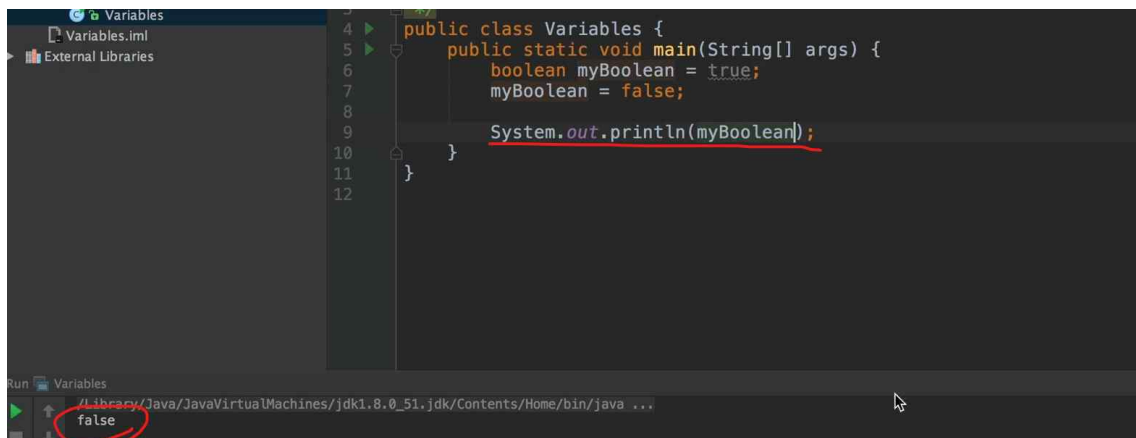


boolean

(아래 이미지로 된 코드 내용들은 모두
필수적으로 따라서 사용하시고 결과값을 ‘꼭’
확인하세요!)

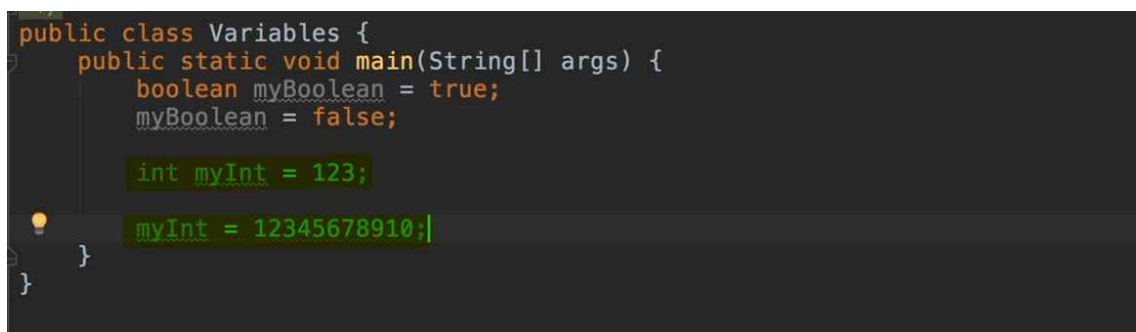
자, 이번에는 boolean에 대해서 배워볼겁니다.
불린은 참과 거짓을 나타내는 중요한 자료형입니다.
이런식으로 트루와 펄스를 넣을 수 있습니다.



```
public class Variables {  
    public static void main(String[] args) {  
        boolean myBoolean = true;  
        myBoolean = false;  
        System.out.println(myBoolean);  
    }  
}
```

Run Variables
false

출력을 해보면 우리가 입력한 값이 잘 나오는 걸 확인 할 수 있습니다.



```
public class Variables {  
    public static void main(String[] args) {  
        boolean myBoolean = true;  
        myBoolean = false;  
        System.out.println(myBoolean);  
        int myInt = 123;  
        myInt = 12345678910;  
    }  
}
```

다음으로는 우리가 이전에 배워봤던 int형입니다.

int형은 이렇게 큰 값을 넣어주게 되면

```
public class Variables {
    public static void main(String[] args) {
        boolean myBoolean = true;
        myBoolean = false;

        int myInt = 123;

        myInt = 12345678910;
    }
}
```

Information: Using javac 1.8.0_51 to compile java sources
Information: java: Errors occurred while compiling module 'Variables'
Information: 10/27/16, 12:38 AM - Compilation completed with 1 error and 0 warnings in 410ms
Error: (11, 17) java: integer number too large: 12345678910

실행을 해보면, 역시 오류가 나게 되네요.

정수형의 범위는 -2의 31승, 약 -21억부터 2의 31승 빼기 1, 약 21억까지입니다.

이렇게 더 큰 값을 저장 할 때에는,

```
public class Variables {
    public static void main(String[] args) {
        boolean myBoolean = true;
        myBoolean = false;

        int myInt = 123;

        long myLong = 12345678910;
    }
}
```

이렇게 롱이라는 자료형을 사용해주어야 합니다.

하지만 여기서 우리가 넣은 값 자체가 사실 int형이라서 여전히 오류가 나게 되는데요,

이럴 때는 어떻게 해야 할까요?

