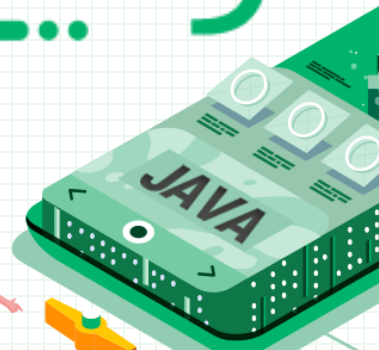




안드로이드 프로그래밍을 위한 자바기초 _...



⑧ 자바 기본 라이브러리 활용하기



학습목표

- Object 및 Wrapper 클래스를 활용하여 프로그래밍을 할 수 있다.
- 기초적인 클래스들을 활용하여 프로그래밍을 할 수 있다.



학습내용

- Object 및 Wrapper 클래스 활용하기
- 기초 클래스 활용하기

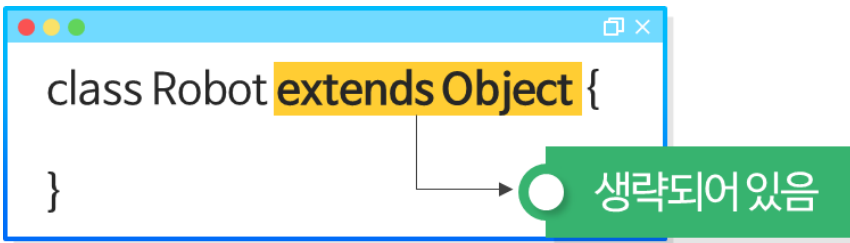
⑧ 자바 기본 라이브러리 활용하기

Object 및 Wrapper 클래스 활용하기

➤ Object 클래스 이해하기

1) Object 클래스란?

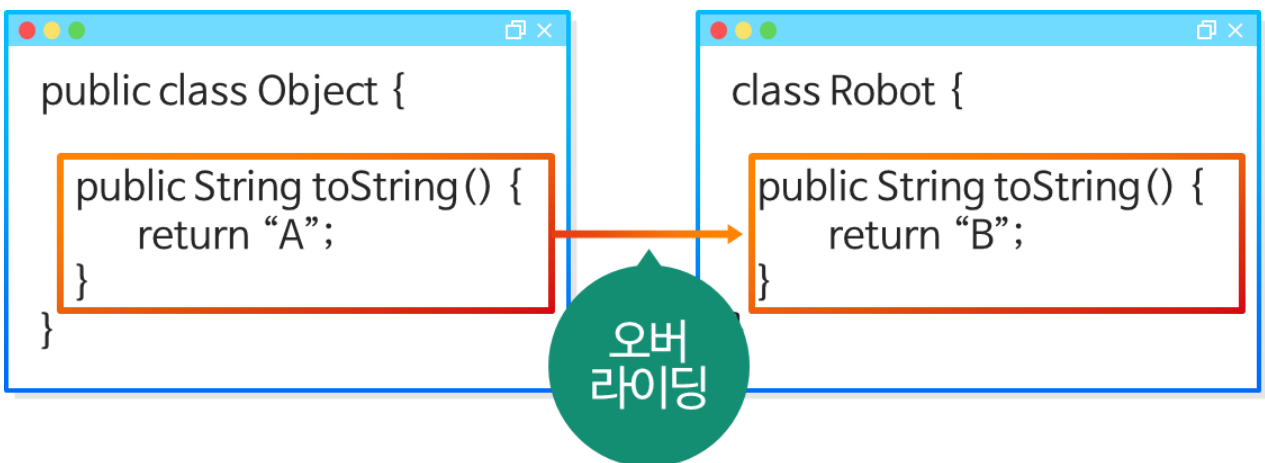
- 모든 자바 클래스의 최상위 부모 클래스
- 패키지 : java.lang
- 자바 컴파일러가 자동으로 상속처리 함
- 자바의 모든 클래스들은 Object 클래스와 동일한 메소드들을 가짐



2) Object 클래스의 주요 메소드

(1) toString()메소드

- ✓ 객체에 대한 정보를 문자열로 반환
- ✓ 오버라이딩하여 원하는 객체 정보로 수정하여 활용함



⑧ 자바 기본 라이브러리 활용하기

Object 및 Wrapper 클래스 활용하기

(2) equals () 메소드

- ✓ 두 객체를 비교하여 동일한지 판단
 - 동일하면 true, 동일하지 않으면 false 반환
- ✓ 두 객체가 참조하는 객체가 동일한지 판단

```

Robot a = new Robot();
Robot b = new Robot();

if( a.equals( b ) ) {
    // 동일하면 true
} else {
    // 동일하지 않으면 false
}

