

Teilweise hat man die Kontrolle darüber welches von beiden man verwendet, meistens wird das aber automatisiert, so dass man keine Kontrolle darüber hat.

Bei primitiven Datentypen (bool, char, short, int, long) wird meistens ein Call-by-Value genutzt und bei Arrays und Listen wird meistens ein Call-by-Reference genutzt. Das ist aber von Programmiersprache zu Programmiersprache unterschiedlich.

In Java ist es bspw. genauso, in C++ werden Pointer verwendet wenn man einen Call-by-Reference haben möchte und bei C# kann man ein Schlüsselwort mit angeben um ein Call-by-Reference nutzen zu können.

Beim Call-by-Value wird also die Variable kopiert, wenn sie als Argument an eine Funktion übergeben wird, so dass sich die Variable nach der Funktionsausführung nicht ändert.

```
void function foo(int a){
```

```
    a = 1337
```

```
}
```

```
int a = 42
```

```
foo(a)
```

```
print(a) // 42
```

Beim Call-by-Reference wird die Speicheradresse der Variablen übergeben, so dass die tatsächlichen Werte die im Speicher liegen in der Funktion verwendet werden und nicht eine Kopie, die erneut Speicher belegt.

```
void function foo(List list) {  
    list = {4, 5, 6}  
}  
List list = {1, 2, 3}  
foo(list)  
print(list) // 4, 5, 6
```

Wenn man nun einen primitiven Datentypen den eigentlich durch einen Call-by-Value an die Funktion übergeben wird als Call-by-Reference übergeben will, kann man bspw. das `ret`-Schlüsselwort nutzen.

```
void function foo(ret int a) {  
    a = 1337  
}  
int a = 42  
foo(a)  
print(a) // 1337
```

So kann man im Prinzip mehrere Rückgabewerte aus einer Funktion heraus kriegen.