이렇게 뒤에 L을 붙여주면 롱 형식으로 이해하고 사용할 수 있게 됩니다.

```

    }
    long myLong = 12345678910L;
}

```

그리고 숫자를 더하는 방식에 대해서도 알아보게요.

```

public class Operators {
    public static void main(String[] args) {
        System.out.println("숫자 연산자");
        int a = 9;
        int b = 5;
        System.out.println(a + " + " + b + " = " + (a + b));
        System.out.println(a + " - " + b + " = " + (a - b));
        System.out.println(a + " * " + b + " = " + (a * b));
        System.out.println(a + " / " + b + " = " + (a / b));
        System.out.println(a + " % " + b + " = " + (a % b));
    }
}

```

제가 써놓은 코드를 보면, 더하기, 빼기 곱하기 나누기, 나머지 연산자가 있습니다.

이렇게 실행을 해보면, 모두 잘 나오는 걸 확인할 수 있습니다.

그런데 보시면

나눗셈 결과값이 1이네요 java에서 정수형과 정수형 나누기의 결과값은 소수 형식이 아니라 무조건 정수형입니다. 이 경우에 9 나누기 5는 1.8인데 소수 부분이 버려져서 1이 나오게 된 거죠.

그렇다면 int를 더블로 바꿔봅시다.

```

public class Operators {
    public static void main(String[] args) {
        System.out.println("숫자 연산자");
        int a = 9;
        double b = 5;
        System.out.println(a + " + " + b + " = " + (a + b));
        System.out.println(a + " - " + b + " = " + (a - b));
        System.out.println(a + " * " + b + " = " + (a * b));
        System.out.println(a + " / " + b + " = " + (a / b));
        System.out.println(a + " % " + b + " = " + (a % b));
    }
}

```

이러고 실행을 해주면,

```
run Operators
/Library/Java/JavaVirtualMachines/jdk1.8.0_51.jdk/Contents/Home/bin/java ...
숫자 연산자
9 + 5.0 = 14.0
9 - 5.0 = 4.0
9 * 5.0 = 45.0
9 / 5.0 = 1.8
9 % 5.0 = 4.0
Process finished with exit code 0
```

결과값이 모두 소수형이 나오고, 잘못 된 나누기 연산도 잘 나온 걸 확인할 수 있습니다.

지금 보셨다시피, 더해지는 과정에서 하나라도 소수형이 있다면 결과값은 소수형으로 나오게 됩니다.

설명을 좀 쉽게 드리자면, 정수형보다는 소수형이 힘이 더 셌다고 볼 수 있는 거죠.

그럼 이번에는 불린 연산자를 봐볼게요.

```
public class Operators {
    public static void main(String[] args) {
        System.out.println("비교 연산자");
        System.out.println(3 + " > " + 2 + " : " + (3 > 2));
        System.out.println(3 + " < " + 2 + " : " + (3 < 2));
        System.out.println(3 + " >= " + 2 + " : " + (3 >= 2));
        System.out.println(3 + " <= " + 2 + " : " + (3 <= 2));
        System.out.println(3 + " == " + 2 + " : " + (3 == 2));
        System.out.println(3 + " != " + 2 + " : " + (3 != 2));
    }
}
```

보면 3은 2보다 크기 때문에 true가 나왔고, 다들 정상적으로 나오는 모습을 볼 수 있습니다.

자, 아래 이미지도 따라서 써보세요.



```

public class Operators {
    public static void main(String[] args) {
        System.out.println("불린 연산자 - and");
        System.out.println("true && true : " + (true && true));
        System.out.println("true && false : " + (true && false));
        System.out.println("false && true : " + (false && true));
        System.out.println("false && false : " + (false && false));

        System.out.println("불린 연산자 - or");
        System.out.println("true || true : " + (true || true));
        System.out.println("true || false : " + (true || false));
        System.out.println("false || true : " + (false || true));
        System.out.println("false || false : " + (false || false));

        System.out.println("불린 연산자 - not");
        System.out.println("!true : " + !true);
        System.out.println("!false : " + !false);
    }
}

```

자, 일단은 여기서 주목하실 점은 &&와 ||인데요, 이 엔드는 그리고라는 의미를 가지고 있고, OR은 둘 중 하나라도 라는 의미가 있습니다. 엔드는 양쪽 값이 모두 참이어야하고, OR은 둘 중 하나라도 참이라면 True로 인식하게 되는 거죠.

1. 형변환과 타입캐스팅

제가 저번에 말씀드렸었죠? 정수형과 실수형을 더하게 되면, 실수형이 더 세기 때문에 실수형으로 변환이 된다고요.

사실 이게 별 거 아닐지 몰라도, 현업에서 굉장히 많이 사용하는 것 중 하나입니다.



