

# Day20

## Break & Continue

break : 반복문을 강제로 빠져나오기 위해 사용  
Continue : 반복 중 건너뛰기를 위해 사용됨

### break 문



```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <script>
5
6       var sum = 0;
7       var i = 0;
8
9       while(true){
10
11         i++;
12
13         if (i>=100){
14           break;
15         }
16
17         if(i%2==1){
18           sum = i+sum;
19         }
20       }
21       document.write(+sum);
22
23     </script>
24   </body>
25 </html>
```

2500

### Continue 문

```
<> JS08_2.html > html > body > script
1  <!DOCTYPE html>
2  <html>
3      <body>
4          <script>
5             
6              var sum = 0;
7              var i = 0;
8             
9              while(i<100){
10                 i++;
11                 if(i%2!=1){
12                     continue;
13                 }
14                 sum = i + sum;
15             }
16             document.write(sum);
17         </script>
18     </body>
19 </html>
```

2500

## 배열 Array

여러개의 변수를 저장할 수 있게 사용되는 특수 변수

var 배열명 = {값1, 값2, 값3}

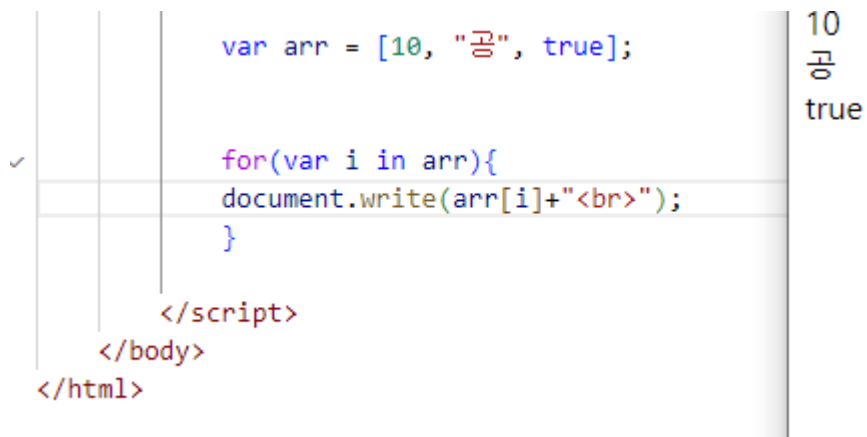
```
html <> JS07_2.html M x <> JS08_1.html U <> JS08_2.html U <> JS08_3.html U x
<> JS08_3.html > html > body > script
1  <!DOCTYPE html>
2  <html>
3      <body>
4          <script>
5             
6              var arr = [1,2,3,4,5];
7             
8              document.write(arr[0]+"<br>");
9              document.write(arr[1]+"<br>");
10             document.write(arr[2]+"<br>");
11             document.write(arr[3]+"<br>");
12             document.write(arr[4]+"<br>");
13             document.write(arr[5]+"<br>");
14         </script>
15     </body>
16 </html>
```