```

객체 a와 b가
동일한지 판단

- ✓ 오버라이딩 하여 활용

```

class Robot {
    int x, y;
    public boolean equals(Object obj) {
        Robot r = (Robot)obj;
        if( r.x == x && r.y == y )
            return true;
        else
            return false;
    }
}

```

```

Robot a = new Robot();
Robot b = new Robot();
a.x = 10;
b.x = 20;
if( a.equals( b ) ) {
    // 동일하면 true
} else {
    // 동일하지 않으면 false
}

```

⑧ 자바 기본 라이브러리 활용하기

Object 및 Wrapper 클래스 활용하기

3) 클래스와 객체 비교 연산자

- '==' 연산자 : 오버라이딩하지 않은 Object 클래스의 equals() 메소드와 동일
- 'instanceof' 연산자 : 객체가 해당 클래스로 생성된 객체인지 판단

```
Robot a = new Robot();  
Robot b = new Robot();
```

```
if( a == b )
```

➡ 객체 a와 b가 동일한지 판단(false)

```
if( a instanceof Robot )
```

➡ 객체 a가 Robot 클래스의
객체인지 판단(true)

```
if( a instanceof Object )
```

➡ 객체 a가 Object 클래스의
객체인지 판단(true)

⑧ 자바 기본 라이브러리 활용하기



실습



Object 클래스 실습



실행 화면

[toString() 메소드 오버라이딩 안함]

objectclass.Robot@54bedef2

objectclass.Robot@5caf905d

[equals() 메소드 오버라이딩 안함]

[a.equals(b)] : false

[toString() 메소드 오버라이딩 함]

[Robot]:[X=10][Y=20][ANGLE=30][JOB=40]

[Robot]:[X=10][Y=20][ANGLE=30][JOB=40]

[equals() 메소드 오버라이딩 함]

[a.equals(b)] : true

[클래스 및 객체 비교] ...

- 소스 파일명 : [ObjectClass.java], [Robot.java]
- 자세한 내용은 실습 영상을 확인해보세요.

⑧ 자바 기본 라이브러리 활용하기

Object 및 Wrapper 클래스 활용하기

▶ Wrapper 클래스 이해하기

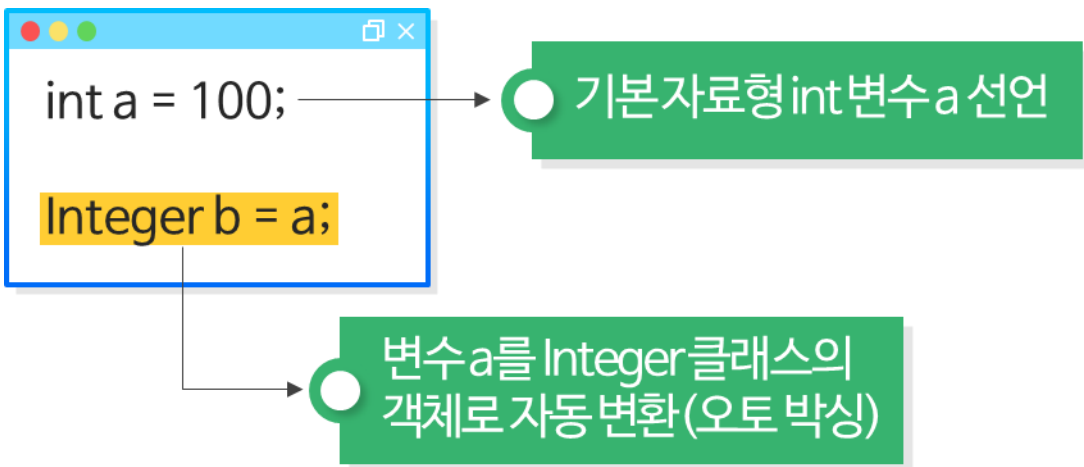
1) Wrapper 클래스란?