숫자의 랭킹은 이렇다고 보시면 되세요. 그러면 더블이 제일 강하고, byte가 제일 약하겠네요. 그럼 어떤 기준으로 이 순위를 정한 걸까요. 바로, 각 자료형이 가질 수 있는 값의 범위입니다. 자, 예를 들어볼게요

아래로

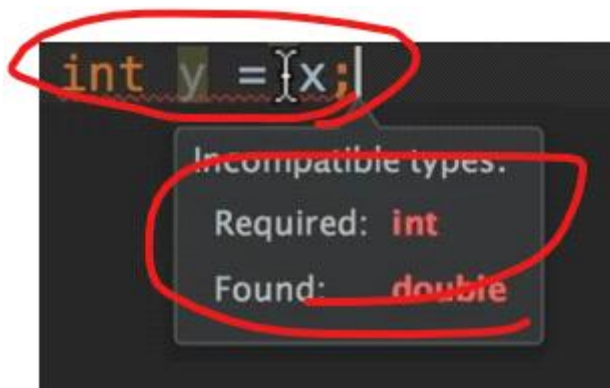
```

public class TypeConversionCasting {
    public static void main(String[] args) {
        double x = 3.14;
        int y = x;
    }
}

```

이런식으로 더블에 들어있는 값을 int형 y에 집어넣습니다.

그런데, 제가 마우스를 올려보면,



오류가 나는 걸 확인할 수 있습니다.

더블은 인트보다 범위가 더 크기 때문에, 인트에 더블의 값을 넣어주려고 하면, 정보를 잃을 수도 있습니다. 그래서 애초에 자바가 허락하지 않는 겁니다.

반대로,

```

public class TypeConversionCasting {
    public static void main(String[] args) {
        int x = 3;
        double y = x;
    }
}

```

이런 식으로 바꿔해보겠습니다.

이렇게 하면 오류가 나지 않게 되죠. 이것을 형변환이라고 합니다.

```

public class TypeConversionCasting {
    public static void main(String[] args) {
        double x = 3.14;
        int y = x;
    }
}

```

다시 바꿔주구요.

그런데 역지로나도 더블을 인트로 바꾸고 싶을 수도 있습니다.

이럴 땐,

```


public class TypeConversionCasting {
    public static void main(String[] args) {
        double x = 3.14;
        int y = (int) x;
    }
}

```

타입 캐스팅을 해주면 됩니다.

이렇게 되면 x에 소수 부분인 0.14가 버려지고, 정수 y에 3이 들어가게 됩니다.

실행을 해보면, 잘 나오는 걸 확인할 수 있습니다.



```

public class TypeConversionCasting {
    public static void main(String[] args) {
        double x = 3.14;
        int y = (int) x;
        System.out.println(y);
    }
}

```

Run TypeConversionCasting
Library/Java/JavaVirtualMachines/jdk1.8.0_51.jdk/Contents/Home/bin/java ...
3

이 경우엔 1이 나오는데, 1.8이 나오길 바라면 예전처럼 하나를 더블로 바꿔줘도 되겠지만,


```

public class TypeConversionCasting {
    public static void main(String[] args) {
        double x = 3.14;
        int y = (int) x;

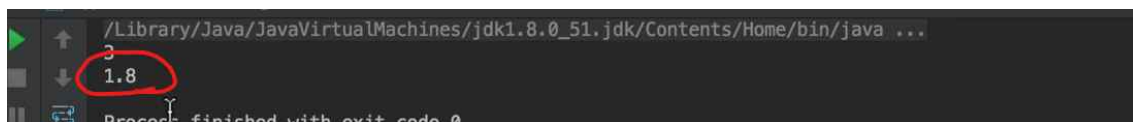
        System.out.println(y);

        int a = 9;
        int b = 5;
        System.out.println(a / b);
    }
}

int a = 9;
int b = 5;
System.out.println((double) a / b);

```

이렇게 해주는 방법도 있습니다.

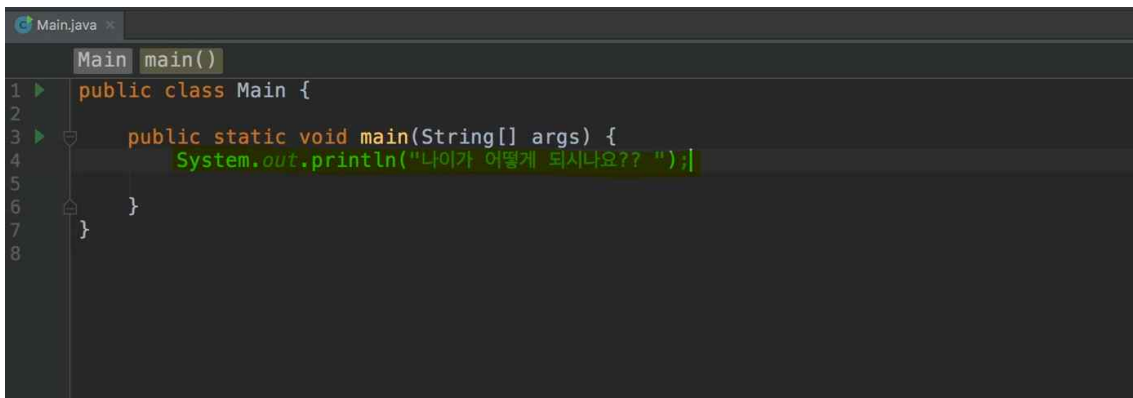


그럼 이렇게 잘 나오게 됩니다.