1  
2  
3  
4  
5  
undefined



```
var arr = [10, "공", true];
```

- 자바와 달리 다양한 형의 데이터를 하나의 배열에 넣을수 있음



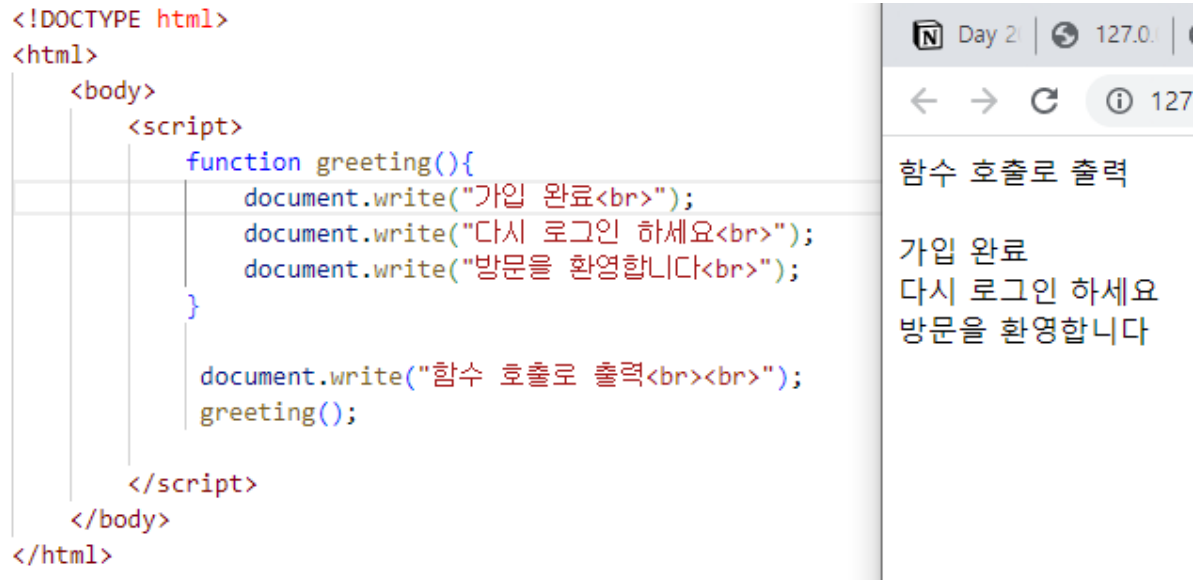
- 자바와 달리 for in 구문을 사용해 간단하게 처리가 가능

## Function 함수

```
function 함수이름(인수들){
    실행할 구문들;
}
```

```
}
```

- 기본적으로 코드의 반복 작성을 피하기 위해 많이 사용



## 함수의 특징

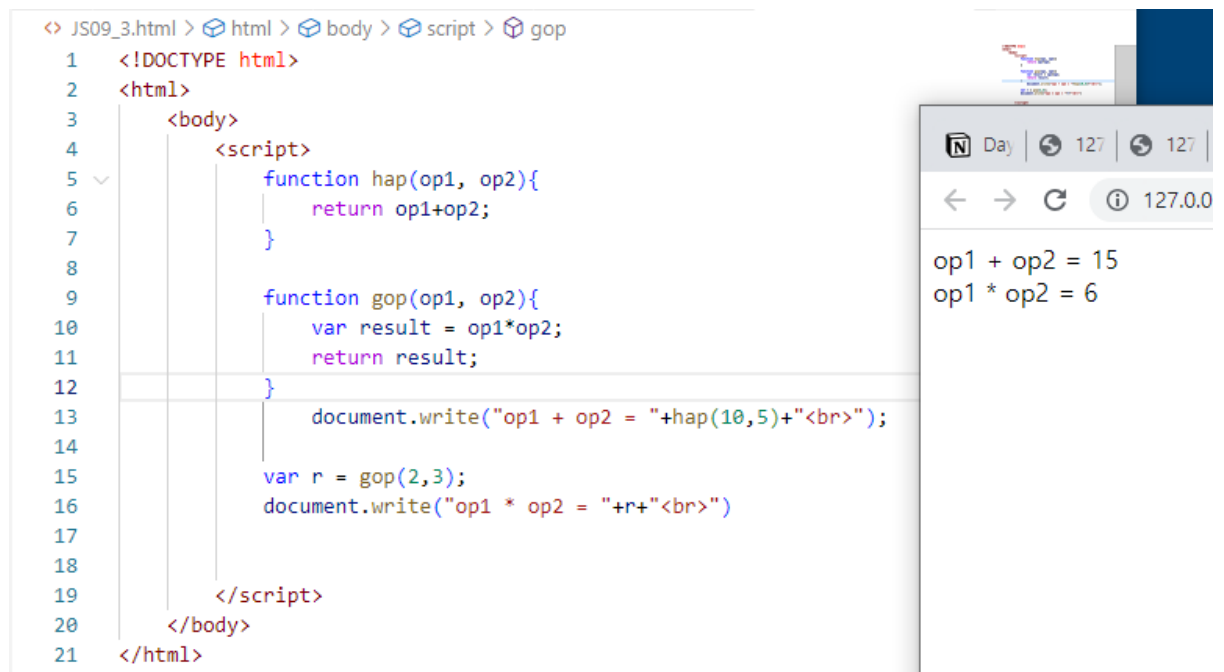
1. 함수 선언은 `function` 으로 시작하여 뒤에 함수명을 기술하고 `()` 안에 인수를 기술하되 인수는 없을수 있음
2. 함수는 선언만으로는 결코 실행되지 않으며 반드시 호출에 의해서 실행됨  
호출은 "함수명()" 으로 함
3. 주된 목적은 반복되는 소스코드량을 줄이기 위함

