

Methods Calling

A young boy with short brown hair, wearing a green t-shirt, is shown in profile from the chest up. He has his mouth wide open in a shout and his hands clasped together in front of his face, palms facing each other. He is looking upwards and to the right. The background is a blurred green, suggesting an outdoor setting like a park or fair.

MOMMMM!!

DAD?!?

**CALLING
CONTEST**

At a fall fair.

Coder–Created

```
public boolean Win()
```

```
public void bClick(View view)
```

```
public void setPic(ImageView pic)
```

```
public int area (int length, int width)
```

Called on an Object

TextView

- setText

EditText

- getText
- setText

ImageView

- setImageResource

String

- replace
- charAt
- length

15.12. Method Invocation Expressions

A method invocation expression is used to invoke a class or instance method.

```
MethodInvocation:  
  MethodName ( ArgumentListopt )  
  Primary . NonWildTypeArgumentsopt Identifier ( ArgumentListopt )  
  super . NonWildTypeArgumentsopt Identifier ( ArgumentListopt )  
  ClassName . super . NonWildTypeArgumentsopt Identifier ( ArgumentListopt )  
  TypeName . NonWildTypeArguments Identifier ( ArgumentListopt )
```

The definition of `ArgumentList` from [§15.9](#) is repeated here for convenience:

```
ArgumentList:  
  Expression  
  ArgumentList , Expression
```



returnType variable = **methodName** (parameters);
methodName ();



There are 4 cases.



returnType variable = **methodName** (parameters);
methodName (parameters);



returnType variable = **methodName** (parameters);
returnType variable = **methodName** ();



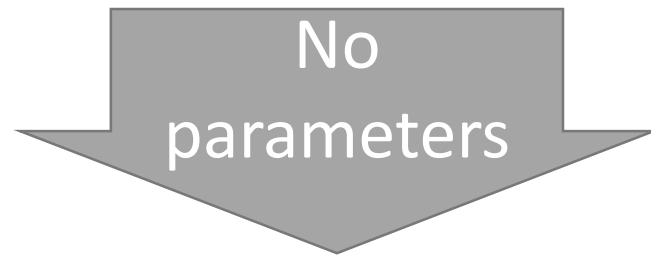
returnType variable = **methodName** (parameters);



Yours

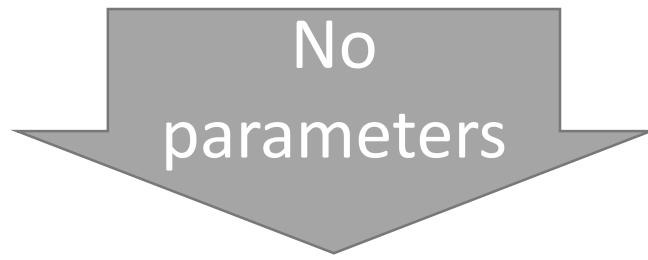
*Which boil
down to this.*

```
parameterType parameter = value;  
returnType variable = methodName (parameter);
```



public void reset ()

```
parameterType parameter = value;  
returnType variable = methodName (parameter);
```



```
public void reset ()
```

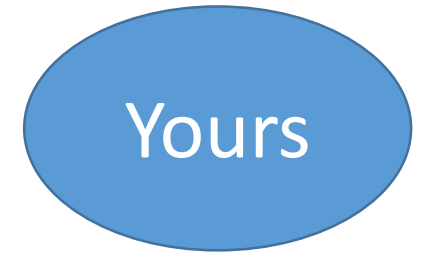
```
reset();
```

```
parameterType parameter = value;  
returnType variable = methodName (parameter);
```




```
public void setPic(ImageView i)
```

