

자주 쓰는 Java API

▶ Object Class

✓ Object Class

모든 클래스의 **최상위 부모 클래스**.

모든 클래스는 Object를 상속하고 있고, Object가 제공하는 메서드를 사용, 오버라이딩 가능.

✓ Object 주요 메서드

메서드	설명
boolean equals(Object obj)	현재 객체와 전달 받은 객체가 같은지 비교
int hashCode()	객체의 해시 코드 반환
String toString()	객체의 정보를 문자열 형태로 반환

▶ Object Class

✓ boolean Object.equals(Object obj)

현재 객체를 참조하는 주소와 전달 받은 객체 참조 **주소가 같은 비교**(== 연산)
(**동일**한 인스턴스를 참조하는지 비교)

✓ equals() 오버라이딩

비교하려는 두 객체의 **필드 값이 같은지 비교**
(**동등**한 객체인지 비교)

→ 객체 지향 프로그래밍에선 수많은 객체가 존재하기 때문에
서로 다른 두 객체를 비교하는 경우 동일 객체를 참조하는지 비교하기보단
두 객체가 같은 값을 가지고 있는지 비교해서 같다, 다르다를 판별하도록 함

▶ Object Class

✓ equals() 오버라이딩 예시

```
public class Person {  
    private String name;  
    private int age;  
  
    @Override  
    public boolean equals(Object obj) {  
        if (this == obj) return true; // 같은 객체인지 확인  
        if (obj == null || getClass() != obj.getClass()) return false; // null 체크 및 클래스 타입 체크  
  
        Person person = (Person) obj; // 타입 캐스팅  
        return age == person.age && name.equals(person.name); // 필드 값 비교  
    }  
}
```

▶ Object Class

✓ `int Object.hashCode()`

객체를 식별하는 값. 객체의 메모리 주소를 이용해서 생성

✓ `hashCode()` 오버라이딩

`equals()` 메서드 오버라이딩 시 `hashCode()`도 같이 오버라이딩 하도록

Java 공식 문서에 명시되어 있음 (규약으로 정해짐, 강제성은 없지만 지키는 것이 좋음)

(동등한 값을 가지는 객체는 같은 식별 번호를 가져야 한다)

▶ Object Class

✓ hashCode() 오버라이딩 예시

```
public class Person {  
    private String name;  
    private int age;  
  
    // equals() 오버라이딩  
    @Override  
    public boolean equals(Object obj) { ... }  
  
    @Override  
    public int hashCode() {  
        return Objects.hash(name, age);  
        // equals()를 오버라이딩할 때는 hashCode()도 함께 오버라이딩하는 것이 좋다.  
    }  
}
```

▶ String Class

✓ String Class

- 자바에서 제공하는 문자열 자료형 (기본 자료형 X)
- 문자열 생성 및 관련 메서드 제공
- 문자열 값 수정 불가능(final로 선언됨) == **immutable(불변)**
수정 시 수정된 String 객체가 새로 할당 되어 새 주소를 넘김

```
@Stable  
private final byte[] value;
```

String Class 필드 중 문자열 값을 저장하는 value.

- final로 선언되어 있음
- 문자열을 byte 배열로 변환하여 저장

► String Class

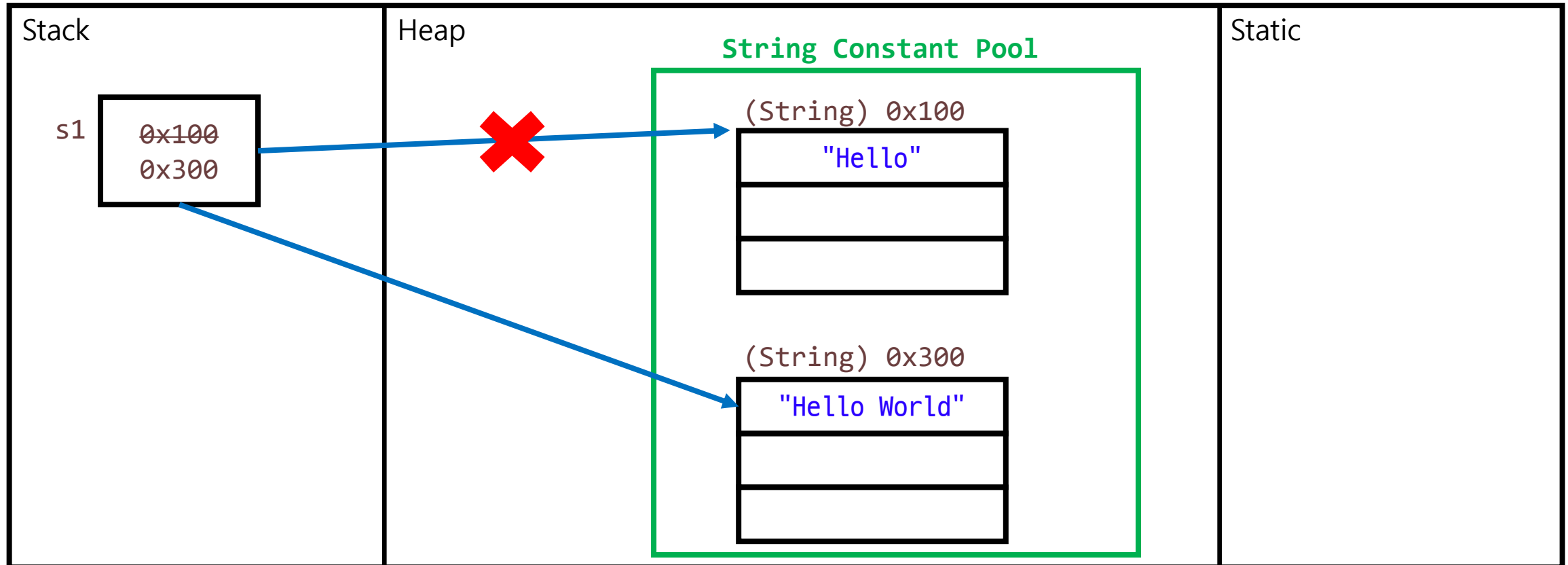
[String 불변성]

```
String s1 = "Hello";
```

```
s1 = s1 + " World";
```