## 인수를 받아서 사용하는 함수



- 함수를 호출할때 인수에 따라 다른 결과를 얻을수 있음

## 결과값을 반환하는 함수



## 많이 사용되는 내장 함수

- 사용자가 직접 만들지 않아도 이미 사용 할 수 있게 정의해놓은 함수

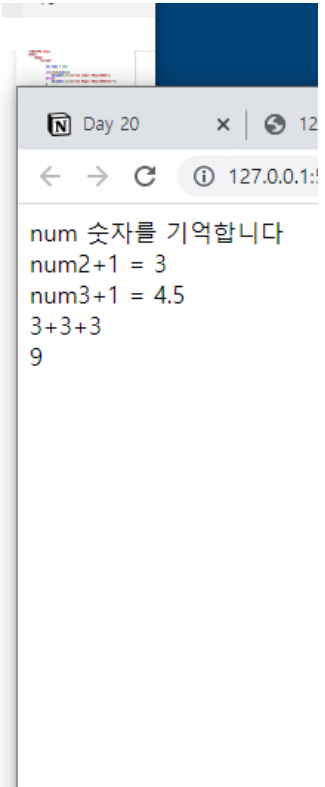
isNaN() : 숫자가 아니면 true 숫자이면 false를 반환

parseInt() : 정수 문자를 정수로 변환

parseFloat() : 실수 문자를 실수로 변환

eval() : 문자열로 구성된 식을 계산하여 결과를 반환

```
<> JS10_1.html > html > body > script
1  <!DOCTYPE html>
2  <html>
3      <body>
4          <script>
5             
6              var num1 = 123;
7             
8              if(isNaN(num1)){
9                  document.write("num 문자를 기억합니다");
10             }else{
11                 document.write("num 숫자를 기억합니다<br>");
12             }
13            
14             var num2 = "2";
15             document.write("num2+1 = " + (parseInt(num2)+1)+"<br>");
16            
17             var num3 = "3.5";
18             document.write("num3+1 = " + (parseFloat(num3)+1)+"<br>");
19            
20             var num4 = "3+3+3";
21             document.write(num4+"<br>");
22             document.write(eval(num4));
23            
24         </script>
25     </body>
26 </html>
```



## 전역 변수와 지역 변수

전역변수 : 프로그램이 실행되는 동안 메모리에 계속 생존

프로그램이 종료되면 사라짐

```

... nl U <> JS09_2.html U <> JS09_3.html U <> JS10_1.html U <> JS10_2.html
<> JS10_2.html > html > body > script > add
1 <html>
2   <body>
3     <script>
4
5       var counter = 0;
6
7       function add() {
8
9         return counter = counter + 1;
10      }
11      document.write(add()+"<br>");
12      document.write(add()+"<br>");
13      document.write(add()+"<br>");
14
15    </script>
16  </body>
17 </html>

```

Day 20

1  
2  
3

지역변수 : 함수 내부에서 선언된 변수

함수 종료되면 사라짐

```

<> JS10_2.html > html > body > script
1 <html>
2   <body>
3     <script>
4
5       function add() {
6         var counter = 0;
7         return counter = counter + 1;
8       }
9
10      document.write(add()+"<br>");
11      document.write(add()+"<br>");
12      document.write(add()+"<br>");
13
14    </script>
15  </body>
16 </html>

```

Day 20

1  
1  
1

## 객체 Object

- 자바스크립트도 객체 지향 언어

```

<> JS10_3.html > html > body > script
1 <html>
2 <body>
3 <script>
4
5 // 괄호가 없으면 프로퍼티
6 // 객체.프로퍼티
7 var str = "ABCDEF";
8 document.write(str.length+"<br>");
9 document.write("ABCDEF".length+"<br>");
10
11 // 괄호가 있으면 매서드
12 // 객체.매서드()
13 var num = 3.141592;
14 document.write(num.toFixed(2)+"<br>");
15 document.write(3.141592.toFixed(2)+"<br>");
16
17 </script>
18 </body>
19 </html>

