

The background of the slide is a complex network of thin grey lines connecting various circular nodes. The nodes vary in size and color, including dark blue, light blue, and grey. Some nodes are highlighted with larger, concentric circles. A large black rectangle is positioned in the lower right quadrant, containing the text 'TUPEL' and the speaker's name and affiliation.

TUPEL

Vincent Uhlmann
IT-Akademie Dr. Heuer GmbH

TUPEL (VALUETUPLE)

- Einführung in C# 7.0 mit .NET Framework 4.7
- Datenstruktur, die das Gruppieren mehrerer Datenelemente ermöglicht

SYNTAX

- Erstellung mit runden Klammern, Elemente durch Kommas getrennt
- Namen für die Variablen sind optional

```
(double, int) t1 = (4.5, 3);  
Console.WriteLine($"Tuple mit den Elementen {t1.Item1} und {t1.Item2}.");
```

```
(double X, int Y) t2 = (4.5, 3);  
Console.WriteLine($"Tuple mit den Elementen {t2.X} und {t2.Y}.");
```

- Verwendung der `ValueTuple.Create()` Methode

```
(int x, int b) z = ValueTuple.Create(1, 2);
```

SYNTAX

- Methoden können in einem Tupeltyp nicht definiert werden
- Von .NET bereitgestellte Methoden können verwendet werden

```
(int, int) t = (4, 3);  
Console.WriteLine(t.ToString()); // (4, 3)
```

ANWENDUNGSFÄLLE

- Häufig als Methodenrückgabebetyp

```
public static void Main()
{
    (int I, int X) d = Test();
    Console.WriteLine(d.I);
    Console.WriteLine(d.X);
}

private static (int I, int X) Test()
{
    return (1, 2);
}
```

TUPELGLEICHHEIT

- Tupeltypen unterstützen die Operatoren == und !=
- Member des linken Operanden werden gemäß der Reihenfolge der Tupelelemente mit den Membern des rechten Operanden verglichen

```
(int a, byte b) left = (5, 10);  
(long a, int b) right = (5, 10);  
Console.WriteLine(left == right); // output: True  
Console.WriteLine(left != right); // output: False
```

UNTERSCHIEDE VALUETUPLE UND TUPLE

- `System.ValueTuple` ist ein Werttyp, während `System.Tuple` ein Referenztyp ist
- `System.ValueTuple` ist veränderlich (mutable), während `System.Tuple` unveränderlich (immutable) ist
- `System.ValueTuple` macht seine Elemente über Felder zugänglich, während `System.Tuple` diese über Eigenschaften macht
- `System.ValueTuple` ermöglicht die Verwendung von benannten Parametern

VAR

- var wird verwendet, um implizit typisierte lokale Variablen zu deklarieren
- Compiler leitet den Typ der Variable aus dem Initialisierungsausdruck ab
- Typ wird zur Kompilierzeit bestimmt
- var kann verwendet werden, um Tupel zu deklarieren, ohne explizit die Typen der Elemente anzugeben

```
var result = ValueTuple.Create(4, 2);  
Console.WriteLine(result.Item1);
```

```
var person = (Name: "John", Age: 30);  
Console.WriteLine(person.Name);
```