

launch

```
public fun launch(  
    context: CoroutineContext = DefaultDispatcher,  
    ...,  
    block: suspend CoroutineScope.() → Unit  
): Job
```

```
val job: Job = launch {  
    val x = doSomething()  
    doSomethingElse(x)  
}
```

async

```
public fun <T> async(  
    context: CoroutineContext = DefaultDispatcher,  
    ...,  
    block: suspend CoroutineScope.() → T  
) : Deferred<T>
```

```
public suspend fun <T> Deferred<T>.await(): T
```

```
val asyncValue: Deferred<String> = async {  
    val x = doSomething()  
    doSomethingElse(x)  
}
```

```
val value = asyncValue.await()
```