```

Day 20

6  
6  
3.14  
3.14

## 문자열 객체

속성 : property

length : 문자열의 길이 속성을 알려줌

매서드 : method

charAt() : 문자열 내에서 특정 위치의 문자를 반환

concat() : 2개 이상의 문자열을 결합

indexOf() : 문자열 내의 첫번째 특정 문자의 위치값을 반환

replace() : 특정 문자열을 다른 문자열로 대치

split() : 하나의 문자열을 여러개의 문자열로 분할

substr() : 특정위치로부터 특정 개수의 문자열 추출

toLowerCase() : 문자열을 소문자로 변환

toUpperCase() : 문자열을 대문자로 변환



```
_1.html > html
<!DOCTYPE html>
<html>
  <body>
    <script>
      var str = "abc def";

      document.write(str.length + "<br>");
      document.write(str.charAt(1)+"<br>");
      document.write(str.replace("abc","XYZ")+"<br>");
      document.write(str.split()+"<br>");
      document.write(str.substring(1,5)+"<br>");
      document.write(str.toUpperCase(str)+"<br>");
    </script>
  </body>
</html>
```

7  
b  
XYZ def  
abc def  
bc d  
ABC DEF

## 배열 객체

length 배열의 길이를 반환

concat() : 2개 이상의 배열을 하나로 합침

indexOf() : 배열 내의 특정 요소에 인덱스 값을 구함

pop() : 배열의 마지막 요소 하나를 삭제

push() : 배열의 마지막에 요소를 하나 추가

reverse() : 배열 내 요소들을 역순으로 정렬

sort() : 배열 내 요소들의 순서를 기본 오름차순으로 정렬

## ArrayList

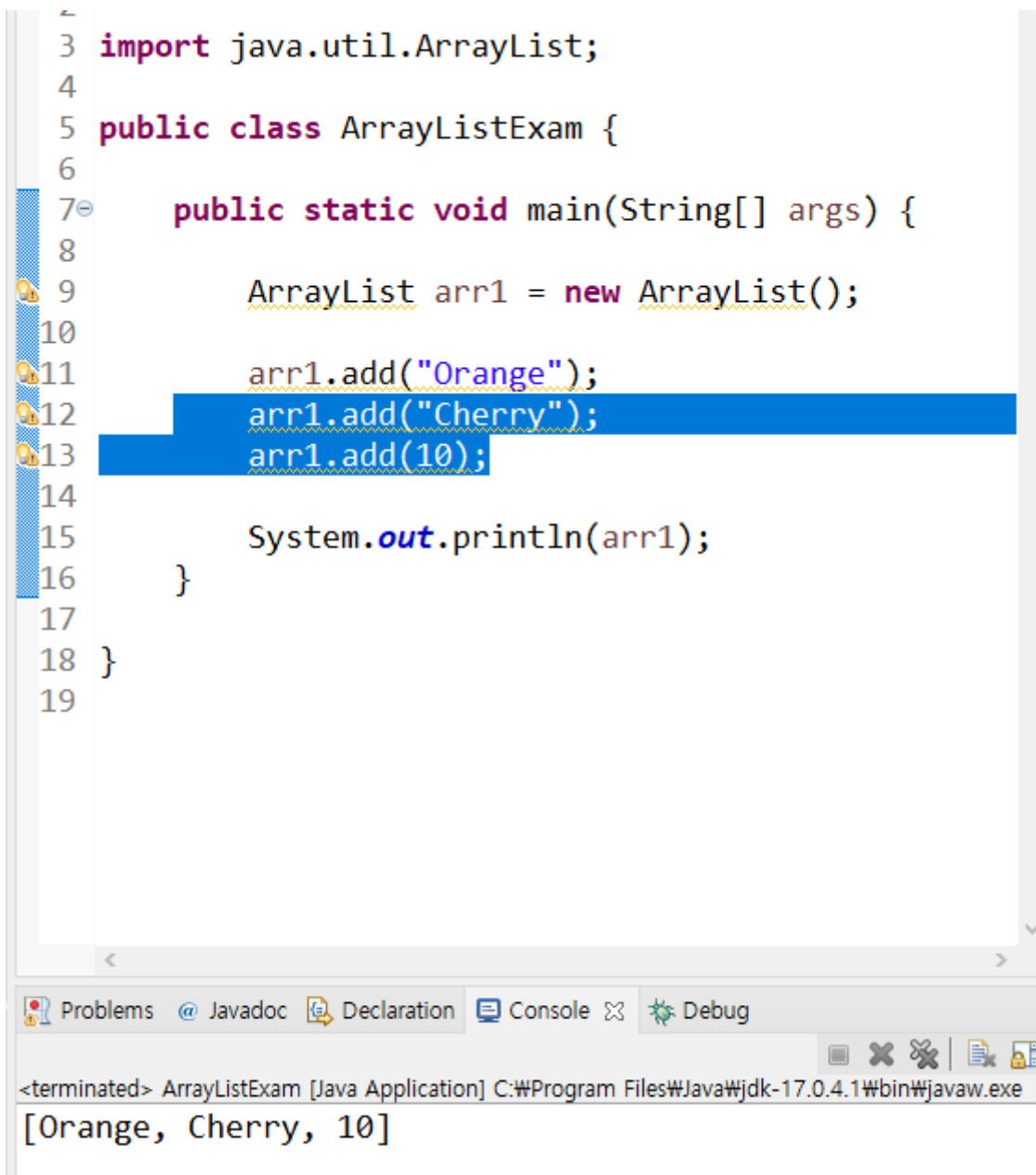
배열과 동일하게 연속된 메모리 공간을 사용하고  
인덱스도 0에서 시작

## 1. 생성

```
import java.util.ArrayList;

ArrayList arr1 = new ArrayList();
```

- 기본적으로 데이터의 타입을 확인하지도 않고 다양한 데이터를 담을 수 있음
- 하나의 데이터 타입만 저장하는 것을 강력히 권장



```
3 import java.util.ArrayList;
4
5 public class ArrayListExam {
6
7     public static void main(String[] args) {
8
9         ArrayList arr1 = new ArrayList();
10
11         arr1.add("Orange");
12         arr1.add("Cherry");
13         arr1.add(10);
14
15         System.out.println(arr1);
16     }
17 }
18 }
19 }
```

