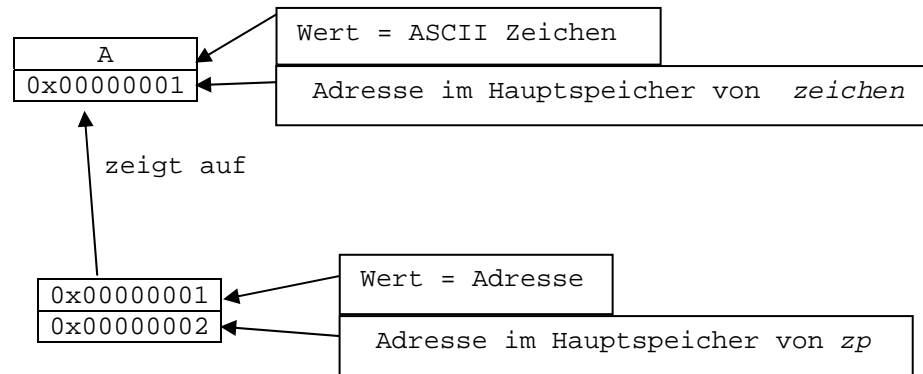


Pointer = Zeiger

```
char zeichen;  
char * zp;
```



```
zeichen = 'A';  
zp = &zeichen;
```

Man sagt: `zp` zeigt auf `zeichen` !

`*zp` dereferenziert die Adresse von `zeichen` - es löst die Adresse auf.

`*zp` liefert den Wert **'A'**, also das, was in der Adresse steht, worauf `zp` zeigt.

`printf("%c", zeichen);` → Gibt Wert von `zeichen` auf `stdout`(=Bildschirm)
aus. → **A**

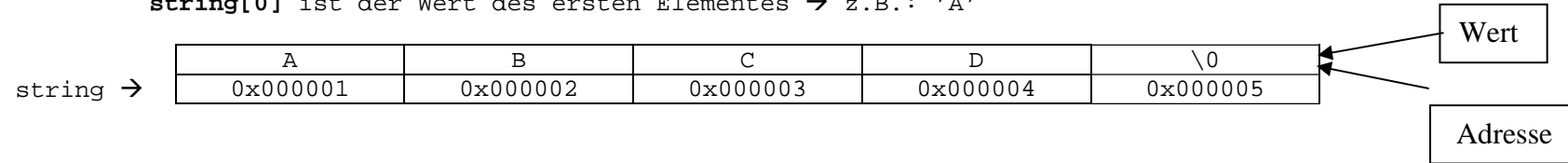
`scanf("%c", &zeichen);` → Speichert mir beliebiges **char** von
`stdin`(=Tastatur) in `zeichen`.

`printf("%p", zp);` → Gibt Wert von `zp` auf `stdout`(=Bildschirm)
aus. → **0x00000001**

`printf("%c", *zp);` → Gibt Wert von `zp` auf `stdout`(=Bildschirm)
aus. → **A**

Zeiger - Fortsetzung

`char string[5];` `string` zeigt auf die Startadresse von `char string[5]` → `0x000001`
`string` ist also ein Zeiger !
`string[0]` ist der Wert des ersten Elementes → z.B.: 'A'

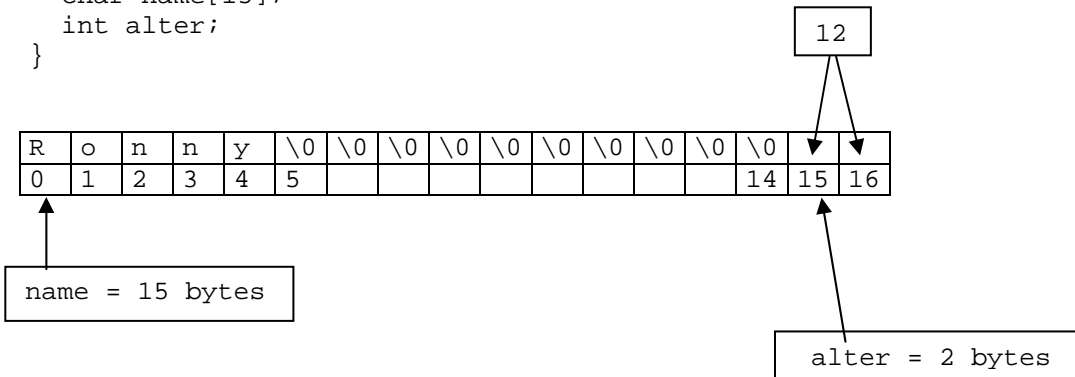


`printf("%c", string);` → A // Wert der Startadresse, also das erste Zeichen.
`printf("%c", string + 1);` → B // Wert der Startadresse + 1, also das zweite Zeichen.
`printf("%c", string[1]);` → B // Wert der Startadresse + 1, also das zweite Zeichen.

struct = Strukturen

struct erlaubt das definieren eigener Datentypen.

```
struct Schueler // name und alter stehen im Speicher direkt hintereinander.  
{  
    char name[15];  
    int alter;  
}
```



Schueler s; // statische Variable: '.' (Punkt) ist Dereferenzierungsoperator.

```
strcpy(s.name, "Ronny");  
s.name = 12;  
printf("%c", s.name); → "R"  
printf("%s", s.name); → "Ronny"
```

Schueler * ss; // dynamische Variable: '->' (Pfeil) ist Dereferenzierungsoperator.

Man beachte: ss ist ein Zeiger, der auf NULL zeigt, ich muss erst mittels der Funktion **malloc()** Speicher anfordern.

```
ss = (Schueler*) malloc( sizeof(Schueler) );
```

```
strcpy(s->name, "Ronny");  
s->name = 12;  
printf("%c", s->name); → "R"  
printf("%s", s->name); → "Ronny"
```

FILE = Dateien

```
FILE * f; // f kann auf einen Speicherbereich zeigen, in dem eine Datei gespeichert ist.  
          // noch zeigt f auf NULL.  
  
fopen() öffnet eine Datei und liefert einen Zeiger auf diese (FILE *).  
  
f = fopen("Test.txt","r"); // öffnet die Datei "Test.txt" zum Lesen.  
                          // f zeigt jetzt auf den Dateianfang.  
  
char c;  
c = fgetc(f);           // fgetc() liest demnach das erste zeichen der Datei und speichert es in c.
```