2. Scanner

java의 스캐너라는 것을 사용 해볼 겁니다.

문법적으로 이해하기 위해서는 아직 배워야 할 것들이 많지만, 사용 자체는 어렵지 않으니 한번 해보도록 하겠습니다.

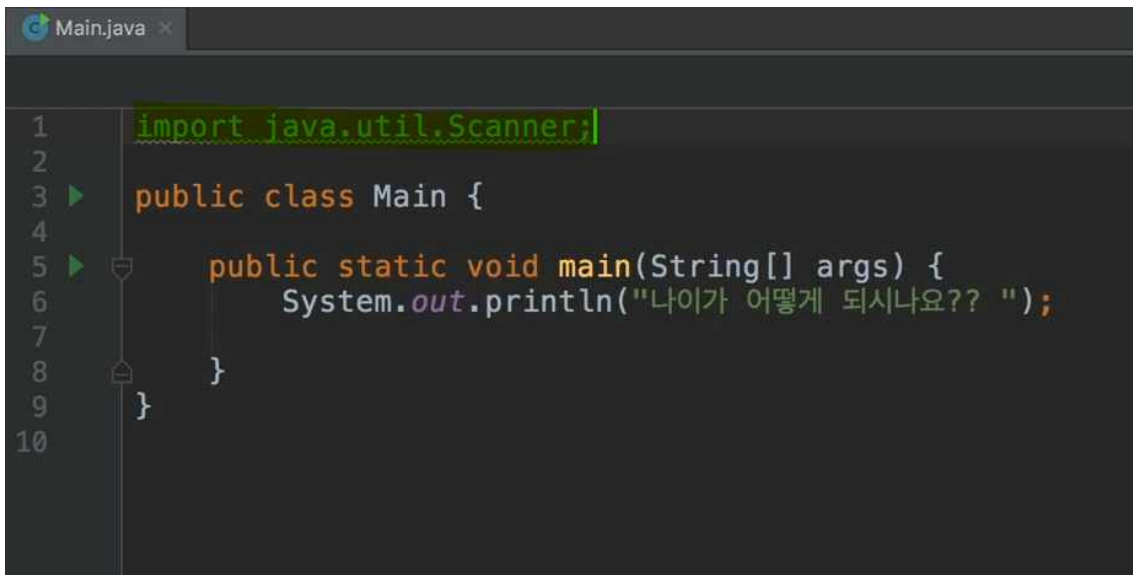


```
1 ▶ public class Main {
2
3 ▶   public static void main(String[] args) {
4     System.out.println("나이가 어떻게 되시나요?? ");
5
6   }
7 }
8
```

위 보시면 뭔가를 묻는 장면이 나오게 되죠.

이제 실제로 입력을 받아볼건데요.

그러려면 먼저,



```
1 import java.util.Scanner;
2
3 ▶ public class Main {
4
5 ▶   public static void main(String[] args) {
6     System.out.println("나이가 어떻게 되시나요?? ");
7
8   }
9 }
10
```

이렇게 스캐너의 자료를 임포트 해줘야 합니다.

아래로

이렇게 되면 우리 이제 스캐너를 사용할 준비가 된 거구요.

스캐너를 사용하려면,

```
import java.util.Scanner;

public class Main {

    public static void main(String[] args) {
        System.out.println("나이가 어떻게 되시나요?? ");
        Scanner scanner;
    }
}
```

반수정

이렇게 해주면 되는데,

사실 이렇게 적어주는 과정이 조금 너무 번거롭죠.

그래서 현재 우리가 사용하는 IDE의 도움을 받아서 만들어보겠습니다.

```
Main.java
Main main()
1 ▶ public class Main {
2
3 ▶   public static void main(String[] args) {
4     System.out.println("나이가 어떻게 되시나요?? ");
5
6
7
8   }
9
10 }
```

이렇게 다시 원상태로 돌려주고.

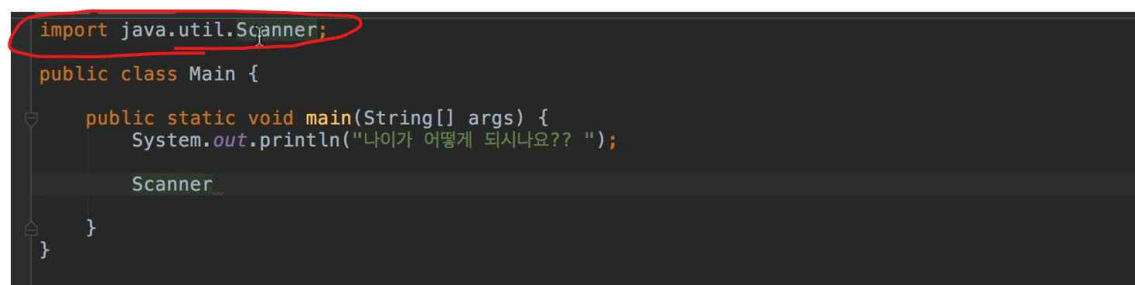
```
Main.java
Main main()
1 ▶ public class Main {
2
3 ▶   public static void main(String[] args) {
4     System.out.println("나이가 어떻게 되시나요?? ");
5
6     Sca
7
8     ScatteringByteChannel (java.nio.channels)
9     ScalableIconUIResource (com.sun.java.swing,...
10    Scale (javafx.scene.transform)
    Scale (sun.tools.jstat)
    ScaledBlit (sun.java2d.loops)
    ScaleTransition (javafx.animation)
```

이렇게 적다 보면 자동완성 기능이 나오게 됩니다.



