

kotlin

wy

October 24, 2017

## Contents

<b>1</b>	<b>keyword</b>	<b>3</b>
1.1	as	3
1.2	as? Type	3
1.3	by	3
1.4	delegate user agent	3
1.5	dynamic	3
1.6	get,set	3
1.7	where	3
1.8	open,final	3
1.9	infix	3
1.10	inner	4
1.11	private,protected,internal,public	4
	1.11.1 in top-level in file	4
	1.11.2 in class,interface	4
<b>2</b>	<b>doc</b>	<b>5</b>
2.1	fun call	5
2.2	fun override	5
2.3	var val	5
	2.3.1 var	5
	2.3.2 val	5
2.4	switch,when	5

## **1 keyword**

### **1.1 as**

1. as Type convert variable to Other Type
2. as Name alias otherName for import

### **1.2 as? Type**

1. as convert variable to Other Type?

### **1.3 by**

1. let the impl of interface delegate to another object
2. delegate set/get value for var,or get value for val.  
operator fun setValue(thisRef:Any?,property: KProperty<\*>)  
operator fun getValue(thisRef:Any?,property: KProperty<\*>)

### **1.4 delegate user agent**

### **1.5 dynamic**

1. off kotlin type check for target JS
2. use JS dynamic type

### **1.6 get,set**

1. get , var/val v : Type get() = v
2. set , var v: Type set(value) v = value;; can't init

### **1.7 where**

1. genericity must have ability;

### **1.8 open,final**

1. final ,can't inherit
2. open , opposite of Java's finalss

### **1.9 infix**

1. one method hava only one parameter
2. marked as infix
3. require last two,you can call it by — type fun param —

## **1.10 inner**

1. mark class as an inner class

## **1.11 private,protected,internal,public**

1. the default access control is public
2. inherit access control.

### **1.11.1 in top-level in file**

1. public: can access every one
2. private: can access only in file
3. internal: can access in the same module
4. protected: illegal

### **1.11.2 in class,interface**

1. private: can access only in this class
2. protected: access int this class and it's subclass
3. internal: can access the internal member of which class it can see
4. public: can access the member of which class it can see

## **2 doc**

### **2.1 fun call**

1. fun a(a:Int=1,b:Int), can be call by named param,i.e. a(b=2);
2. named param should place after general param

### **2.2 fun override**

1. use keyword override, override fun b()
2. the default param's value will be the base fun's default value

### **2.3 var val**

#### **2.3.1 var**

declare mutable variable

#### **2.3.2 val**

declare read-only variable

### **2.4 switch,when**