String 객체는 문자열을 final(상수)로 저장함

그래서 새로운 문자열을 추가하게 되면
기존 객체가 아닌 새로운 객체를 만들어서 참조하게함



▶ String 제공 메서드

메서드	설명
length()	문자열의 길이를 반환합니다.
charAt(int index)	지정된 인덱스의 문자를 반환합니다.
substring(int beginIndex, int endIndex)	지정된 범위의 문자열을 반환합니다.
indexOf(String str)	특정 문자열이 처음으로 등장하는 인덱스를 반환합니다.
lastIndexOf(String str)	특정 문자열이 마지막으로 등장하는 인덱스를 반환합니다.
toLowerCase()	문자열을 소문자로 변환하여 반환합니다.
toUpperCase()	문자열을 대문자로 변환하여 반환합니다.
trim()	문자열의 앞뒤 공백을 제거합니다.
replace(char oldChar, char newChar)	특정 문자를 다른 문자로 대체합니다.
equals(Object obj)	두 문자열이 같은지 비교합니다.
equalsIgnoreCase(String anotherString)	대소문자 구분 없이 두 문자열이 같은지 비교합니다.
split(String regex)	정규 표현식을 기준으로 문자열을 분리하여 배열로 반환합니다.
contains(CharSequence s)	특정 문자열이 포함되어 있는지 확인합니다.
startsWith(String prefix)	특정 접두사로 시작하는지 확인합니다.
endsWith(String suffix)	특정 접미사로 끝나는지 확인합니다.

▶ String 대체 클래스

✓ StringBuffer

문자열 값 수정 가능, mutable(가변)

수정, 삭제 등이 **기존 문자열에 수정**되어 적용

기본 16문자 크기로 지정된 버퍼를 이용하며 크기 증가 가능

쓰레드 safe기능 제공(성능 저하 요인)

✓ StringBuilder

StringBuffer와 동일하나 쓰레드 safe기능을 제공하지 않음(빠름)

► StringBuilder 메서드

메서드	설명
append(String s)	문자열을 끝에 추가
insert(int offset, String s)	특정 위치에 문자열 삽입
replace(int start, int end, String s)	지정한 범위의 문자열을 다른 문자열로 변경
delete(int start, int end)	지정한 범위의 문자열 삭제
reverse()	문자열을 뒤집음
substring(int start)	지정한 위치부터 끝까지 문자열 반환
substring(int start, int end)	지정한 범위의 문자열 반환
indexOf(String s)	특정 문자열의 첫 번째 위치 반환
lastIndexOf(String s)	특정 문자열의 마지막 위치 반환
length()	문자열의 길이 반환
capacity()	내부 버퍼 크기 반환
trimToSize()	버퍼 크기를 문자열 길이에 맞춤
toString()	StringBuilder를 String으로 변환

▶ Wrapper Class

Primitive Data Type(기본 자료형)을 인스턴스화 시켜주는 클래스

ex) 매개 변수, 반환형이 Object 타입 / 컬렉션 드

Primitive Data Type	Wrapper Class
boolean	Boolean
byte	Byte
char	Character
short	Short
int	Integer
long	Long
float	Float
double	Double

▶ Wrapper 클래스

✓ String을 기본 자료형으로 바꾸기

```
byte b = Byte.parseByte("1");  
short s = Short.parseShort("2");  
int i = Integer.parseInt("3");  
long l = Long.parseLong("4");  
float f = Float.parseFloat("0.1");  
double d = Double.parseDouble("0.2");  
boolean bool = Boolean.parseBoolean("true");  
  
char c = "abc".charAt(0);
```

▶ Wrapper 클래스

✓ 기본 자료형을 String으로 바꾸기

```
String b = Byte.valueOf((byte)1).toString();  
String s = Short.valueOf((short)2).toString();  
String i = Integer.valueOf(3).toString();  
String l = Long.valueOf(4L).toString();  
String f = Float.valueOf(0.1f).toString();  
String d = Double.valueOf(0.2).toString();  
String bool = Boolean.valueOf(true).toString();  
String ch = Character.valueOf('a').toString();
```