- 기본 자료형에 대응하는 클래스
- 패키지 : java.lang
- 클래스의 다양한 메소드 활용 가능

기본 자료형	Wrapper 클래스
boolean	Boolean
char	Character
byte	Byte
short	Short
int	Integer
long	Long
float	Float
double	Double

2) 오토 박싱과 오토 언박싱

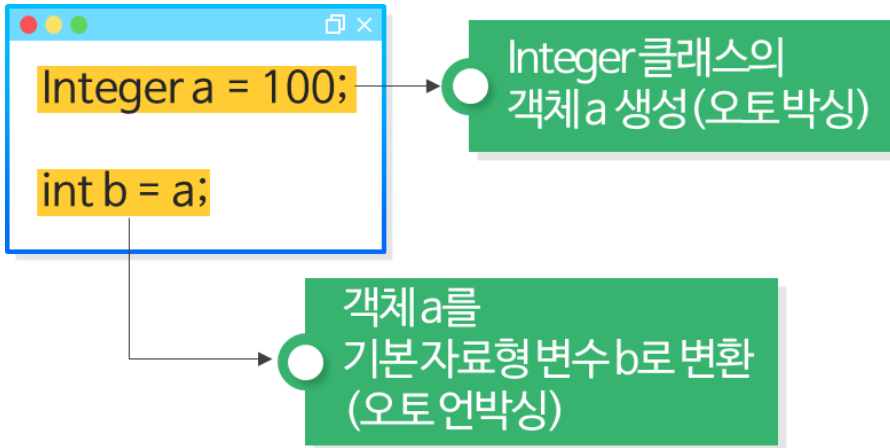
- 오토 박싱(auto boxing)
 - ✓ 기본 자료형 값을 Wrapper 클래스의 객체로 자동 변환해 주는 것



⑧ 자바 기본 라이브러리 활용하기

Object 및 Wrapper 클래스 활용하기

- 오토 언박싱(auto unboxing)
 - ✓ Wrapper 클래스의 객체에 있는 값을 기본 자료형으로 자동 변환해 주는 것



3) Wrapper 클래스 구성

멤버	유형	예	설명
상수	클래스 특성에 맞는 상수들 존재		
대표적인 멤버 메소드	xxxValue()	intValue() floatValue() ...	기본 자료형의 값으로 반환
	isXXX()	isFinite() isNan() ...	값을 검사
	toXXXString()	toString() toHexString() ...	값을 문자열로 표현
	valueOf(xxx)	valueOf(int) valueOf(float) ...	기본 자료형을 Wrapper 클래스의 객체로 변환

⑧ 자바 기본 라이브러리 활용하기

Object 및 Wrapper 클래스 활용하기

4) Wrapper 클래스 사용하기

(1) integer 클래스

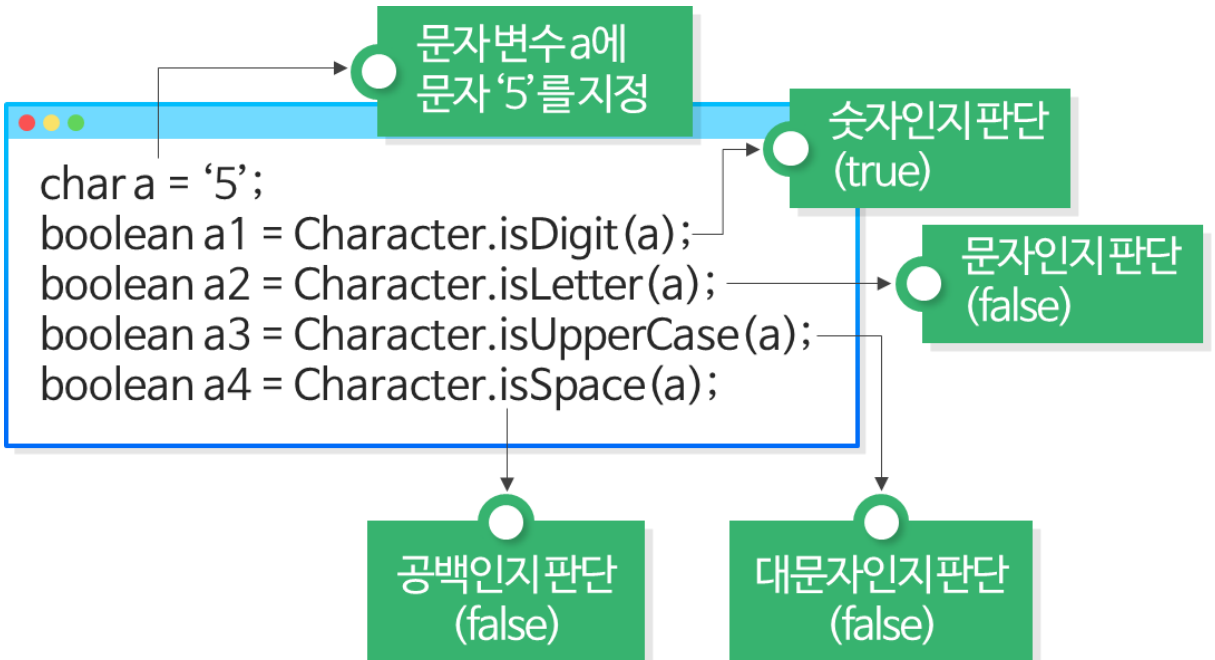
- ✓ 문자열 숫자 ("123")를 숫자(123)로 변환하는 경우

```
int a = Integer.parseInt("123");
```

