

K G 아 이 티 뱅 크

J A V A

J A V A

InetAddress

## InetAddress

### ❖ Java.net 패키지에 포함되어 있는, IP주소를 위한 클래스

- 물리적으로 떨어진 컴퓨터는 네트워크를 통하여 통신함
- 이를 쉽게 쓰기 위하여 도메인 이름을 적용하여 쓰기도 함

### ❖ IP주소가 있어야만 어디와 통신을 하는지 지정할 수 있음

- 대부분의 컴퓨터는 DHCP이기 때문에 매번 획득이 필요함

```
import java.net.InetAddress;
import java.net.UnknownHostException;
try {
    // 생성자가 protected로 설정되어 있음
    InetAddress add1 = InetAddress.getByName("주소");
    InetAddress add2 = InetAddress.getByAddress(ipArr);
    InetAddress[] adds = InetAddress.getAllByName("주소");
    InetAddress local = InetAddress.getLocalHost();
} catch (UnknownHostException e){
    System.out.println(e);
}
```

## InetAddress

- ❖ **getByName("도메인이름") : 연결시의 IP주소를 획득함**
  - 기본값처럼 사용되는 메서드이며 객체를 하나 반환함
- ❖ **getAllByName("도메인이름") : 모든 IP주소를 획득함**
  - 연결할 수 있는 모든 IP를 알아내고자 할 때 사용함
- ❖ **getByAddress(ipArr) : 지정한 IP에 대한 것을 획득함**
  - 특정 IP에 대한 동작이 필요할 때 이를 사용함

// getByName을 기본으로 사용하는 것이 권장됨

```
InetAddress add = InetAddress.getByName("www.google.com");  
System.out.println(add.getHostName()); // 도메인명 반환  
System.out.println(add.getHostAddress()); // IP주소 반환  
byte[] addIp = add.getAddress(); // IP를 배열로 반환  
InetAddress oth = InetAddress.getByAddress(addIp);
```

// getAllByName은 InetAddress 객체가 포함된 배열을 반환함

```
InetAddress[] adds = InetAddress.getAllByName("www.google.com");  
// 각 인덱스의 사용법은 동일하며, 필요시 컬렉션으로 형변환 진행  
List<InetAddress> list = new ArrayList<>(Arrays.asList(adds));
```

## InetAddress

- ❖ Local Host를 다룰 때는 메서드가 일부 다르니 주의해야 함
  - 특히, IP주소가 loopback인지 여부가 중요함
- ❖ `getLocalHost()` : 현재 컴퓨터의 `InetAddress` 객체를 반환
  - 동일하게 `getHostName`, `getHostAddress`를 쓸 수 있음
- ❖ `isLoopbackAddress()` : loopback 주소인지를 반환함
  - Loopback은 **네트워크를 경유하지 않으니 주의**해야 함

```
// 같은 객체이기에 사용법은 동일, 생성방식은 다름
InetAddress local = InetAddress.getLocalHost();

// IP가 loopback인지를 반드시 체크해줘야 함
System.out.println(add.getHostName());
if (local.isLoopbackAddress()) {
    System.out.println("시스템에 문제가 있음!");
}
else {
    System.out.println("외부와 통신이 가능함!");
}
```