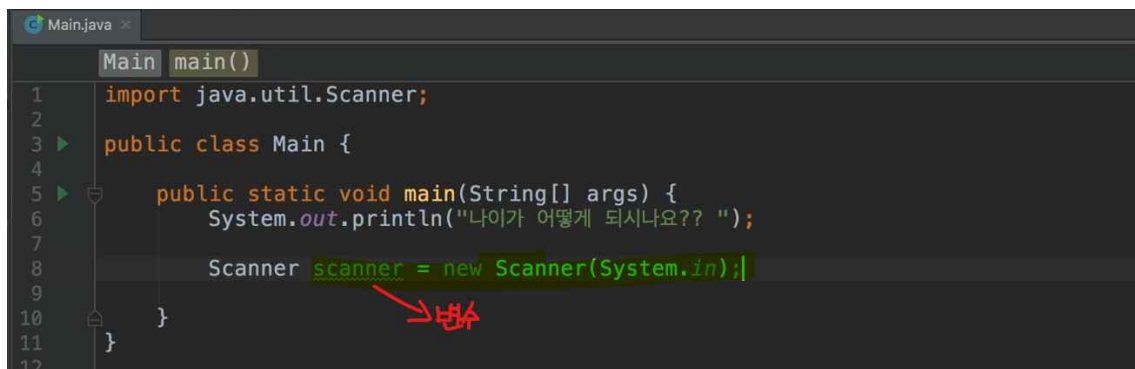
```
1 public class Main {
2
3     public static void main(String[] args) {
4         System.out.println("나이가 어떻게 되시나요?? ");
5
6         Scal
7         Scanner (java.util)
8         ScatteringByteChannel (java.nio.channels)
9         ScalableIconUIResource (com.sun.java.swing...)
10        Scale (javaafx.scene.transform)
11        Scale (sun.tools.jstat)
12        ScalablePlot (com.sun.java.swing...)
```

여기서 스캐너를 선택하고 엔터를 누르게 되면,



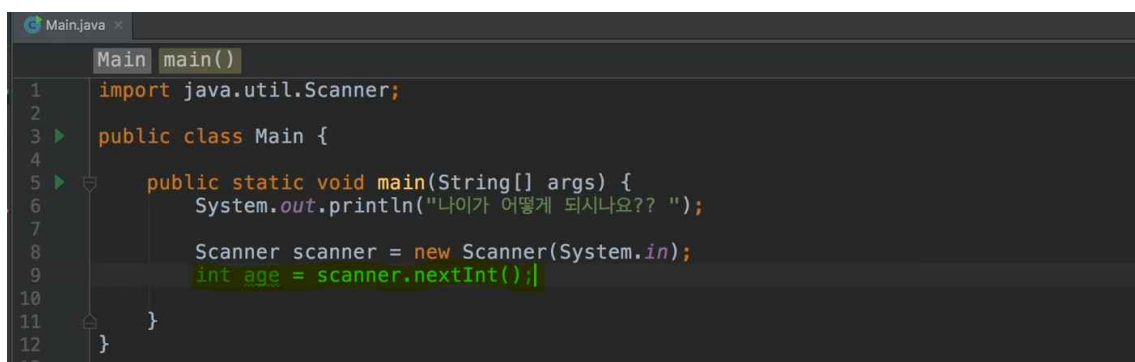
```
import java.util.Scanner;
public class Main {
    public static void main(String[] args) {
        System.out.println("나이가 어떻게 되시나요?? ");
        Scanner
    }
}
```

자동으로 임포트가 됩니다.



```
1 import java.util.Scanner;
2
3 public class Main {
4
5     public static void main(String[] args) {
6         System.out.println("나이가 어떻게 되시나요?? ");
7
8         Scanner scanner = new Scanner(System.in);
9
10    }
11 }
12
```

