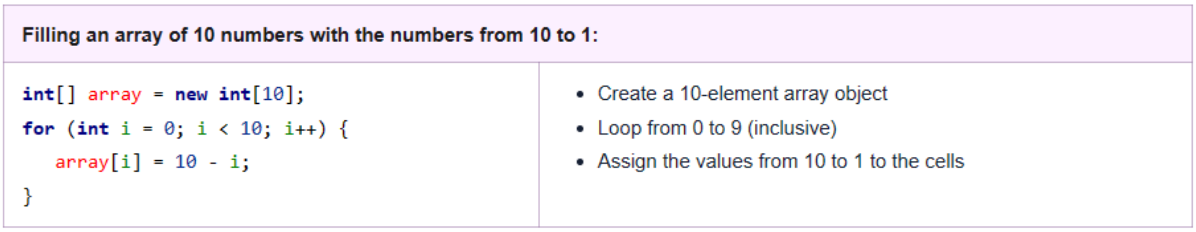
- Create a 10-element array object
- Loop from 0 to 9 (inclusive)
- Assign the values from 0 to 9 to the cells



Filling an array of 10 numbers with the numbers from 1 to 10:

```java
int[] array = new int[10];
for (int i = 0; i < 10; i++) {
   array[i] = i + 1;
}
```

- Create a 10-element array object
- Loop from 0 to 9 (inclusive)
- Assign the values from 1 to 10 to the cells



Filling an array of 10 numbers with the numbers from 10 to 1:

```java
int[] array = new int[10];
for (int i = 0; i < 10; i++) {
   array[i] = 10 - i;
}
```

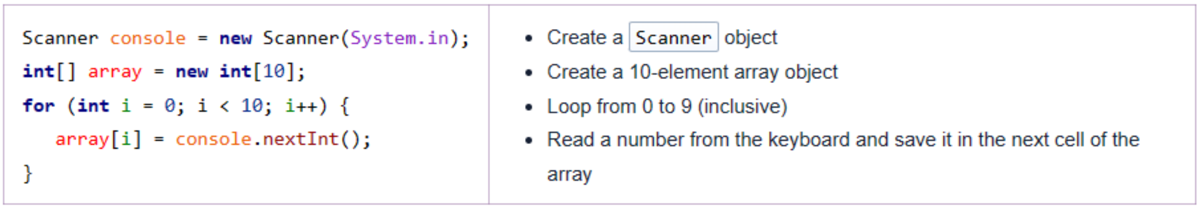
- Create a 10-element array object
- Loop from 0 to 9 (inclusive)
- Assign the values from 10 to 1 to the cells

2. Displaying numbers in reverse order

Now let's move on to more complex and interesting examples. We'll start with the following task: read 10 numbers from the keyboard and display them in reverse order.

We know how to read numbers from the keyboard. But how do we read 10 numbers? We could, of course, create 10 variables: `a1`, `a2`, etc. But that would be super inconvenient. And what if we had to read in 100 numbers? Would we create 100 variables? As it happens, we just learned about arrays, which are created to store lots of values.

The code for reading in 10 values would look something like this (this snippet would appear inside the `main` method):

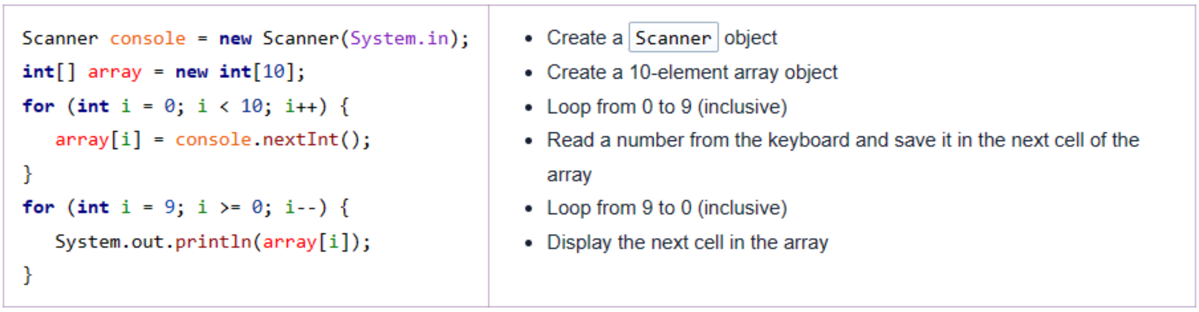


```
Scanner console = new Scanner(System.in);
int[] array = new int[10];
for (int i = 0; i < 10; i++) {
   array[i] = console.nextInt();
}
```

- Create a `Scanner` object
- Create a 10-element array object
- Loop from 0 to 9 (inclusive)
- Read a number from the keyboard and save it in the next cell of the array

But how do you print the values of the array in reverse order?