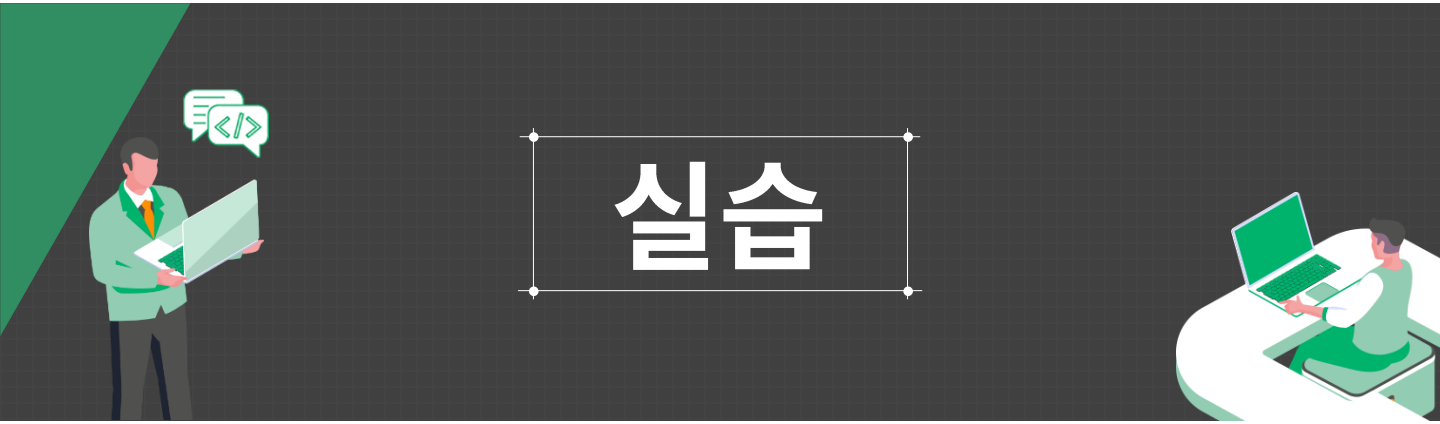
- ✓ 숫자(123)을 문자열 숫자("123")로 변환하는 경우

```
String a = Integer.toString(123);
```

(2) Character 클래스



⑧ 자바 기본 라이브러리 활용하기



Wrapper 클래스 실습



실행 화면

```
Boolean : true  
Byte : 100  
Character : a  
Short : 200  
Integer : 300  
Long : 400  
Float : 10.5  
Double : 10.5  
Integer --> float : 300.0  
Binary : 100101100  
Octal : 454  
Hexadecimal : 12c  
a=Unicode...
```

- 소스 파일명 : [WrapperClass.java]
- 자세한 내용은 실습 영상을 확인해보세요.