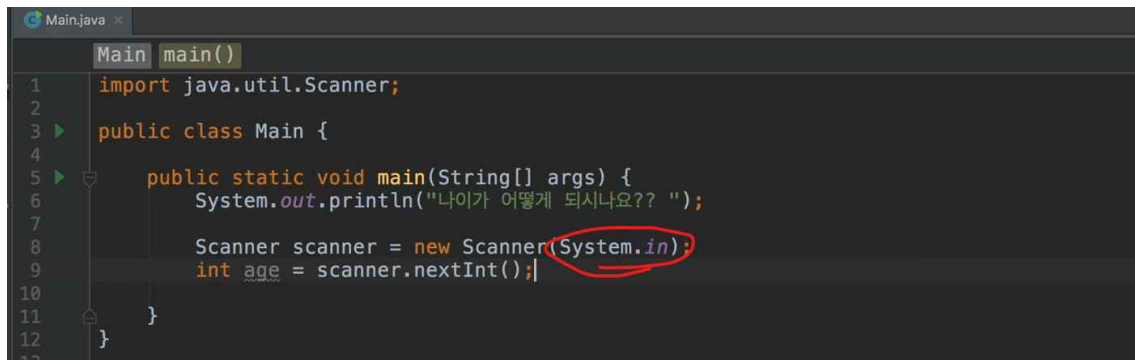
그럼 제가 실제로 사용해볼게요.



```
1 import java.util.Scanner;
2
3 public class Main {
4
5     public static void main(String[] args) {
6         System.out.println("나이가 어떻게 되시나요?? ");
7
8         Scanner scanner = new Scanner(System.in);
9         int age = scanner.nextInt();
10    }
11 }
12
```

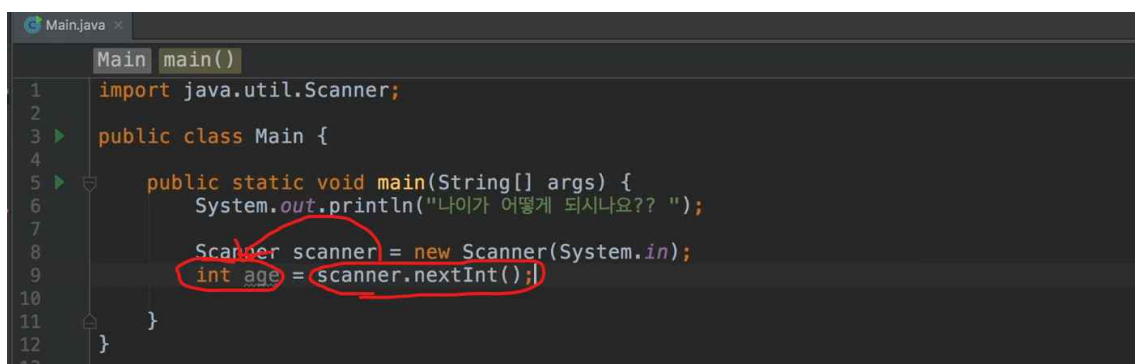
아래로

자, 지금 보시는 것처럼



```
1  import java.util.Scanner;
2
3  public class Main {
4
5      public static void main(String[] args) {
6          System.out.println("나이가 어떻게 되시나요?? ");
7
8          Scanner scanner = new Scanner(System.in);
9          int age = scanner.nextInt();
10
11      }
12  }
```

system.in이라고 적어주면 콘솔에서 입력을 받을 수 있게 되는 거고,

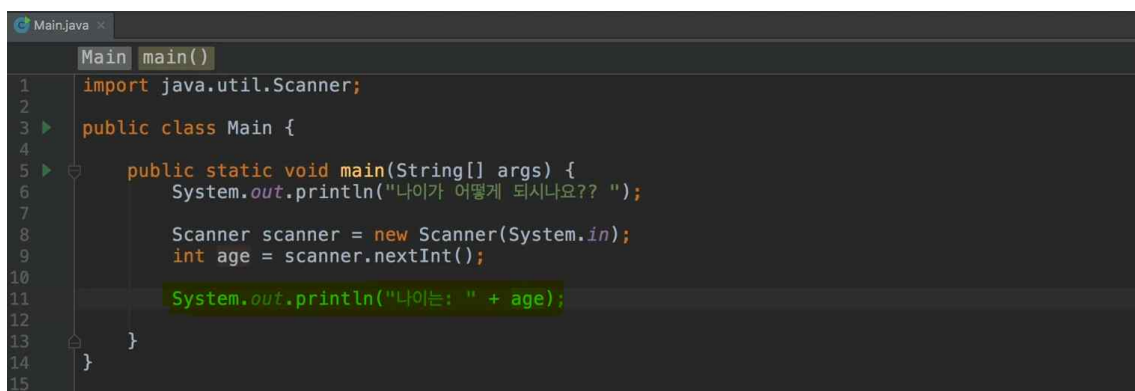


```
1  import java.util.Scanner;
2
3  public class Main {
4
5      public static void main(String[] args) {
6          System.out.println("나이가 어떻게 되시나요?? ");
7
8          Scanner scanner = new Scanner(System.in);
9          int age = scanner.nextInt();
10
11      }
12  }
```

nextInt 라고 해주게 되면 우리의 변수에 정수형을 입력받을 수 있게 됩니다.

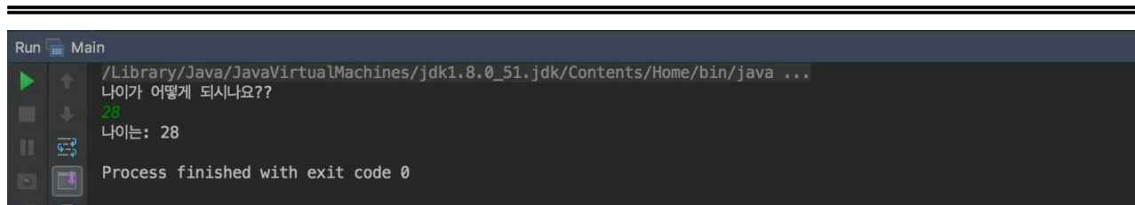
그 정수를 int age에 넣는 거죠.

