**package** class\_;

**public** **class** StringMain {

**public** **static** **void** main(String[] args) {

String a = "apple"; // String은 문자열 리터럴 생성이 가능하다.

//다른 그 어떠한 클래스도 new를 생성해야줘야 하는데, String만 유별나게 문자열 리터럴 생성이 가능하다.

String b = "apple"; // 똑같은 문자열 리터럴은 메모리에 1번만 생성된다.

따라서, apple이라는 문자열은 하나지만 두개의 객체(String a, String b)를 갖게되는 것이다.

**heap영역**

//리터럴을 잡으면 메모리의 heap영역에 자리한다. new연산자도 마찬가지)

//heap영역의 특징은 이름이 없기 때문에 항상 주소를 준다.

a ---------🡪 “apple”

a에 있는 주소를 apple에게 전달

리터럴(위쪽 영역)

뉴(아래쪽 영역)

**if** (a == b) //주소값을 물어보는 거

System.***out***.println("a와 b의 참조값은 같다");

**else**

System.***out***.println("a와 b의 참조값은 다르다");

**if** (a.equals(b)) //문자열을 물어보는 거

System.***out***.println("a와 b의 문자열은 같다");

**else**

System.***out***.println("a와 b의 문자열은 다르다");

System.***out***.println();

String c = **new** String("apple");

String d = **new** String("apple");

**if** (c == d)

System.***out***.println("c와 d의 참조값은 같다");

**else**

System.***out***.println("c와 d의 참조값은 다르다");

**if** (c.equals(d))

System.***out***.println("c와 d의 문자열은 같다");

**else**

System.***out***.println("c와 d의 문자열은 다르다");

System.***out***.println();

String e = "오늘 날짜는 " + 1991 + 7 + 16;

System.***out***.println("e = " + e);

System.***out***.println("문자열 크기 =" + e.length()); // 14

//배열.length는 ()이 필요치 않다

System.***out***.println("첫번째 문자 = " + e.charAt(0)); //오

System.***out***.println("두번째 문자 = " + e.charAt(1)); //늘

System.***out***.println("마지막 문자 = " + e.charAt(13)); //6

System.***out***.println("부분 문자열 추출 = " + e.substring(7)); //1991716

System.***out***.println("부분 문자열 추출 = " + e.substring(7, 11)); // 1991 - 7번부터 11이전까지

System.***out***.println("대문자 변경 = " + "Hello".toUpperCase()); //HELLO

System.***out***.println("소문자 변경 = " + "Hello".toLowerCase()); //hello

System.***out***.println("문자열 검색 = " + e.indexOf("짜")); // 4

System.***out***.println("문자열 검색 = " + e.indexOf("날짜")); // 3

System.***out***.println("문자열 검색 = " + e.indexOf("개바부")); // -1 - 찾는 문자가 없을 시 -1로 표시됨

System.***out***.println("문자열 검색 = " + e.indexOf("1")); // 7 - 찾는 문자가 여러번 등장하면 제일 앞에 문자의 위치

System.***out***.println("문자열 검색 = " + e.indexOf("1", 0)); // 7

System.***out***.println("문자열 검색 = " + e.indexOf("1", 8)); // 10 - 8번부터 시작해 10번째 위치에 1이 있다

System.***out***.println("문자열 치환 = " + e.replace("날짜", "일자")); //문자열 치환 = 오늘 일자는 1991716

}

}

/\*

문자열은 편집이 안되므로 메모리 4번 생성된다 JVM에 의해서 삭제 시 Garbage Collector로 보낸다 Garbage

Collector가 실행되면(휴지통 비우기) 컴퓨터는 멈춘다

"오늘 날짜는 " "오늘 날짜는 1991" "오늘 날짜는 19917" 012 789 e ---> "오늘 날짜는 1991716"

\*/

[결과]

a와 b의 참조값은 같다

a와 b의 문자열은 같다

c와 d의 참조값은 다르다

c와 d의 문자열은 같다

e = 오늘 날짜는 1991716

문자열 크기 =14

첫번째 문자 = 오

두번째 문자 = 늘

마지막 문자 = 6

부분 문자열 추출 = 1991716

부분 문자열 추출 = 1991

대문자 변경 = HELLO

소문자 변경 = hello

문자열 검색 = 4

문자열 검색 = 3

문자열 검색 = -1

문자열 검색 = 7

문자열 검색 = 7

문자열 검색 = 10

문자열 치환 = 오늘 일자는 1991716