

# 배열 (Array) 기본

# 배열이란

- 자료형이 같은 다수의 변수를 효율적으로 관리하기 위한 자료형
- 효율적인 관리란 데이터를 읽고 쓰기 용이하다는 의미
- 우리가 아는 모든 자료형은 모두 배열로 사용이 가능

## 배열을 사용하는 이유

Int형 변수 100개를 선언하고 사용한다고 해보자. 변수 100개를 타이핑 한다는 것도 문제이고, 100개의 변수에 각각 다른 변수명을 주는 것도 쉬운 일이 아니다. 또한, 100개의 변수명을 기억하는 것도 불가능에 가깝다. 더욱이 100개의 변수를 선언했다고 해도, 100개의 변수에 들어간 수를 모두 +1 시킨다고 한다면 어떻게 해야 할까..

# 배열의 선언 및 생성

```
int[] array; // 정수를 담을 수 있는 이름이 array인 배열을 선언  
array = new int[5]; // 배열의 생성(배열은 초기화라 부르지 않는다)
```

- 위의 문장을 통해 배열 array는 정수를 저장할 수 있는 공간 5개를 생성한다. 또한, 생성된 공간 5개를 모두 0으로 초기화한다.
- 정수형 배열은 0으로 초기화, 실수형 배열은 0.0으로 초기화, 문자열 배열은 null로 초기화

```
int[] array = new int[5]; // 배열의 선언과 생성을 동시에.  
String[] arr1 = new String[3];  
Double[] arr2 = new double[10];  
int arr3[] = new int[3];
```

## 배열의 또 다른 초기화

```
int[] arr = new int[5];
```

위와 같이 배열을 선언하고 생성하면 정수 5개를 저장할 공간이 만들어지고, 각각 0으로 초기화가 된다. 그렇다면 생성과 동시에 0이 아닌 다른 수로 초기화를 하기 위해선 어떻게 해야 할까.

```
Int[] arr1 = {1, 2, 3};
```

```
String[] arr2 = {"aa", "bb", "cc"};
```

이렇게 사용하면 원하는 대로 초기화를 할 수 있으며, 자바가 알아서 필요한 만큼 공간도 생성도 해준다. 단, 실무에서 이렇게 배열을 생성하는 것은 극히 드물다. 하지만 꼭 기억해야 할 내용이다.

# 배열과 기본 자료형의 차이(1)

```
int[] arr = new int[5];
```

위의 코드 실행 결과를 아래와 같이 표현할 수 있다.

arr =>	0	0	0	0	0
--------	---	---	---	---	---

일반적인 변수의 경우..

```
int num0 = 0;
```

0
num0

```
int num1 = 0;
```

0
num1

```
int num2 = 0;
```

0
num2

```
int num3 = 0;
```

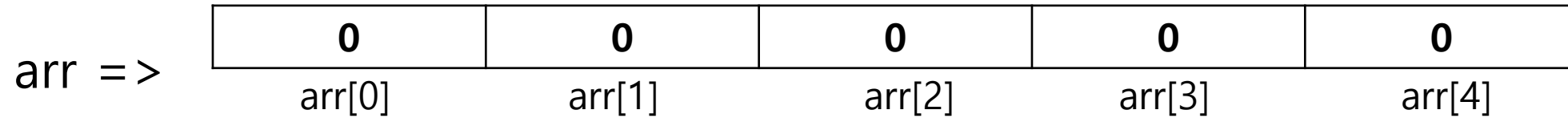
0
num3

```
int num4 = 0;
```

0
num4

- 정수형 변수 5개를 만들면 각각에 명칭이 있다.
- 배열을 통해 만들어진 공간은 그 공간 전체의 이름만 갖는다.
- 그렇다면 배열 하나 하나는 어떻게 불러야 할까. 혹시 어떻게 접근해야 할까.

## 배열과 기본 자료형의 차이 (2)



- 배열 생성을 통한 공간 하나하나를 '배열의 요소' 라고 부른다.
- 배열은 0번째 요소부터 시작한다.

### 배열의 값 읽기

int num = 5; 일 때 우리는 num값을 출력하기 위해 print(num); 이라고 쓰고 결과는 5가 출력됨을 알고 있다. 그렇다면 배열 arr에 저장된 값을 어떻게 출력해볼 수 있을까. 아니.. 출력이 된다면 어떻게 출력하는게 맞는지 생각해보자.

print(arr); // 출력결과를 예상해보자. 예상한 실행결과가 나타났는가?

# 배열의 값 읽기와 쓰기(1)

arr =>

0	0	0	0	0
arr[0]	arr[1]	arr[2]	arr[3]	arr[4]

## 배열의 값 읽기

- 배열의 값을 읽기 위해 `print(arr);` 을 실행하면 알 수 없는 문자열이 나온다.

## 왜 이런 결과가 나왔을까

다시 잘 생각해보자. 배열은 무엇이었는가. `int[] arr = new int[5];`라는 코딩의 의미는 정수를 저장할 수 있는 공간 5개를 만든 것이다. 다시 말해, `print(arr);`는 5개의 변수에 들어있는 수를 출력하라는 말이 된다. 그렇다면 배열요소 각각에 1,2,3,4,5 가 들어가 있다면 1,2,3,4,5로 출력이 되어야 하는 건가? 혹은 1 2 3 4 5로 출력이 되어야 하는 건가? 아니면 0번째 요소인 1이 출력이 되어야 하는 건가. 불행히도 이 모든 방법은 납득이 가지 않는다..

## 배열의 값 읽기와 쓰기(2)

arr =>

0	0	0	0	0
arr[0]	arr[1]	arr[2]	arr[3]	arr[4]

### 배열의 값 읽기

`int[] arr = new int[5];` 는 다시 말해서 `arr`을 출력할 수가 없다. 정확히 말하자면 출력을 해도 우리에게는 의미없는 출력값만 던져줄 뿐이다. 결론을 말하자면 5개의 공간의 값을 읽기 위해서는 하나씩 접근할 수 밖에 없다.

- `print(arr[0]);`
- `print(arr[4]);`
- `print(arr[5]);` // 오류발생! 주의 요함!

자,, 배열 요소는 0 ~ 5까지이다. 이를 한번에 모든 요소로 접근해서 쉽게 읽을 수 있는 방법이 있다. 생각나는 것이 있는가.



## 배열의 값 읽기와 쓰기(3)

arr =>

0	0	0	0	0
arr[0]	arr[1]	arr[2]	arr[3]	arr[4]

### 배열의 값 쓰기

`int[] arr = new int[5];` 에서 각 요소 마다 저장된 값을 변경하기 위해서는 읽기와 마찬가지로 하나씩 해결해야 한다.

- `arr[0] = 1;`
- `arr[2] = 2;`
- `arr[5] = 5; //오류발생! 주의 요함!`
- `arr[3] = "abc"; //오류발생`