⑧ 자바 기본 라이브러리 활용하기

기초 클래스 활용하기

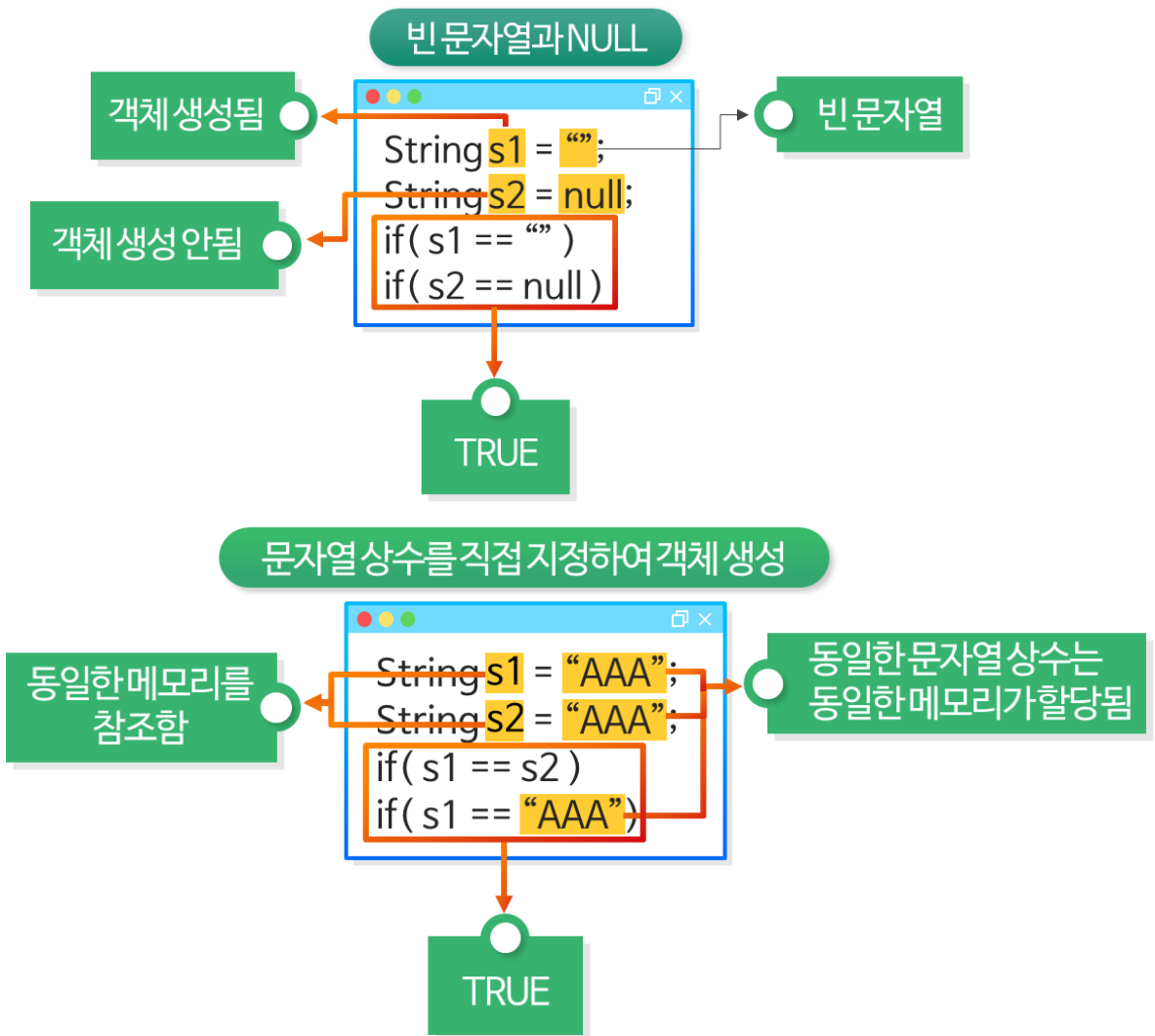
▶ String 클래스 이해하기

1) String 클래스란?