The screenshot shows an IDE window with the above code. The lines `arr1.add("Cherry");` and `arr1.add(10);` are highlighted in blue. Below the code editor, the IDE's output window is visible, showing the command prompt output: `<terminated> ArrayListExam [Java Application] C:\Program Files\Java\jdk-17.0.4.1\bin\javaw.exe [Orange, Cherry, 10]`.

## Generic

- 데이터 타입을 한정시켜 다른 데이터 타입을 넣을 수 없음

```
ArrayList<String> arr2 = new ArrayList<>();

arr2.add("Orange");
arr2.add("Cherry");
arr2.add(10);

System.out.println(arr2);
```

- 데이터 추가 삭제

```
18
19 arr2.add("Orange");
20 arr2.add("Cherry");
21 arr2.add("apple");
22 System.out.println(arr2);
23
24 arr2.remove("Cherry");
25 System.out.println(arr2);
26
27 }
28
29 }
30
```

<terminated> ArrayListExam [Java Application] C:\Program Files\Java\W

[Orange, Cherry, apple]  
[Orange, apple]

```

16
17     ArrayList<String> arr2 = new ArrayList<>();
18
19     arr2.add("Orange");
20     arr2.add("Cherry");
21     arr2.add("apple");
22     System.out.println(arr2);
23
24     arr2.remove("Cherry");
25     System.out.println(arr2);
26
27     System.out.println(arr2.indexOf("apple"));
28
29 }
30

```

Problems @ Javadoc Declaration Console Debug  
 <terminated> ArrayListExam [Java Application] C:\Program Files\Java\jdk-17.0.4.1\bin\javaw.exe (202

[Orange, Cherry, apple]

[Orange, apple]

1

실행 결과

a[0] = 0.0  
 a[1] = 0.0  
 a[2] = 0.0  
 a[3] = 0.0  
 a[4] = 0.0

```
1 package ArrayList;
2
3 import java.util.ArrayList;
4
5 public class ArrayListExam2 {
6
7     public static void main(String[] args) {
8
9         ArrayList<Double> arr1 = new ArrayList<>();
10
11         arr1.add(3.5);
12         arr1.add(5.5);
13         arr1.add(7.0);
14         arr1.add(0.2);
15         arr1.add(2.7);
16
17         for (int i=0;i<5;i++) {
18
19             System.out.println("a["+i+"] = "
20                               +(arr1.get(i)));
21
22         }
23     }
24 }
```

