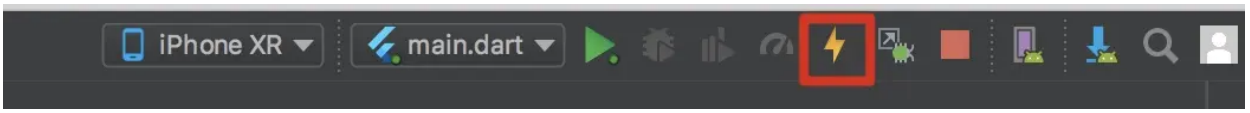
一、Hot Reload使用

在程序run起来之后，我们可以通过as点击Hot Reload图标来进行热重载



点击图标中的闪电图标即可进行热重载

二、Hot Reload的原理

Hot Reload只能在debug模式下使用，因为debug模式下使用JIT动态编译，代码运行在Dart vm,JIT会将Dart代码编译成可以运行在Dart vm沙灰姑娘的Dart Kernel，而Dart Kernel是可以动态更新的，因此可以实现代码实时更新

过程：

1、首先扫描代码，找到和上次编译只有有变化的Dart代码

2、将这些变化的代码转化为增量的Dart Kernel文件

3、增量的Dart Kernel发送到正在移动设备上运行的Dart Vm

4、Dart vm将增量的Dart kernel与原有的文件进行合并，然后重新加载全新的Dart Kernel

5、虽然重新加载了Dart Kernel，但是不会重新加载代码，而是通知Flutter Framework重建Widget

总结来看，Hot Reload不会重新执行一次代码，而是触发Flutter重新绘制改变的Widget，并且保留Flutter之间的状态。

三、无法使用Hot Reload的场景

1、代码编译错误

需要解决编译错误后才能使用

2、代码更改会影响app状态

代码修改导致前后状态发生改变时，无法使用，例如某个组件从无状态改成有状态，此时使用HotReload无效

3、全局变量和静态字段更改

在flutter中，全局变量和静态字段被视为状态，HotReload期间不会重新初始化

4、main（）方法中的更改

main方法不会在HtoReload过程中重新执行

5、枚举改为常规类或者常规类改为枚举

```
enum Color {
 red,
 green,
 blue
}
```

改为:

```
class Color {
 Color(this.i, this.j);
 final int i;
 final int j;
 }
```

6、修改通用类型声明

```
class A<T> {
 T i;
}
```

改为:

```
class A<T, V> {
 T i;
 V v;
}
```

四、Hot Reload和Hot Restart

针对无法使用HotReload的情形就需要使用Hot Restart，Hot Restart可以重启应用程序而无需结束调试会话