- 문자열 처리를 위한 클래스
- 한번 지정된 문자열은 변경이 불가능함

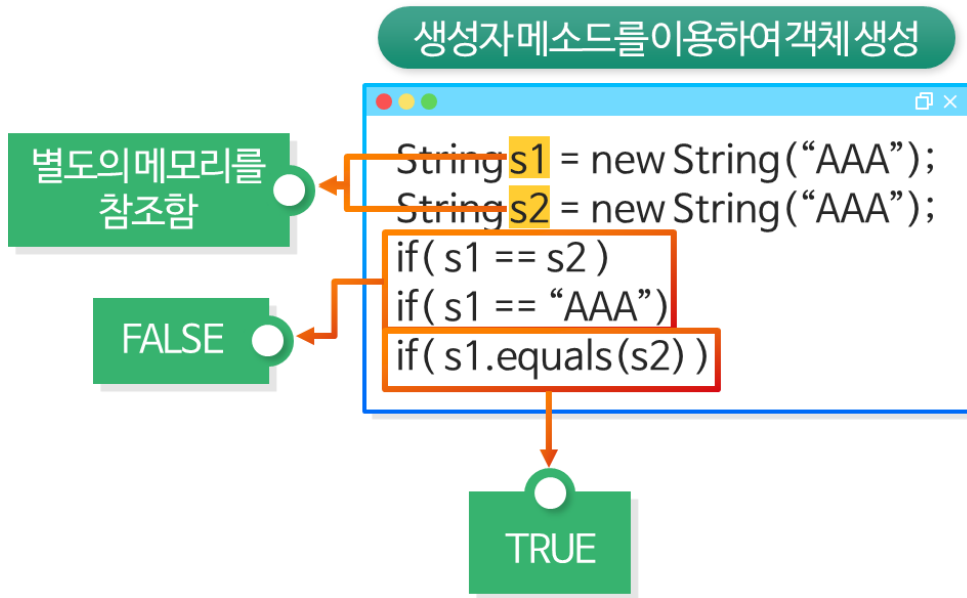
2) String 클래스 객체 생성

- 문자열 상수를 직접 지정하는 방법
- String 클래스의 생성자 메소드를 이용하는 방법



⑧ 자바 기본 라이브러리 활용하기

기초 클래스 활용하기



3) String 클래스의 주요 메소드

주요 메소드	설명
int length()	문자열 길이
boolean equals(Object obj)	객체의 내용을 비교
boolean equalsIgnoreCase(String)	대소문자 구분없이 비교
String subString(int index)	index부터 마지막까지 문자열 반환
String concat(String str)	문자열을 연결하여 문자열 반환
String replace(char old, char new)	문자를 교체한 문자열 반환
String toLowerCase()	소문자로 변경한 문자열 반환
String toUpperCase()	대문자로 변경한 문자열 반환
char charAt(int index)	index 위치의 문자를 반환
int indexOf(String str)	처음부터 str 문자열의 위치를 반환
int lastIndexOf(String str)	뒤에서부터 str 문자열 위치 반환
String trim()	문자열의 시작과 끝 공백을 삭제한 문자열 반환

⑧ 자바 기본 라이브러리 활용하기

기초 클래스 활용하기

➤ StringBuffer 클래스 이해하기

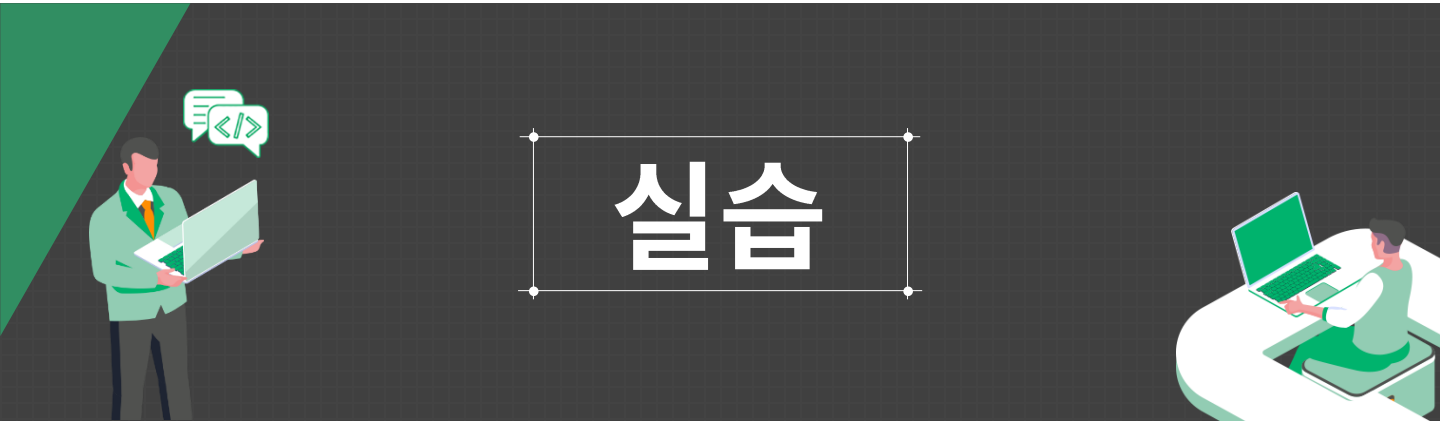
1) StringBuffer 클래스란?

- 버퍼 메모리를 확보해 놓고 문자열을 처리
- String 클래스보다 처리 속도가 빠름
- 저장된 문자열을 수정 및 변경할 수 있어서 다양한 문자열 처리가 가능함

2) StringBuffer 클래스의 주요 메소드

주요 메소드	설명
int length()	문자열 개수 반환
int capacity()	현재 할당된 버퍼의 크기 반환
StringBuffer append(String str)	str 문자열을 뒤에 붙여서 추가
StringBuffer insert(int offset, String str)	str을 offset 위치에 끼워넣기
StringBuffer delete(int start, int end)	start 부터 end-1 사이의 문자열 삭제
StringBuffer replace(int s, int e, String c)	s에서 e-1 위치의 문자열 삭제 후 c 삽입함
StringBuffer reverse()	문자열의 순서를 뒤집음

⑧ 자바 기본 라이브러리 활용하기



문자열 관련 클래스 실습



실행 화면

```
== String 클래스 ==  
[빈 문자열과 NULL]  
빈문자열 비교 : true  
NULL 문자 비교 : true  
[문자열 상수를 직접지정하여 객체 생성]  
s4 == s5 : true  
s4 == 'AAA' : true  
[생성자 메소드를 이용하여 객체 생성]  
s6 == s7 : false  
s6 == 'AAA' : false  
s6.equals(s7) : true  
s6.equals('AAA') : true  
[주요 메소드 사용] ...
```

- 소스 파일명 : [StringClass.java]
- 자세한 내용은 실습 영상을 확인해보세요.