<terminated> ArrayListExam2 [Java Application] C:\Program Files\Java\jdk-17.0.4\bin\javaw.exe

```
a[0] = 3.5
a[1] = 5.5
a[2] = 7.0
a[3] = 0.2
a[4] = 2.7
```

package ArrayList;

import java.util.ArrayList;

public class ArrayListExam2 {

```
    public static void main(String[] args) {
```

```
        ArrayList<Double> arr1 = new ArrayList<>();
```

```
arr1.add(3.5);
arr1.add(5.5);
arr1.add(7.0);
arr1.add(0.2);
arr1.add(2.7);

for (int i=0;i<5;i++) {

    System.out.println("a["+i+"] = "+(arr1.get(i)));

}
}
```

## 쓰레드 Thread

- 한 프로세스 내에 두개 이상의 일을 동시에 할수 있음

Thread를 상속 받아 사용하는 방법

The screenshot shows an IDE with three tabs: RandomExam3..., RandomExam4..., and ThreadExam1.... The ThreadExam1... tab is active, displaying the following Java code:

```
1 package ArrayList;
2
3 public class ThreadExam1 extends Thread {
4
5     public void run() {
6         System.out.println("Thread run");
7     }
8
9
10    public static void main(String[] args) {
11
12        ThreadExam1 sample = new ThreadExam1();
13        sample.start();
14
15    }
16
17 }
18
```

Below the code editor, the 'Console' tab is active, showing the output of the program:

```
<terminated> ThreadExam1 [Java Application] C:\Program Files\Java\jdk-17.0.4.1\bin\javaw.exe (202:
Thread run
```

- start() 실행시 run 메서드가 수행되도록 내부적으로 이미 동작이 정의 되어있음

## 난수 발생하는 방법

1. Math.random()

```
1 package ArrayList;
2
3 public class MathRandom {
4
5     public static void main(String[]
6
7         System.out.println
8         |((int)(Math.random()*100));
9
10     }
11
12 }
13
```

<terminated> MathRandom [Java Application] C:\Program Files\Java\jdk

42

2. import java.util.Random 추가해 사용하는 방법



```
RandomExam.java RandomExam1.... ThreadExam1.... »s
1 package ArrayList;
2
3 import java.util.Random;
4
5 public class RandomExam {
6
7     public static void main(String[] args) {
8
9         Random rand = new Random();
10
11         System.out.println(rand.nextInt());
12         System.out.println(rand.nextInt(100));
13
14         System.out.println(rand.nextBoolean());
15         System.out.println(rand.nextDouble());
16         System.out.println(rand.nextLong());
17         System.out.println(rand.nextFloat());
18
19     }
20
21 }
22
```

Problems Javadoc Declaration Console Debug

<terminated> RandomExam [Java Application] C:\Program Files\Java\jdk-17.0.4.1\bin\javaw.exe (2022. 10)

