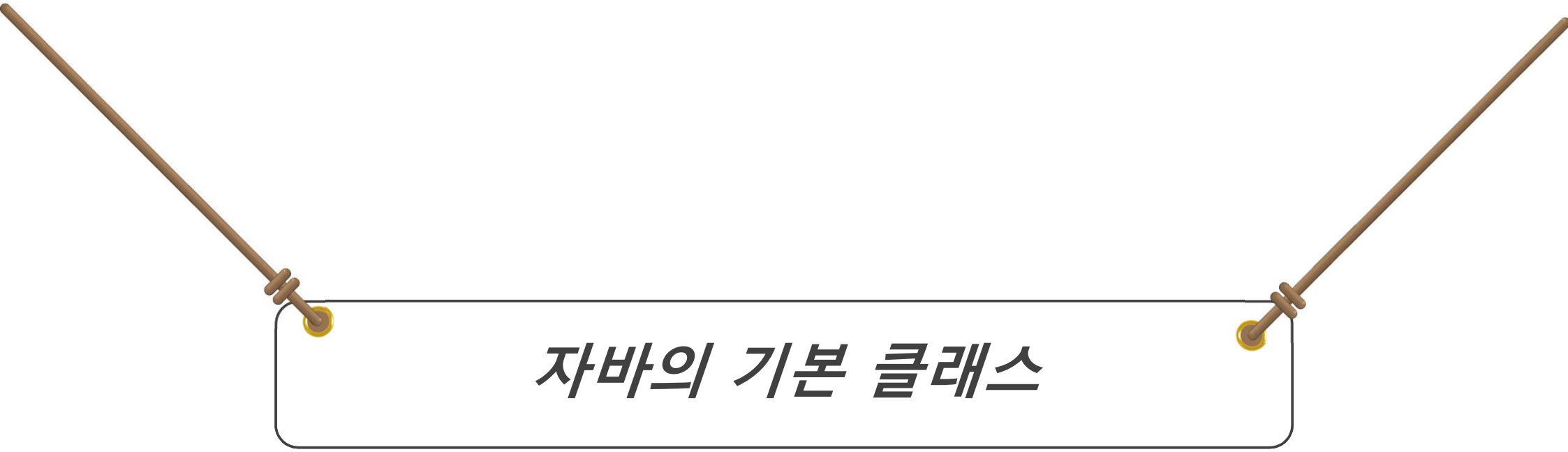
자바의 기본 클래스

자바의 기본 클래스

- 기본 자료형의 값을 감싸는 래퍼 클래스
  - 기본 자료형의 값을 감싸는 클래스
  - Boolean, Integer, Long, Double 등
  - 기본 자료형(int, double, boolean 등)의 값을 인스턴스로 표현해야 할 경우 사용
  - toString 메소드를 기본적으로 오버라이딩하고 있으므로, println 메소드의 인자로 전달하여 인스턴스의 값 출력 가능

```
public class UserWrapperClass {
    public static void main(String[] args) {
        Integer iInst = new Integer(3);
        showData(iInst);
        Double dInst = new Double(3.14);
        showData(dInst);
    }

    public static void showData(Object obj) {
        System.out.println(obj);
    }
}
```

자바의 기본 클래스

- 래퍼 클래스의 두 가지 기능
  - Boxing : 값을 인스턴스로 감싸는 기능, 인스턴스의 생성을 통해 기능 사용
  - Unboxing : 인스턴스에서 값을 로드하는 기능, 래퍼 클래스에 정의된 메소드의 호출을 통해 기능 사용

```
public class BoxingUnboxing {
    public static void main(String[] args) {
        Integer iObj = new Integer(10); // 박싱
        Double dObj = new Double(3.14); // 박싱
        System.out.println(iObj);   // 언박싱
        System.out.println(dObj);   // 언박싱

        int num1 = iObj.intValue(); // 언박싱
        double num2 = dObj.doubleValue();   //언박싱
        System.out.println(num1);   // 언박싱
        System.out.println(num2);

        iObj = new Integer(iObj.intValue() + 10);   // 래퍼 인스턴스 값 증가
        dObj = new Double(dObj.doubleValue() + 3.14); // 래퍼 인스턴스 값 증가
        System.out.println(iObj);
        System.out.println(dObj);
    }
}
```

자바의 기본 클래스

- 오토 박싱과 오토 언박싱
  - 자바 5부터 박싱과 언박싱을 자동으로 처리
  - 일반적인 변수 사용법으로 사용

```
public class AutoBoxingUnboxing {
    public static void main(String[] args) {
        Integer iObj = 10;
        System.out.println(iObj);
        int num1 = iObj;
        System.out.println(num1);
        iObj++;
        System.out.println(iObj);
        iObj = iObj + 3;
        System.out.println(iObj);
    }
}
```

자바의 기본 클래스

- Number 클래스
  - 래퍼 클래스는 "java.lang.Number" 클래스 상속
  - Number 클래스에 다음의 추상 메소드 존재 : Number 클래스는 추상 클래스
    - public abstract int intValue( )
    - public abstract long longValue( )
    - public abstract double doubleValue( )
  - 래퍼 클래스는 위의 메소드를 모두 구현하고 있으므로 래퍼 인스턴스에 저장된 값을 다른 자료형의 데이터로 변환 가능

```
public class NumberMethod {
    public static void main(String[] args) {
        Integer num1 = 29;
        System.out.println(num1.intValue());
        System.out.println(num1.doubleValue());
        Double num2 = 3.14;
        System.out.println(num2.intValue());
        System.out.println(num2.doubleValue());
    }
}
```

자바의 기본 클래스

- 래퍼 클래스의 static 메소드들

```
public class WrapperClassMethod {
    public static void main(String[] args) {
        // 클래스 메소드를 통한 인스턴스 생성 방법 두 가지
        Integer num1 = Integer.valueOf(5);  // 숫자 기반 인스턴스 생성
        Integer num2 = Integer.valueOf("1024"); // 문자열 기반 인스턴스 생성

        // 대소 비교과 합을 계산하는 클래스 메소드
        System.out.println("큰 수 : " + Integer.max(num1, num2));
        System.out.println("작은 수 : " + Integer.min(num1, num2));
        System.out.println("합 : " + Integer.sum(num1, num2));
    }
}
```

- Math 클래스
  - Math 클래스에 정의된 메소드는 모두 static으로 선언되어 있어 기능의 제공이 목적 -> 인스턴스 생성 목적 아님
  - Math 클래스에 정의된 메소드 예시 : Math.PI, Math.sqrt(n), Math.log(n) 등
  - Math 클래스에 정의된 메소드는 모두 70여개가 넘어 모두 소개할 수 없으므로 필요 시 자바 문서 참조

자바의 기본 클래스

- 씨드(Seed) 기반의 난수 생성
  - 보통 난수(예측 불가능한 수) 생성 시 System.currentTimeMillis()를 씨드로 전달하여 난수 생성
  - System.currentTimeMillis() : 1970년 1월 1일 자정 이후로 지나온 시간을 밀리초 단위로 계산하여 반환
  - 씨드값이 계속 변하기 때문에 항상 다른 패턴의 난수 발생

```
public class SeedRandom {
    public static void main(String[] args) {
        Random rand = new Random(System.currentTimeMillis());
        for (int i = 0; i < 7; i++) {
            System.out.println(rand.nextInt(1000));
        }
    }
}
```