

# (디지털컨버전스) 스마트 콘텐츠와 웹 융합 응용 SW개발자 양성과정

-6일차 학습 및 질문 노트-

강사 - Innova Lee(이상훈)

gcccompil3r@gmail.com

학생 - Kyeonghwan Lee(이경환)

airtrade7@naver.com

## ■ array (배열)

```

1 public class ArrayTest {
2     public static void main(String[] args) {
3         // 배열은 왜 써야 할까 ?
4         // 동일한 데이터 타입의 변수가 여러개 필요할때
5         // 일일이 int a, b, c, d, e, f, g, h, i ... z 까지 해봐야 26개 밖에 안됨
6         // 만약 회사에서 직원 1000명을 관리해야 한다 가정한다면
7         // 이것을 일일이 변수로 선언한다면 죽을 것이다.
8         // 당연히 배열을 만들어서 관리해야할 것이다.
9         int arr[] = { 1, 2, 3, 4, 5 };
10        // int num1 = 1, num2 = 2, num3 = 3, num4 = 4, num5 = 5;
11        // 데이터가 많으면 많을수록 단일 변수 선언은 지옥을 체험하게 해줄 것이다.
12        // 그러니 우리는 심신의 안정을 위해 배열을 사용해야할 것이다.
13
14        // 배열을 만드는 방법
15        // 1. stack에 할당하는 방법(지역 변수)
16        // 1-1. 일단은 배열의 데이터 타입(int 같은)을 적는다.
17        // 1-2. 배열의 이름이 될 변수명을 적는다.
18        // 1-3. 배열임을 알리기 위해 []을 변수 옆에 적어준다.
19        // 1-4. 필요하다면 배열의 값들을 초기화한다.
20        // (이때 원소로 지정한 숫자에 따라 배열의 길이가 지정된다)
21        // * 가변으로 구성하고 싶다면 new를 사용해야 하는데 이것은 다음주에 학습하도록 한다.
22

```

```

23        // 아래와 같은 데이터를 살펴보자
24        // int arr[] = { 1, 2, 3, 4, 5 };
25        // 위 데이터는 아래와 같은 형식으로 저장된다.
26
27        // -----
28        // arr | 1 | 2 | 3 | 4 | 5 |
29        // -----
30        // [0] [1] [2] [3] [4]
31        // 배열의 인덱스(방) 번호는 0번부터 시작함에 주의하도록 한다.
32        // 그러나 방 번호가 순차적으로 증가하기 때문에
33        // for 문이나 while 문등의 반복문과의 혼합구성에 있어 매우 탁월하다.
34
35        for (int i = 0; i < 5; i++) {
36            // System.out.printf("arr[%d] = %d\n", i, arr[i]);
37            System.out.println("arr[" + i + "] = " + arr[i]);
38        }
39    }
40 }
41
42 // stack(지역변수)에 할당한다는 것은 지역변수로 처리함을 의미합니다.
43 // 그렇기 때문에 나중에 매서드나 클래스를 학습한 이후 스택에 할당하면
44 // 해당 매서드 혹은 클래스 내부에서만 해당 배열이 활성화됩니다.
45
46 // arr[0] = x, arr[1] = y, arr[2] = z, arr[3] = k
47 // System.out.println("arr[" + 0 + "] = " + x + ", arr[" + 1 + "] = " + y + ", arr[ ..... 지옥]);
48 // System.out.printf("arr[%d] = %d, arr[%d] = %d, arr[%d] = %d, arr[%d] = %d\n",
49 //                     0, arr[0], 1, arr[1], 2, arr[2], 3, arr[3])

```

\*array: 동일한 데이터 타입의 변수가 여러 개 필요할 때 사용

## ■ continue

```
1  public class ContinueTest {
2      public static void main(String[] args) {
3          for (int i = 0; i < 10; i++) {
4              if (i % 2 == 0) {
5                  // continue 를 만나면 아래쪽에 진행해야하는 코드가 남아있더라도
6                  // 무조건 for loop의 최상단으로 이동하게 된다.
7                  // 그러므로 증감식이 진행된다.
8                  continue;
9              }
10
11             System.out.println("i = " + i);
12         }
13     }
14 }
```

\*contiune: 아래쪽에 진행해야 하는 코드가 남아있더라도 최상단으로 이동하게 된다.

## ■ Switch

```

1  import java.util.Scanner;
2
3  public class SwitchTest {
4      public static void main(String[] args) {
5          System.out.println("저희 상점에 방문해주셔서 감사합니다. 물건을 고르십시오 호갱님!");
6
7          // Boolean 이란 참, 거짓을 표현할 수 있는 데이터타입이다.
8          Boolean isTrue = true;
9
10         Scanner scan = new Scanner(System.in);
11         int num;
12
13         while (isTrue) {
14             System.out.print("숫자를 눌러 물건을 받으세요: ");
15
16             num = scan.nextInt();
17
18             String str = "hi";
19
20             // 입력된 키보드 값에 따라 적절한 처리를 하게 된다.
21             // 키보드 값에 따라 처리하는 루틴은 case x에 해당한다.
22             // 0번이 눌렀다면 case 0, 1번이라면 case 1과 같은 형식이다.
23
24             // switch에서 판정에 사용하는 것이 String이라면
25             // case에서 사용하는것도 String으로 맞춰서 동작시킬 수 있다.

```

- \* Switch: 지정 값을 입력하여 값을 도출함
- \* Boolean: 참, 거짓을 표현할 수 있는 데이터 타입
- \* break: 더 이상 밑으로 내려가지 않고 이 시점에서 종료시키는 역할
- \* default: 기본값(지정한 입력 값 외에 발생하는 경우의 값)
- \* Switch에 사용되는 데이터 타입과 case에 사용하는 데이터 타입을 일치시킬 필요가 있다.

```

26         switch (num) {
27             // 문자 낱개로는 가능함(홀따옴표)
28             // 현재는 숫자값이라서 현재는 문장 여러개의 문자열(쌍따옴표는 불가능함)
29             case 0:
30                 System.out.println("탈출합니다.");
31                 isTrue = false;
32                 break;
33             case 1:
34                 System.out.println("비누를 장바구니에 담았습니다.");
35                 // break;
36                 // break는 더 이상 밑으로 내려가지 않고
37                 // 이 시점에서 종료할 수 있게 도와주는 역할을 한다.
38             case 2:
39                 System.out.println("신발을 장바구니에 담았습니다.");
40                 break;
41             case 3:
42                 System.out.println("에어팟을 장바구니에 담았습니다.");
43                 break;
44             default:
45                 // 이 default라는 녀석은 말 그대로 기본값에 해당함
46                 // 우리가 예상치 못한 입력이 존재할 수 있음
47                 // 이 경우에 활용하는것이 default라고 보면 됩니다.
48                 System.out.println("그런건 없습니다!");
49                 break;
50         }
51     }

```