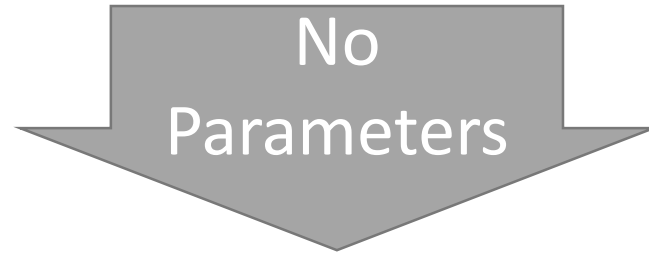
```
parameterType parameter = value;  
returnType variable = methodName (parameter);
```



```
public void setPic(ImageView i)
```

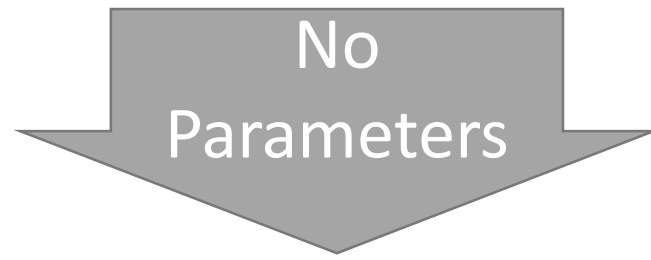
```
ImageView dice = (ImageView) findViewById(R.id.dice);  
setPic(dice);
```

```
parameterType parameter = value;  
returnType variable = methodName (parameter);
```



```
public boolean win ()
```

```
parameterType parameter = value;  
returnType variable = methodName (parameter);
```



```
public boolean win ()
```

```
TextView tv = (TextView) findViewById(R.id.text);  
boolean over = win();  
if(over)  
    tv.setText("You win! ");
```

```
parameterType parameter = value;  
returnType variable = methodName (parameter);
```



```
public boolean win(String text)
```

```
parameterType parameter = value;  
returnType variable = methodName (parameter);
```



```
public boolean win(String text)
```

```
EditText et = (EditText) findViewById(R.id.edit);  
TextView tv = (TextView) findViewById(R.id.text);  
String response = et.getText().toString();  
if(win(response))  
    tv.setText("You win! ");
```

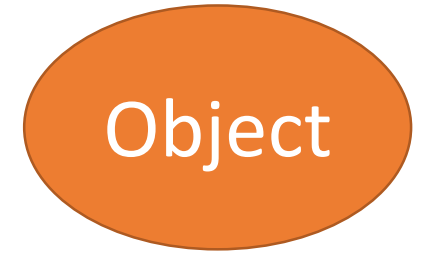
```
parameterType parameter = value;  
returnType variable = methodName (parameter);
```

```
parameterType parameter = value;  
returnType variable = methodName (parameter);
```



Yours

1. public void tree()
2. public String maple()
3. public void oak (String leaf)
4. public char pine (int needles)
5. public void spruce (boolean bird)
6. public boolean isTree()
7. public void chestnut()
8. public void cherry(ImageView i)
9. public int crabApple(char rotten)



returnType variable = **object.methodName** (parameters);
object.methodName ();

There are 4 cases.



returnType variable = **object.methodName** (parameters);
object.methodName (parameters);



returnType variable = **object.methodName** (parameters);
returnType variable = **object.methodName** ();



returnType variable = **object.methodName** (parameters);



Object

*Which boil
down to this.*

parameterType parameter = value;

returnType variable = object.methodName (parameter);

```
public void setText(String text)
```

```
TextView t = (TextView) findViewById(R.id.t);  
t.setText("Hello");  
t.setText(3+ "");
```

```
public String replace(char oldChar, char newChar)
```

```
String s = "kitty cat";  
s = s.replace('t', 'z');
```

```
parameterType parameter = value;  
returnType variable = object.methodName (parameter);
```

parameterType parameter = value;

returnType variable = object.methodName (parameter);



Object

1. public void cat()

2. public void yak(EditText text)

3. public char dog()

4. public int deer(char antler)

5. public void giraffe (boolean tree)

6. public void horse()

7. public String zebra (int stripe)

8. public ImageView cow()

9. public void mouse (double squeak)