[Difference between “this” and“super” keywords in Java](http://stackoverflow.com/questions/4023344/difference-between-this-andsuper-keywords-in-java)

Lets consider this situation

class Animal {

void eat() {

System.out.println("animal : eat");

}

}

class Dog extends Animal {

void eat() {

System.out.println("dog : eat");

}

void anotherEat() {

super.eat();

}

}

public class Test {

public static void main(String[] args) {

Animal a = new Animal();

a.eat();

Dog d = new Dog();

d.eat();

d.anotherEat();

}

}

The output is going to be

animal : eat

dog : eat

animal : eat

The third line is printing "animal:eat" because we are calling super.eat(). If we called this.eat(), it would have printed as "dog:eat".

super is used to access methods of the base class while this is used to access methods of the current class.

Extending the notion, if you write super(), it refers to constructor of the base class, and if you write this(), it refers to the constructor of the very class where you are writing this code.

this refers to a reference of the **current** class.   
super refers to the **parent** of the current class (which called the super keyword).

By doing this, it allows you to access methods/attributes of the current class (including its own private methods/attributes).

super allows you to access public/protected method/attributes of parent(base) class. You cannot see the parent's private method/attributes.

**this is a reference to the object typed as the current class, and super is a reference to the object typed as its parent class.**

In the constructor, this() calls a constructor defined in the current class. super() calls a constructor defined in the parent class. The constructor may be defined in any parent class, but it will refer to the one overridden closest to the current class. Calls to other constructors in this way may only be done as the first line in a constructor.

Calling methods works the same way. Calling this.method() calls a method defined in the current class where super.method() will call the same method as defined in the parent class.