1.

@Controller@Service("iCategoryService")

都包含@Component注解，所以会被自动扫描扫描到

放在类的上面

示例：

@Component  
public class Knight {  
 public void speak() {  
 System.*out*.println("knight speak");  
 }  
}

扩展：

1. 别名：为bean指定别名

示例:

@Component("main\_battle")  
public class Battle {  
  
 @Autowired  
 private Knight knight;  
  
 public void battle() {  
 knight.speak();  
 }  
}

[2.@ComponentScan](mailto:2.@ComponentScan)默认会扫描同包下的@Component注解

默认属性会扫描参数包下的@Component注解

放在配置类的上面

示例：

@Configuration  
@ComponentScan("com.springjiemi.test")  
public class ScanConfig {  
}

扩展：

1. basePackages = “com.springjiemi.test” //指定要扫描的包，注意是复数形式，故可以添加多个包

示例:

@Configuration  
@ComponentScan(basePackages = {"com.springjiemi.test", "com.springjiemi.dao"})  
public class ScanConfig {  
}

1. basePackageClasses = DaoForScan.class //指定要扫描的包，通过包里面所包含的某个类或接口

示例：

@Configuration  
@ComponentScan(basePackageClasses = DaoForScan.class)  
public class ScanConfig {  
}

3. @Configuration

用于定义配置文件，代替web.xml

放在配置类的上面

示例：

@Configuration  
@ComponentScan("com.springjiemi.test")  
public class ScanConfig {  
}

4.@Autowired注解用于自动装配bean，使用此注解的类需要类上面有@Component注解

示例：

@Component  
public class Battle {  
   
 @Autowired  
 private Knight knight;  
   
 public void battle() {  
 knight.speak();  
 }  
}

扩展：

1. 此注解也可以用于其他方法上，spring会尝试满足所声明的依赖。
2. 可以使用required = false 属性设置这个依赖不是必须的（但有可能抛出NPE异常，谨慎）

示例:

@Autowired(required = false)  
private Ghost ghost;

5．@bean 和 @Component类似，都是声明为bean，但@bean是用在javaConfig类的方法上

示例1：

@Configuration  
public class ScanConfig {  
 @Bean  
 public Ghost ghost() {  
 return new Ghost();  
 }  
}

示例2：添加参数，spring会自动装配参数

@Configuration  
public class ScanConfig {  
 @Bean  
 public Ghost ghost(LittleRedHat littleRedHat) {  
 return new Ghost(littleRedHat);  
 }  
}

扩展：

1. name=“” //设置bean的名字
2. javaConfig中注册的Bean的高级应用:随机生成bean

示例：

@Configuration  
public class ScanConfig {  
 @Bean  
 public Human human() {  
 int random = (int) Math.*floor*(Math.*random*() \* 2);  
 if (random == 0) {  
 return new Knight();  
 } else {  
 return new LittleRedHat();  
 }  
 }  
}

6，@Import(xxxConfig.class) //引入其他配置文件

@Configuration  
@ComponentScan("com.springjiemi.test")  
@Import(BeanConfig.class)  
public class ScanConfig {  
}

扩展：

1. 使用更高级别的Config组合低级别的Config

示例：

@Configuration  
@Import({BeanConfig.class, ScanConfig.class})  
public class Config {  
}

1. @ImportResource(“classpath:xxxx.xml”) //引入基于xml的配置文件

示例：

@Configuration  
@Import(ScanConfig.class)  
@ImportResource("classpath:bean-config.xml")  
public class Config {  
}