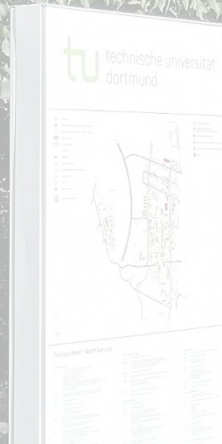


# Zeiger in C

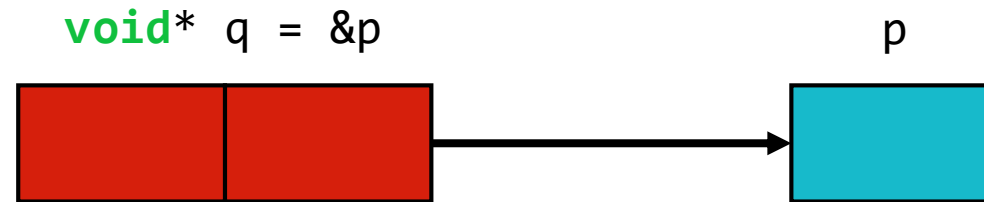
Emilio Pielsticker





# Zeiger

Bei einem **Zeiger (Pointer)** handelt es sich um eine Variable im Speicher, welche eine Adresse enthält und mit dieser auf eine **Speicherzelle** zeigt



`*q` : Referenzierungsoperator

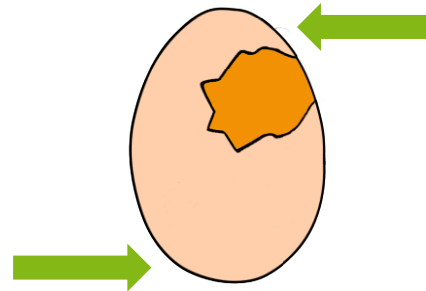
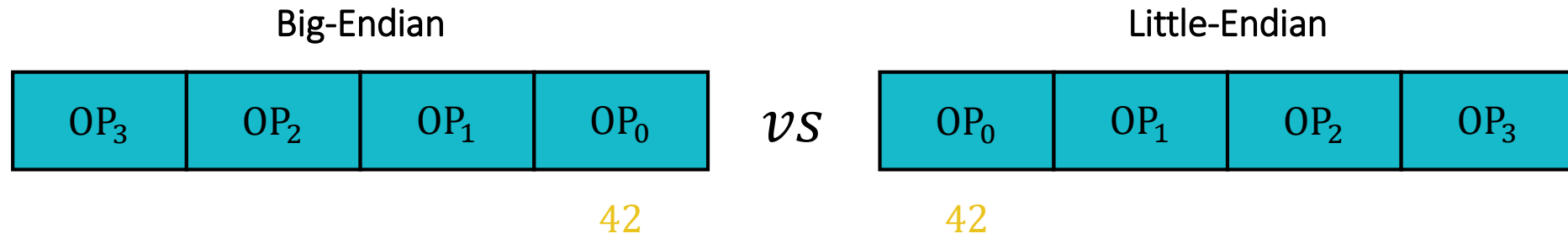
`&q` : Adressoperator

`(*elem).attr = elem->attr`

# Zusammenhang zwischen Feldern und Zeigern

- Beispiel: `int a[4] = {4, 3, 2, 1};`
- `a` ist eigentlich ein Zeiger des Typs `int*`
- `int* c = a` ist gleichbedeutend zu `int* c = &(a[0])`
- Zeiger lassen sich umgekehrt wie Feldbezeichner verwenden
- Beispiel: `a[n] ≡ *(c + n)`

# Endianness



# Gießen (Casting) von Zeigern

- Wie schreibt man den Integer-Wert `0x12345678` in einen Bereich der Größe 42 mit `char* x = malloc(42)`?
- Der Zeiger `x` wird **gegossen** (gecastet):
  - `x` ist ein Byte-Zeiger auf ein `char`
  - `(int*)(x)` castet `x` zu einem Zeiger auf ein `int`
  - Wie gewohnt dereferenzieren:
- `*((int*)(x)) = 0x12345678`