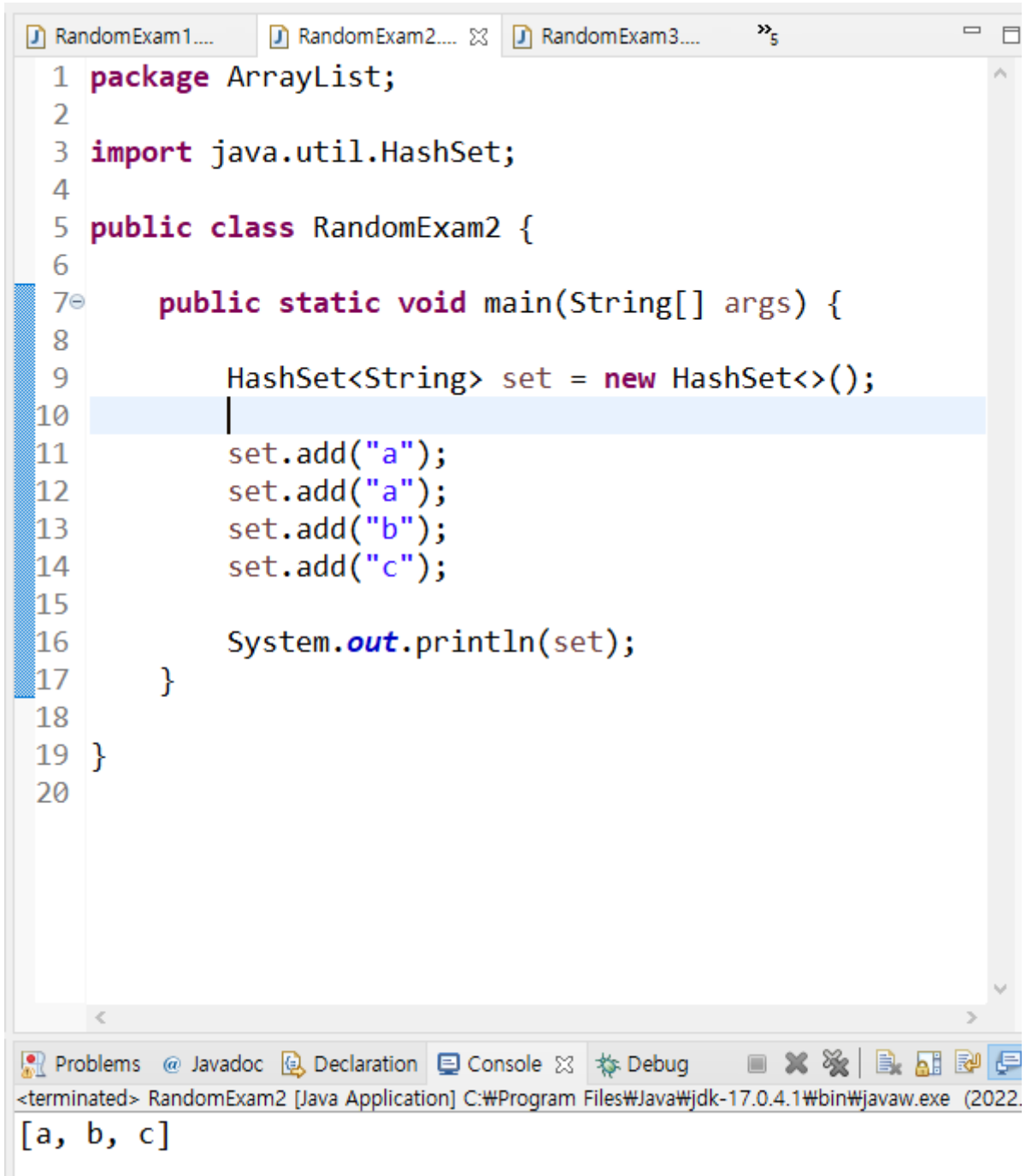
-742935030  
22  
true  
0.7513457234783615  
-8666312608428863797  
0.2560343

- 다양한 형태의 랜덤 출력이 가능함

## HashSet

중복된 값을 저장하지 않는 ArrayList 형태

```
HashSet<데이터타입> 변수명 = new HashSet<>();
```



The screenshot shows an IDE with three tabs: RandomExam1..., RandomExam2..., and RandomExam3.... The active tab, RandomExam2..., contains the following Java code:

```
1 package ArrayList;
2
3 import java.util.HashSet;
4
5 public class RandomExam2 {
6
7     public static void main(String[] args) {
8
9         HashSet<String> set = new HashSet<>();
10
11         set.add("a");
12         set.add("a");
13         set.add("b");
14         set.add("c");
15
16         System.out.println(set);
17     }
18 }
19 }
20
```

