

Java

1.URLHTML

2.

3.

```
public class DateWork {  
    private int remainDays = 10; //  
  
    public void calculateNameWithData(Data data) {  
  
    }  
}
```

`equalsComparatorUtils.equals(Object obj1, Object obj2)`

```
if (a.equals(b)) {  
    // do something  
}
```

```
if (ComparatorUtils.equals(a, b)) {  
    // do something  
}
```

`"abcde"null`

```
if ("abcde".equals(op)) {  
    // do something  
}
```

50
`API4HTTPResponseRequest`
`if-if-elseforwhiledoswitch“{ }”`

```
if (statement) return;
```

```
if (statement) {  
    return;  
}
```

```
if (booleanExpression) {
    return true;
} else {
    return false;
}
```

```
return booleanExpression;
```

//private

publicpublic

public——protected——private

```
public class Test {
    protected String myProtectedData = "Protected";
    public static final String AAA = "aaa";
    private String myPrivate = "Private";
    public String myData = "data";
}
```

```
public class Test {
    public static final String AAA = "aaa";
    public String myData = "data";
    protected String myProtectedData = "Protected";
    private String myPrivate = "Private";
}
```

static final

```
public double area(double r) {
    return 3.14 * r * r;
}
```

```
private static final int PI = 3.14;
public double area(double r) {
    return PI * r * r;
}
```

```
return a + b > 0 ? "student" : "teacher";
```

```
return (a + b > 0) ? "student" : "teacher";
```

```
public static final String WidgetName = "WidgetName";
```

```
public static final String WIDGET_NAME = "WidgetName";
```

setis/hasget

```
public class Student {  
    private boolean male;  
  
    public boolean getMale() {  
        return male;  
    }  
}
```

```
public class Student {  
    private boolean male;  
  
    public boolean isMale() {  
        return male;  
    }  
}
```

nullnull

Utils

```
public class Utils {  
    public static Border createBorder(Font font, Color color) {  
        if (color == null) {  
            color = new Color(223, 122, 123);  
        }  
    }  
}
```

```
Border border = Utils.createBorder(font, null);
```

```
//Utils
public static Border createBorder(Font font) {
    return createBorder(font, null);
}
//
Border border = Utils.createBorder(font);
```

super, FCloneclone

```
@Override
public void doSomething() {
    super.doSomething();
}
```

```
String shapeName = "";
```

```
String shapeName = Inter.getLocText("FR-Designer_Chart_Circle");
```

com.fr.chart.axis

```
String strArray[];
```

```
String[] strArray;
```

```

public String[] getNames4JionTheParty() {
    if (a) {
        return new String[] { "", "", "" };
    } else (b) {
        return new String[0];
    }
}

```

if -

```

if (a > -1 && a != 1 && dim.width > 0 && dim.height > 0) {
    doSomething();
}

```

```

if (shouldDoSomething(a, dim)) {
    doSomething();
}
boolean shuouldDoSomething(int a, Dimension dim) {
    return a > -1 && a != 1 && dim > width && dim.height > 0;
}

```

HashMapHashSetHashTablekeyequalshashCode

switchcasebreakcasebreak

```

public String diffResult(int type) {
    String someDescription = "abc";
    switch (type) {
        //
        case 1:
            someDescription = "def";
        case 2:
            someDescription = "xyz";
            break;
        default:
            someDescription = "mnx";
            break;
    }
    return someDescription;
}

```

-

```

try {
    doSomething();
} catch (Exception e) {
    e.printStackTrace();
}

```

```
try {
    doSomething();
} catch (Exception e) {
    FRContext.getLogger.error(e.getMessage, e);
}
```

ignore

```
try {
    doSomething();
} catch (Exception ignore) {
}
```

JavaJSONJSON

```
{a:"bb"};
{'a':"bb"};
```

```
{"aa":"bb"};
```

JSONJavaJSONArray

JSONObject

```
JSONObject jo = JSONObject.create();
String[] names = new String[]{"aaa", "bbb", "ccc"};
jo.put("names", names);
```

JSON

```
JSONObject jo = JSONObject.create();
JSONArray ja = new JSONArray(); ja.put("aaa").put("bbb").put("ccc"); jo.put("names", ja);
```

StringUtil.EMPTY

```
if (name == null) {
    return "";
}
if (name == null) {
    return StringUtil.EMPTY;
}
```

StringUtil#isEmpty(String);

```
String name = createByAge(20);  
if ("".equals(name)) {  
  
}  
if (StringUtils.isEmpty(name)) { //  
}
```

toStringGeneralUtils#objectToString(Object)

```
Key key = getKey();  
String result = key.toString(); //  
String okStr = GeneralUtils.objectToString(key); //
```

HTTPNetworkHelper#getHTTPRequestParameter(HttpServletRequest,String)

HttpServletRequest#getParameter(String)