

# Call by reference / call by value

---

## Ref Schlüsselwort

- Manchmal möchten wir die Möglichkeit haben, den Wert einer übergebenen Variablen in einer Funktion zu ändern
- Normalerweise wird der Wert der übergebenen Variablen kopiert (call by value) und an die Funktion übergeben
- Mit dem Schlüsselwort `ref` wird jedoch nicht der Wert, sondern die Referenz auf die Variable übergeben
- Dadurch können wir den Wert des Parameters ändern

```
public static void Main()
{
    string text = "my name is...";

    Console.WriteLine(text);
    SetValue(ref text);
    Console.WriteLine(text);
}

public static void SetValue(ref string text)
{
    if (text == "my name is...") {
        Console.WriteLine("Julian");
    }
    text = "TEST!!!";
}
```

## Out-Schlüsselwort

- Es gibt auch ein anderes Schlüsselwort, das beim Aufruf durch Referenz hilft
- Das `out`-Schlüsselwort ist dem `ref`-Schlüsselwort ziemlich ähnlich
- Der große Unterschied besteht jedoch darin, dass die mit dem Schlüsselwort `out` übergebenen Variablen vor dem Übergeben nicht initialisiert werden müssen
- Es ist eine Möglichkeit, zum Beispiel mehr Variablen anstelle eines einzigen Rückgabewerts zurückzugeben

## Statische vs. lokale Variablen

---

- Wenn du Variablen innerhalb Ihrer Methode erstellst, werden diese Variablen immer von Grund auf neu erstellt, wenn du die Funktion aufrufst
- Oft möchtest du dieses Verhalten, da die verwendeten Variablen nur die Funktion bzw. eine Berechnung in der Funktion unterstützen
- Wenn du jedoch Variablen haben möchtest, die von allen Funktionen (Lesen und Schreiben) verwendet werden können, kannst du statische Variablen verwenden
- statische Variablen werden außerhalb der Funktionen erstellt

```
static int a = 5;

public static void Main()
{
    PrintA();
    ChangeA();
    PrintA();
    ChangeA();
    PrintA();
}

public static void PrintA()
{
    Console.WriteLine("nuber: {0}", a);
}
```

```
public static void ChangeA()
{
    Random random = new Random();
    a = random.Next(0, 10);
}
```