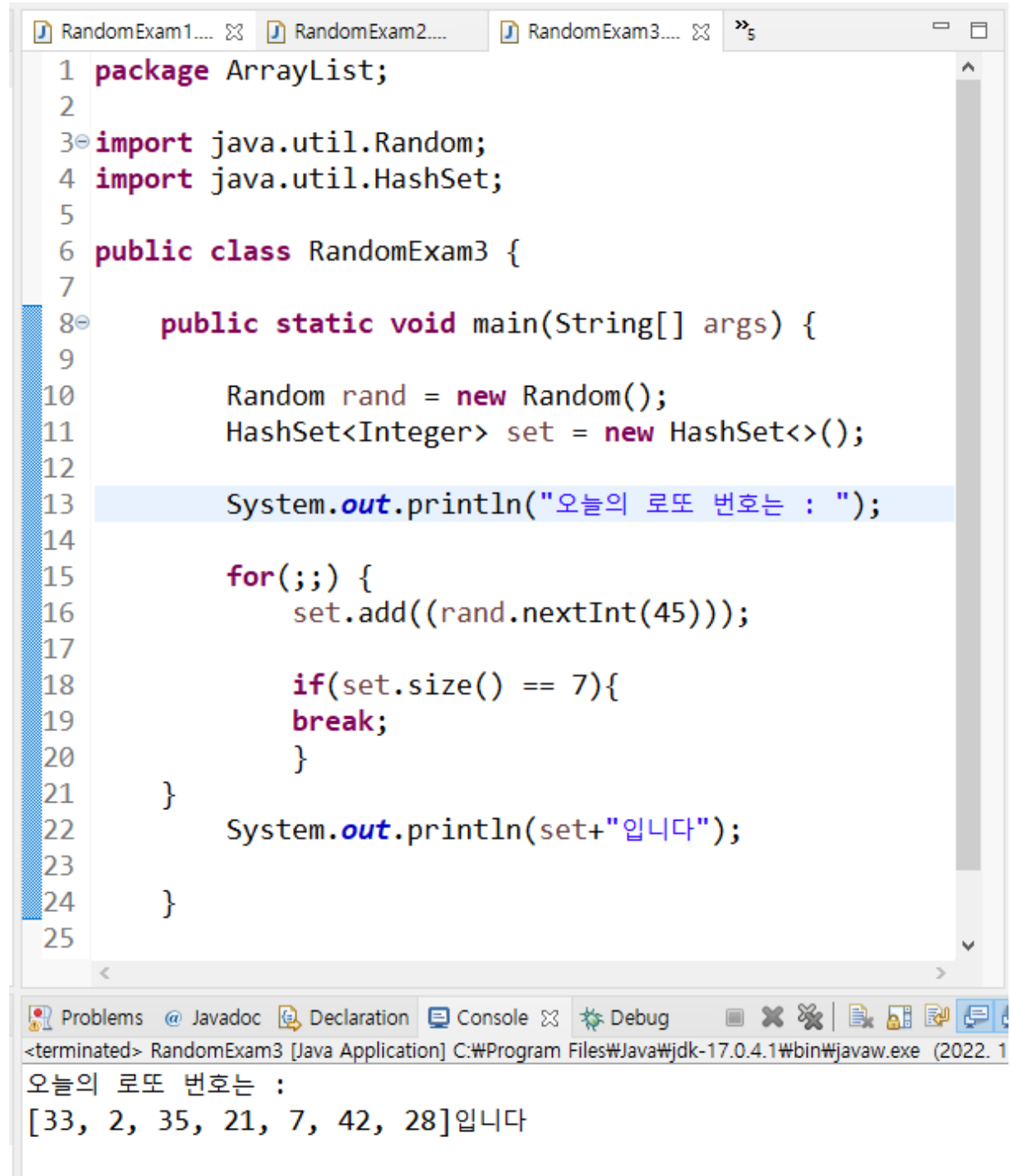
The console output at the bottom shows the result of running the program:

```
<terminated> RandomExam2 [Java Application] C:\Program Files\Java\jdk-17.0.4.1\bin\javaw.exe (2022.
[a, b, c]
```

- 중복된 값은 저장하지 않음

## Quiz 로또 번호 출력

### Random, HashSet 사용



The screenshot shows an IDE with three tabs: RandomExam1..., RandomExam2..., and RandomExam3.... The code in RandomExam3.java is as follows:

```
1 package ArrayList;
2
3 import java.util.Random;
4 import java.util.HashSet;
5
6 public class RandomExam3 {
7
8     public static void main(String[] args) {
9
10         Random rand = new Random();
11         HashSet<Integer> set = new HashSet<>();
12
13         System.out.println("오늘의 로또 번호는 : ");
14
15         for(;;) {
16             set.add((rand.nextInt(45)));
17
18             if(set.size() == 7){
19                 break;
20             }
21         }
22         System.out.println(set+"입니다");
23
24     }
25 }
```

The console output at the bottom shows the execution results:

```
<terminated> RandomExam3 [Java Application] C:\Program Files\Java\jdk-17.0.4.1\bin\javaw.exe (2022. 1
오늘의 로또 번호는 :
[33, 2, 35, 21, 7, 42, 28]입니다
```

```
package ArrayList;
```

```
import java.util.Random;
```

```
import java.util.HashSet;
```

```
public class RandomExam3 {
```

```
    public static void main(String[] args) {  
  
        Random rand = new Random();  
        HashSet<Integer> set = new HashSet<>();  
  
        System.out.println("오늘의 로또 번호는 : ");  
  
        for(;;) {  
            set.add((rand.nextInt(45)));  
  
            if(set.size() == 7){  
                break;  
            }  
        }  
        System.out.println(set+"입니다");  
  
    }
```

```
}
```