이제 이 입력받은 숫자를 출력해보겠습니다.



```
1  import java.util.Scanner;
2
3  public class Main {
4
5      public static void main(String[] args) {
6          System.out.println("나이가 어떻게 되시나요?? ");
7
8          Scanner scanner = new Scanner(System.in);
9          int age = scanner.nextInt();
10
11          System.out.println("나이는: " + age);
12
13      }
14  }
```

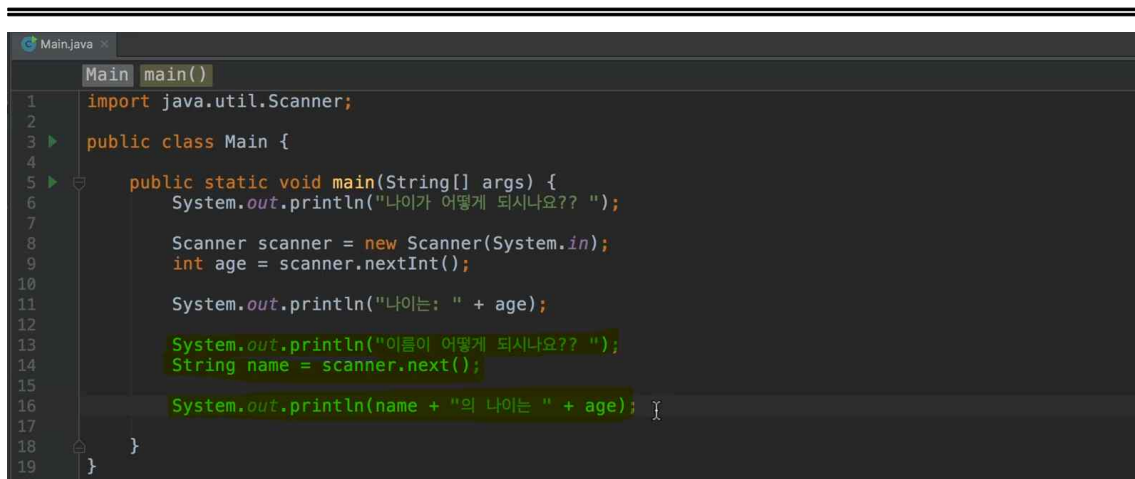
아래로
실행을 해주게 되면.



```
Run Main
/Library/Java/JavaVirtualMachines/jdk1.8.0_51.jdk/Contents/Home/bin/java ...
나이가 어떻게 되시나요??
28
나이는: 28
Process finished with exit code 0
```

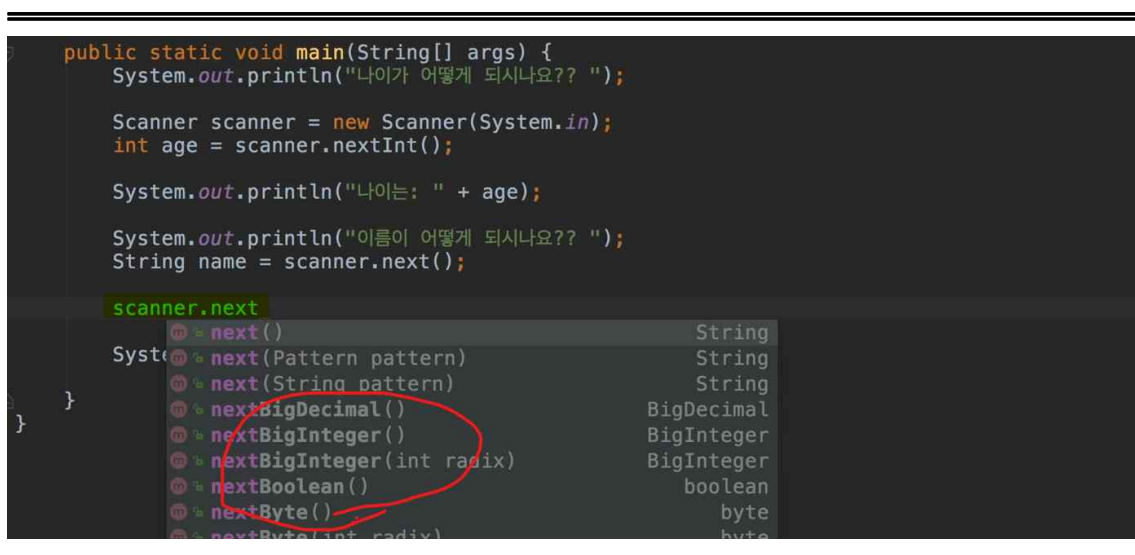
잘 입력이 받아지죠.

