## 프로그램 실행과 경로

C:``\Users\bitcamp > a.exe (enter)

1. 현재 경로에서 찾는다. Unix의 경우 현재 경로를 명시해야 한다. 예) ./a.exe '.' 은 현재 폴더, '..' 은 상위 폴더를 가리킨다. ../a.exe 는 `C:\users\a.exe` 를 나타낸다. .. / .. / a.exe는 `C:\a.exe` 를 나타낸다. .. / .. / dev / a / a.exe 는 `C:\dev\a\a.exe` 를 나타낸다.
2. OS의 PATH 환경 변수에 등록된 경로를 순서대로 찾는다. echo $PATH (Window : echo %PATH%) 참고로 윈도우는 경로와 경로 사이 ';', 맥은 ':' 이다.

## 프로그램 실행 : Windows

a.exe(컴파일 된 기계어 명령 실행파일) ----------> > a.exe (a도 가능)
확장자를 빼고 파일명으로만 실행할 때

1. a.com 을 찾는다.
2. a.exe 를 찾는다.
3. a.bat 을 찾는다.
   a.com 과 a.exe 는 기계어 파일, a.bat은 OS가 이해하는 스크립트로 작성된 명령이 들어있는 파일이다.
   **따라서 컴파일 불필요하다.**

a.doc ----------> > a.doc
a.doc는 기계어 파일이 아니라 직접 실행이 불가능하다.

1. .doc 확장자와 **연결된 프로그램을 실행한다.**
2. **그 프로그램이 a.doc 파일을 읽는다.**

## 프로그램 실행 : Unix

**Unix에서는 실행파일에 대해 사용자가 실행권한을 가지고 있어야만 실행할 수 있다.**
$ ls -al
rw-r--r-- a // 실행파일에 대해 사용자에게 실행 권한을 부여하지 않음. 실행 불가.
*rw- : user 권한 r-- : group 권한 r-- : other 권한 a : 실행파일명*
$ chmod 744 a // 실행 권한을 부여

7 4 4 a

111 100 100

rwx r-- r--

$ a // 실행 가능

a (컴파일 된 기계어 명령 실행파일) ------------------>

*a PATH 환경변수에 등록된 경로 따라가며 찾는다. 따라서* ** ./a 와 같이 현재 폴더에 있는 파일을 실행하고 싶다면 경로를 명시해야 한다.**

## Shell Script와 기타 Script 실행

1. Windows OS

- Shell Script : **> a.bat (enter)** || **> a (enter)** => OS 내부의 인터프리터를 사용해서 스크립트를 실행한다.
- 기타 스크립트 : **node hello.js** => node라는 JavaScript를 실행시키는 "인터프리터" 가 hello.js 라는 JavaScript 파일을 실행한다.

2. Unix

- Shell Script : **$zsh a.sh** => zsh는 스크립트 인터프리터, a.sh 는 스크립트 파일. Shell Script를 자동 실행하려면 a.sh 스크립트 파일 첫번째에 #!/bin/zsh 라고 저장한다. #!/bin/zsh 는 이 파일을 실행시킬 Shell Script 인터프리터를 설정한 것.
  $chmod 744 a.sh로 스크립트 파일에 대한 실행 권한을 부여한 후 $./a.sh 로 Shell Script 파일을 실행한다.

## Collections

*basic.ex03~*

`public E[] toArray(E[] arr)` 을 사용할 때 배열 size를 증가시키는 경우가 있으므로 **리턴한 배열 주소를 사용하도록 한다.**

```
// list의 자리에는 배열 또는 java.lang.Iterable 규칙에 따라 구현한 객체만이 올 수 있다.
for (Member m : list) {
        ...
}
```

ArrayList의 **contains()**

- 해당 인스턴스와 같은 객체가 있는지 알아낸다.
- 단 인스턴스 주소를 비교하는 것이 아니라, equals()의 결과가 true인지 비교한다.

  ArrayList의 **indexOf(값)**
- 목록에 같은 값을 가진 객체의 인덱스를 알아낸다.
- 값을 비교할 때는 contains()와 마찬가지로 equals()의 리턴 값이 true인 경우 같은 값으로 간주한다.

**Iterator를 사용하면 컬렉션의 종류에 상관없이 일관된 방식으로 값을 조회할 수 있다.**

HashSet은 equals()와 hashCode()를 함께 비교하기 때문에 두 메서드 다 오버라이딩 해야 한다.

**HashSet이 중복여부를 검사할 때는 hashCode()와 equals()의 리턴값으로 판단한다.**

## HashSet, HashMap