To do this, we need one more loop, where `i` will take values from 9 to 0 (don't forget that the number of array indices starts from 0). The final program code will look something like this:

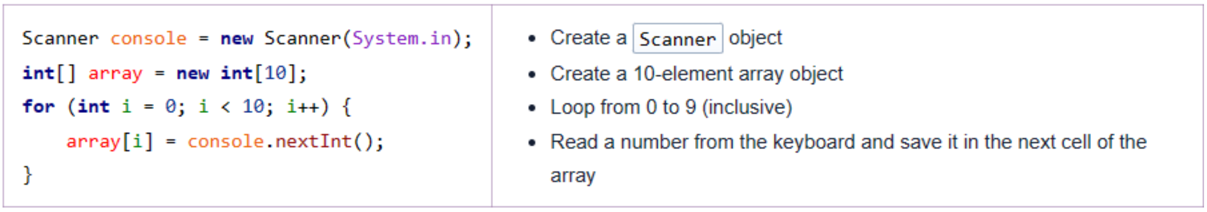


```
Scanner console = new Scanner(System.in);
int[] array = new int[10];
for (int i = 0; i < 10; i++) {
   array[i] = console.nextInt();
}
for (int i = 9; i >= 0; i--) {
   System.out.println(array[i]);
}
```

- Create a `Scanner` object
- Create a 10-element array object
- Loop from 0 to 9 (inclusive)
- Read a number from the keyboard and save it in the next cell of the array
- Loop from 9 to 0 (inclusive)
- Display the next cell in the array

3. Finding the minimum element in an array

Let's take a look at a very interesting and common task: finding the minimum element in an array. We'll grab the code we used to populate the array in the previous task:



```
Scanner console = new Scanner(System.in);
int[] array = new int[10];
for (int i = 0; i < 10; i++) {
    array[i] = console.nextInt();
}
```

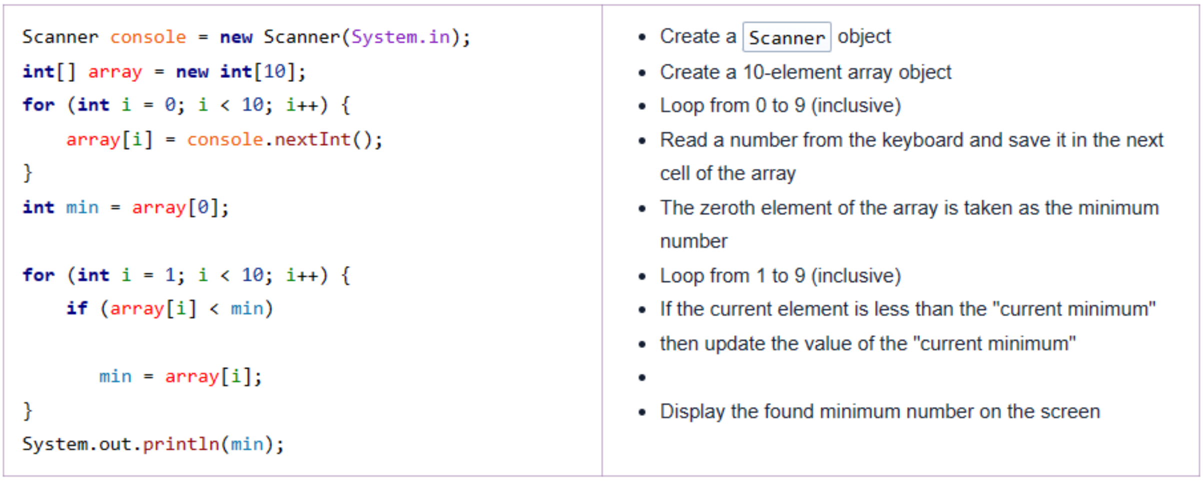
- Create a `Scanner` object
- Create a 10-element array object
- Loop from 0 to 9 (inclusive)
- Read a number from the keyboard and save it in the next cell of the array

Now all we need to do is write code that will find the minimum element in the array and display it on the screen. How do you do that?

Well, to find the minimum element, you need to:

- Take the array's first element as the "current minimum".
- Compare all the elements of the array with it one by one
- If the next element is less than the "current minimum", then update the value of the "current minimum"

This is how it will look in code:



```java
Scanner console = new Scanner(System.in);
int[] array = new int[10];
for (int i = 0; i < 10; i++) {
    array[i] = console.nextInt();
}
int min = array[0];

for (int i = 1; i < 10; i++) {
    if (array[i] < min)

       min = array[i];
}
System.out.println(min);
```

- Create a `Scanner` object
- Create a 10-element array object
- Loop from 0 to 9 (inclusive)
- Read a number from the keyboard and save it in the next cell of the array
- The zeroth element of the array is taken as the minimum number
- Loop from 1 to 9 (inclusive)
- If the current element is less than the "current minimum"
- then update the value of the "current minimum"
-
- Display the found minimum number on the screen