자, 이름도 받아볼게요.



```
Main.java
Main main()
1 import java.util.Scanner;
2
3 public class Main {
4
5     public static void main(String[] args) {
6         System.out.println("나이가 어떻게 되시나요?? ");
7
8         Scanner scanner = new Scanner(System.in);
9         int age = scanner.nextInt();
10
11         System.out.println("나이는: " + age);
12
13         System.out.println("이름이 어떻게 되시나요?? ");
14         String name = scanner.next();
15
16         System.out.println(name + "의 나이는 " + age);
17     }
18 }
19
```

자, 지금 보시면 차이점을 느끼시겠지만, 우리가 정수형 값을 받을 때는 넥스트 인트로 했었는데, 지금 이름을 입력받은 걸 보시면 그냥 넥스트를 썼죠. 자료형이 다르기 때문에 해당 이름도 다른 건데요. 예를 들어서,



```
public static void main(String[] args) {
    System.out.println("나이가 어떻게 되시나요?? ");

    Scanner scanner = new Scanner(System.in);
    int age = scanner.nextInt();

    System.out.println("나이는: " + age);

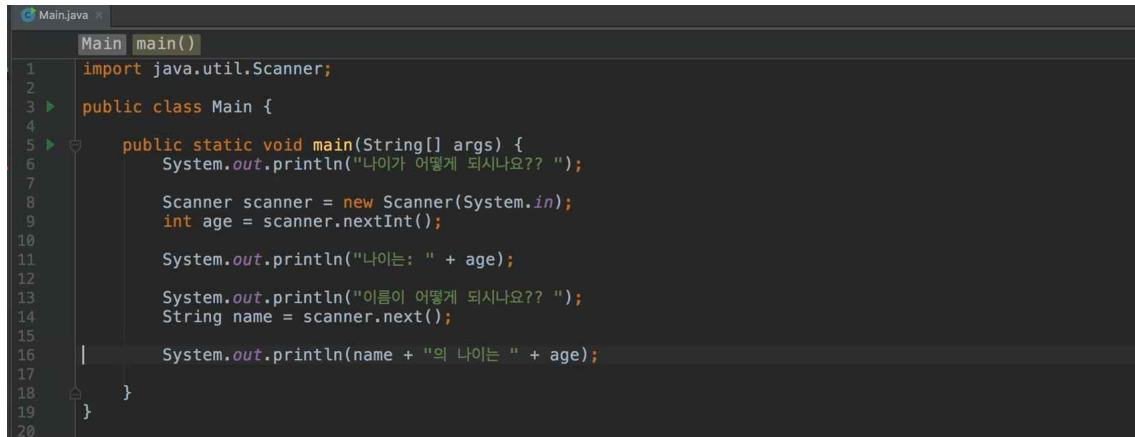
    System.out.println("이름이 어떻게 되시나요?? ");
    String name = scanner.next();

    scanner.next()
    System.out.println(name + "의 나이는 " + age);
}
```

Method	Return Type
next()	String
next(Pattern pattern)	String
next(String pattern)	String
nextBigDecimal()	BigDecimal
nextBigInteger()	BigInteger
nextBigInteger(int radix)	BigInteger
nextBoolean()	boolean
nextByte()	byte
nextByte(int radix)	byte

이런식으로 next까지 쓰게 되면, 실제로 불린이나 플롯트 등 다양하게 받을 수 있는 자동완성 기능을 볼 수 있습니다.

보셨다시피 스캐너는 어떤 데이터타입을 받느냐에 따라 사용방법이 조금씩 다릅니다.



```
1 import java.util.Scanner;
2
3 public class Main {
4
5     public static void main(String[] args) {
6         System.out.println("나이가 어떻게 되시나요?? ");
7
8         Scanner scanner = new Scanner(System.in);
9         int age = scanner.nextInt();
10
11         System.out.println("나이는: " + age);
12
13         System.out.println("이름이 어떻게 되시나요?? ");
14         String name = scanner.next();
15
16         System.out.println(name + "의 나이는 " + age);
17     }
18 }
19
20
```

다시 바꿔주고, 실행을 해보면 잘 나오는 걸 확인할 수 있습니다.

그리고 한 가지 중요한 건, 지금 우리가 배운 것들에서 나이에 이름을 입력하는 등 하게 되면 오류가 나게 되는데, 해당 방법 역시 수정할 수 있는 방법이 여럿 있지만, 지금 당장 해당 내용을 이해하기에는 조금 어렵기 때문에 나중에 강의하도록 할 테니, 그 전까지는 주의해서 사용하는 걸로 하겠습니다.

사람 이름, 음식 이름, 음식 가격은 변수를 입력받아주시고,
지불이라는 단어도 변수로 지정해주세요.

철수와 영희는 함께 음식을 준비했습니다.
철수는 토마토, 감자, 주스를 샀습니다.
영희는 떡, 마늘, 양파를 샀습니다.
토마토는 2000원, 감자는 800원, 주스는 1200원입니다.
떡은 6200원, 마늘은 2000원, 양파는 3200원입니다.
총 합은 ?원이 나왔습니다.
철수는 ?원을 지불하고
영희는 ?원을 지불하였습니다.
영희는 철수보다 ?원을 더 지불했습니다.