



배열

배열이란?

배열은 "일정한 크기의 연속적인 공간"입니다. 쉽게 말해, 여러 개의 값을 하나의 변수로 관리할 수 있는 데이터 구조입니다. 예를 들어, 변수가 하나의 값을 저장하는 상자라면, 배열은 여러 개의 상자를 일렬로 늘어놓은 것입니다.

배열 선언과 생성

배열을 사용하기 위해서는 먼저 배열을 선언하고 생성해야 합니다. 배열 선언은 배열의 데이터 타입과 배열의 이름을 지정하는 것이고, 생성은 배열에 메모리 공간을 할당하는 것입니다.

다음은 정수형 배열을 선언하고 생성하는 예시입니다.

```
// 배열 선언
int[] arrayBox;

// 배열 생성 (크기가 9인 배열 생성)
arrayBox = new int[9];
```

여기서 `int` 는 배열에 담을 수 있는 데이터의 타입을 나타내며, `arrayBox` 는 배열의 이름입니다. `new int[9]` 는 크기가 9인 정수형 배열을 생성하는 부분입니다.

배열 초기화

배열에 값을 넣는 것을 초기화라고 합니다. 배열의 각 요소에 값을 하나씩 대입할 수 있습니다.

```
arrayBox[0] = 1;
arrayBox[1] = 2;
arrayBox[2] = 3;
arrayBox[3] = 4;
arrayBox[4] = 5;
arrayBox[5] = 6;
arrayBox[6] = 7;
```

```
arrayBox[7] = 8;
arrayBox[8] = 9;
```

위 코드는 `arrayBox` 배열의 각 인덱스에 1부터 9까지의 값을 차례로 넣는 과정입니다.
또는 배열을 선언하면서 동시에 초기화할 수 있습니다.

```
int[] arrayBox = {1, 2, 3, 4, 5, 6, 7, 8, 9};
```

배열의 인덱스

배열의 인덱스는 0부터 시작합니다. 즉, `arrayBox[0]` 은 배열의 첫 번째 요소이고, `arrayBox[8]` 은 배열의 마지막 요소입니다. 이렇게 인덱스를 통해 배열의 특정 위치에 접근할 수 있습니다.

배열의 특징

1. **고정된 크기:** 배열은 생성될 때 크기가 정해지며, 이후에는 크기를 변경할 수 없습니다.
2. **인덱스를 통한 접근:** 배열의 각 요소는 인덱스를 통해 접근합니다.

반복문을 사용하여 배열 출력하기

배열을 사용할 때 반복문을 활용하면 편리하게 배열의 모든 요소에 접근할 수 있습니다.

```
for (int i = 0; i < arrayBox.length; i++) {
    System.out.println("arrayBox[" + i + "] = " + arrayBox[i]);
}
```

위 코드에서는 `arrayBox.length` 를 사용하여 배열의 크기를 가져옵니다. `i` 가 0부터 배열의 길이보다 작을 때까지 반복하면서 각 요소의 값을 출력합니다.

배열을 활용한 예제

여기서는 배열의 값을 모두 더하는 예제를 보겠습니다.

```
int sum = 0;
for (int i = 0; i < arrayBox.length; i++) {
    sum += arrayBox[i];
}
System.out.println("배열의 모든 요소의 합은: " + sum);
```

이 코드는 `for` 문을 통해 배열의 모든 요소를 더하여 총합을 출력합니다.

마무리

이제 여러분은 배열의 기본적인 개념, 선언, 생성, 초기화, 그리고 반복문을 통한 배열의 활용법을 알게 되었습니다. 다음에 실제로 코드를 작성해보면서 배열을 더욱 익숙하게 만들어보세요.