⑧ 자바 기본 라이브러리 활용하기

기초 클래스 활용하기

➤ Random 클래스 이해하기

1) Random 클래스란?

- 임의의 값을 발생시키는 메소드를 가진 클래스
- 객체 생성 후 여러 번 임의의 값을 가져올 수 있음
- 패키지 : java.util

2) Random 클래스의 주요 메소드

주요 메소드	설명
int nextInt()	정수형 임의의 값을 반환
int nextInt(int n)	0에서 n 사이의 임의의 값 반환 (n은 포함하지 않음)
float nextFloat() double nextDouble()	0.0에서 1.0사이의 임의의 값 반환 (1.0은 포함하지 않음)

➤ Math 클래스 이해하기

1) Math 클래스란?

- 수학과 관련된 메소드를 제공하는 클래스
- 대부분 정적 멤버 메소드로 정의되어 있음

2) Math 클래스의 주요 메소드

주요 메소드	설명
static int abs(int a)	절대값 반환(자료형 별로 오버로딩)
static int max(int a, int b)	큰값을 반환(자료형 별로 오버로딩)
static int min(int a, int b)	작은값을 반환(자료형 별로 오버로딩)
static double log(double a)	로그 함수값 반환
static long round(double a)	반올림 값을 반환
static double sqrt(double a)	\sqrt{a} 값을 반환
static double pow(double a, double b)	a의 b승을 반환
static double sin(double a)	사인값을 반환

⑧ 자바 기본 라이브러리 활용하기

기초 클래스 활용하기

➤ Calendar클래스 이해하기

1) Calendar 클래스란?

- (1) 날짜와 시간을 표현하는 클래스
- (2) 시스템의 날짜와 시간을 가져오기 위해서 getInstance()메소드로 객체를 생성

```
Calendar now = Calendar.getInstance();
```

(3) 패키지 : java.util

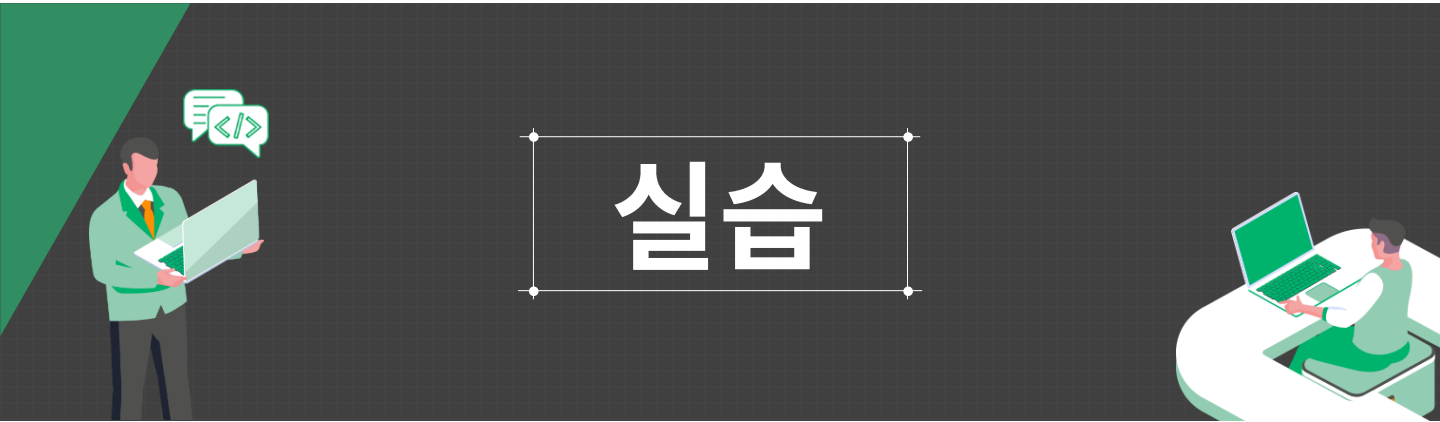
2) Calendar 클래스의 주요 메소드

주요 메소드	설명
int get(int field)	field 값에 따라 해당 값을 반환

3) Calendar 클래스의 주요 상수

주요 상수	설명
YEAR	년
MONTH	월
DATE	일
DAY_OF_WEEK	요일 SUNDAY=1, MONDAY=2, TUESDAY=3, WEDNESDAY=4, THURSDAY=5, FRIDAY=6, SATURDAY=7
HOUR	시 - 12시간 기준
HOUR_OF_DAY	시 - 24시간 기준
MINUTE	분
SECOND	초
AM_PM	오전(Calendar.AM), 오후(Calendar.PM)

