# Сопрограммы – Coroutines

# Сопрограммы обеспечивают возможность избежать блокировки исполняющегося потока путём использования более дешёвой и управляемой операции:  приостановки (suspend) сопрограммы.

# Сопрограммы упрощают асинхронное программирование, оставив все осложнения внутри библиотек.

# Логика программы может быть выражена последовательно в сопрограммах, а базовая библиотека будет её реализовывать асинхронно для нас

Блокирование против приостановки

# Главным отличительным признаком сопрограмм является то, что они являются вычислениями, которые могут быть приостановлены без блокирования потока (вытеснения средствами операционной системы).

# Еще одно отличие заключается в том, что сопрограммы не могут быть приостановлены на произвольной инструкции, а только в так называемых точках остановки (приостановки), которые вызываются в специально маркируемых функциях.

Приостановка происходит в случае вызова функции, обозначенной специальным модификатором suspend:

suspend **fun** doSomething(foo: **Foo**): Bar {

...

}

Такие функции называются функциями остановки (приостановки), поскольку их вызовы могут приостановить выполнение сопрограммы

Функции остановки могут иметь параметры и возвращать значения точно так же, как и все обычные функции, но они могут быть вызваны только из сопрограмм или других функций остановки. В конечном итоге, при старте сопрограммы она должна содержать как минимум одну функцию остановки, и функция эта обычно анонимная (лямбда-функция остановки). Давайте взглянем, для примера, на упрощённую функцию async() (из библиотеки [kotlinx.coroutines](https://kotlinlang.ru/docs/reference/coroutines.html#generators-api-in-kotlincoroutines)):