⑧ 자바 기본 라이브러리 활용하기



Random, Math, Calendar 클래스 실습



실행 화면

```
[Random 클래스]
nextInt(100) : 22
nextInt(100) : 85
nextInt(100) : 48
nextInt(100) : 16
nextInt(100) : 64
nextDouble() : 0.7634806285186301
nextDouble() : 0.8620516262754176
nextDouble() : 0.060467169004568055
nextDouble() : 0.18454061677531397
nextDouble() : 0.8794152685875463
[Math 클래스]
abs(-20):20...
```

- 소스 파일명 : [RMCCClass.java]
- 자세한 내용은 실습 영상을 확인해보세요.



정리하기

■ Object 및 Wrapper 클래스 활용하기

- Object 클래스
 - 모든 자바 클래스의 최상위 부모 클래스
 - 패키지 : java.lang
 - 자바 컴파일러가 자동으로 상속처리함
 - 자바의 모든 클래스들은 Object 클래스와 동일한 메소드들을 가짐
- Object 클래스의 주요 메소드
 - toString() 메소드 : 객체에 대한 정보를 문자열로 반환
 - equals() 메소드 : 두 객체를 비교하여 동일한지 판단함
- Wrapper 클래스
 - 기본 자료형에 대응하는 클래스
 - 패키지 : java.lang
 - 클래스의 다양한 메소드 활용 가능
- 오토 박싱(auto boxing)
 - 기본 자료형 값을 Wrapper 클래스의 객체로 자동 변환해 주는 것
- 오토 언박싱(auto unboxing)
 - Wrapper 클래스의 객체에 있는 값을 기본 자료형으로 자동 변환해 주는 것



정리하기

■ 기초 클래스 활용하기

- String 클래스
 - 문자열 처리를 위한 클래스
 - 문자열은 변경이 불가능 함
 - String 클래스 객체 생성
 - 문자열 상수를 직접 지정하는 방법
 - String 클래스의 생성자 메소드를 이용하는 방법
- StringBuffer 클래스
 - 버퍼 메모리를 확보해 놓고 문자열을 처리
 - String 클래스 보다 처리 속도가 빠름
 - 문자열을 수정 및 변경할 수 있어서 다양한 문자열 처리가 가능함
- Random 클래스
 - 임의의 값을 발생시키는 클래스
 - 객체 생성 후 여러 번 임의의 값을 가져올 수 있음
 - 패키지 : java.util
- Math 클래스
 - 수학과 관련된 메소드를 제공하는 클래스
 - 대부분 정적 멤버 메소드로 정의되어 있음
- Calendar 클래스
 - 날짜와 시간을 표현하는 클래스
 - 시스템의 날짜와 시간을 가져오기 위해서 getInstance() 메소드로 객체를 생성
 - 패키지 : java.util



정리하기

■ 기초 클래스 활용하기

- 배열과 반복문 사용하기
 - 배열과 반복문의 관계
 - 배열의 인덱스값은 0부터 순차적으로 증가함
 - 반복문을 이용하면 인덱스값을 순차적으로 증가할 수 있음
 - 배열이름 .length : 배열 요소의 개수를 가짐
- 문자열 배열 사용하기
 - 문자열 배열 선언 및 생성하기: `String[] a = new String[3];`
 - 문자열 배열 초기화: `String[] a = new String[] { "apple", "banana", "melon" };`
- main()메소드 인수 처리
 - 실행시 main() 메소드의 인수로 값이 넘어옴
 - 인수 지정하는 방법
 - [명령 프롬프트]에서 실행 시 지정하는 방법
 - [이클립스]에서 지정하는 방법



정리하기

■ 다차원 배열 이해하기

- 다차원 배열이란?
 - 2차원 배열 이상
 - 종류
 - 2차원 배열 : 행 N개, 열 N개 (1차원 배열이 N개)
 - 3차원 배열 : 총 N개, 행 N개, 열 N개(2차원 배열이 N개)
 - 2차원 배열 : 2차원 배열 선언 및 생성 후 배열 요소에 초기값 지정
 - 2차원 가변 배열 : 행의 개수는 정해져 있고, 열의 개수가 가변임
 - 2차원 배열의 length 멤버 변수(필드)
 - 배열이름.length : 2차원 배열의 행의 개수
 - 배열이름[행].length : 2차원 배열의 열의 개수