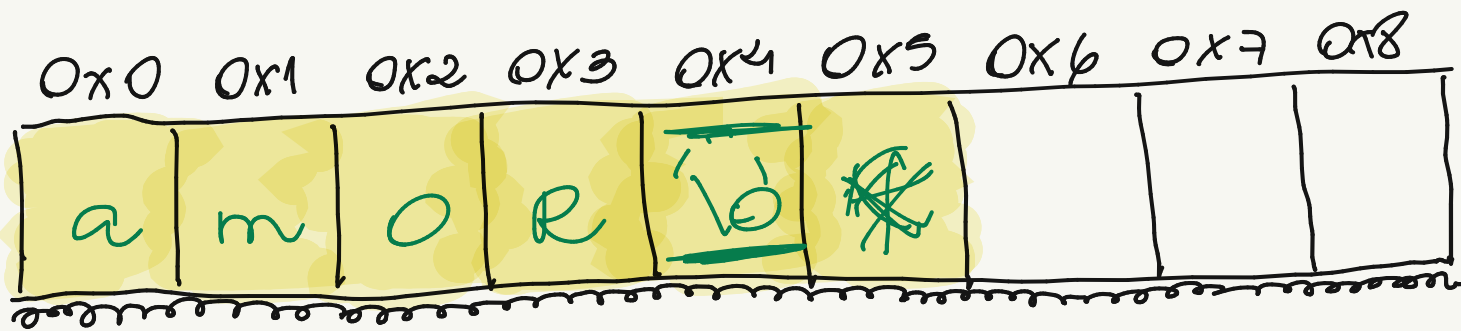


Vetor de caracteres



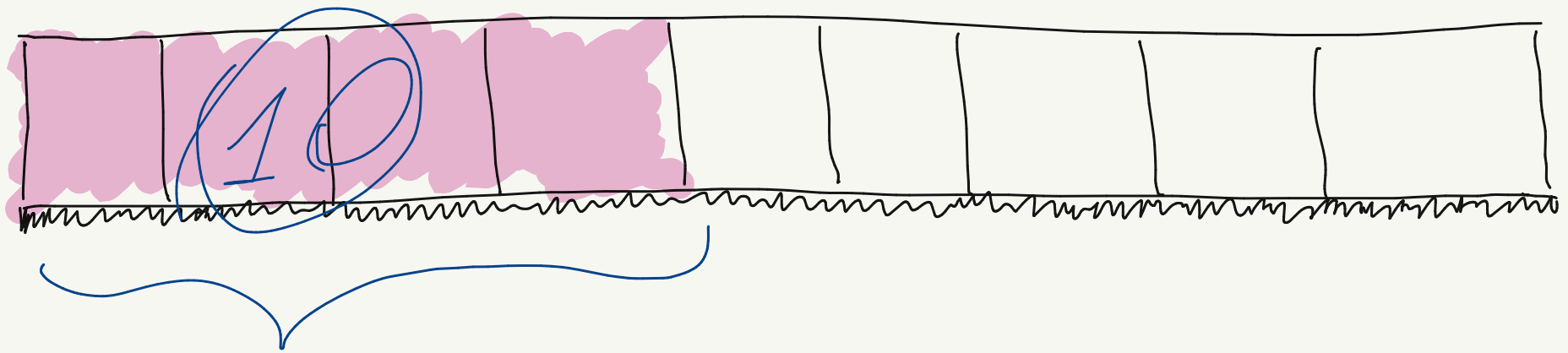
char palavra[6];

↪ tamanho do vetor de caracteres

strlen(palavra) = 4 letras até o caractere '\0'

⇓

Biblioteca string.h



a

short → 1 Byte
int → 4 Bytes

} números inteiros

char → 1 Byte → caractere

float → 4 Bytes
double → 8 Bytes

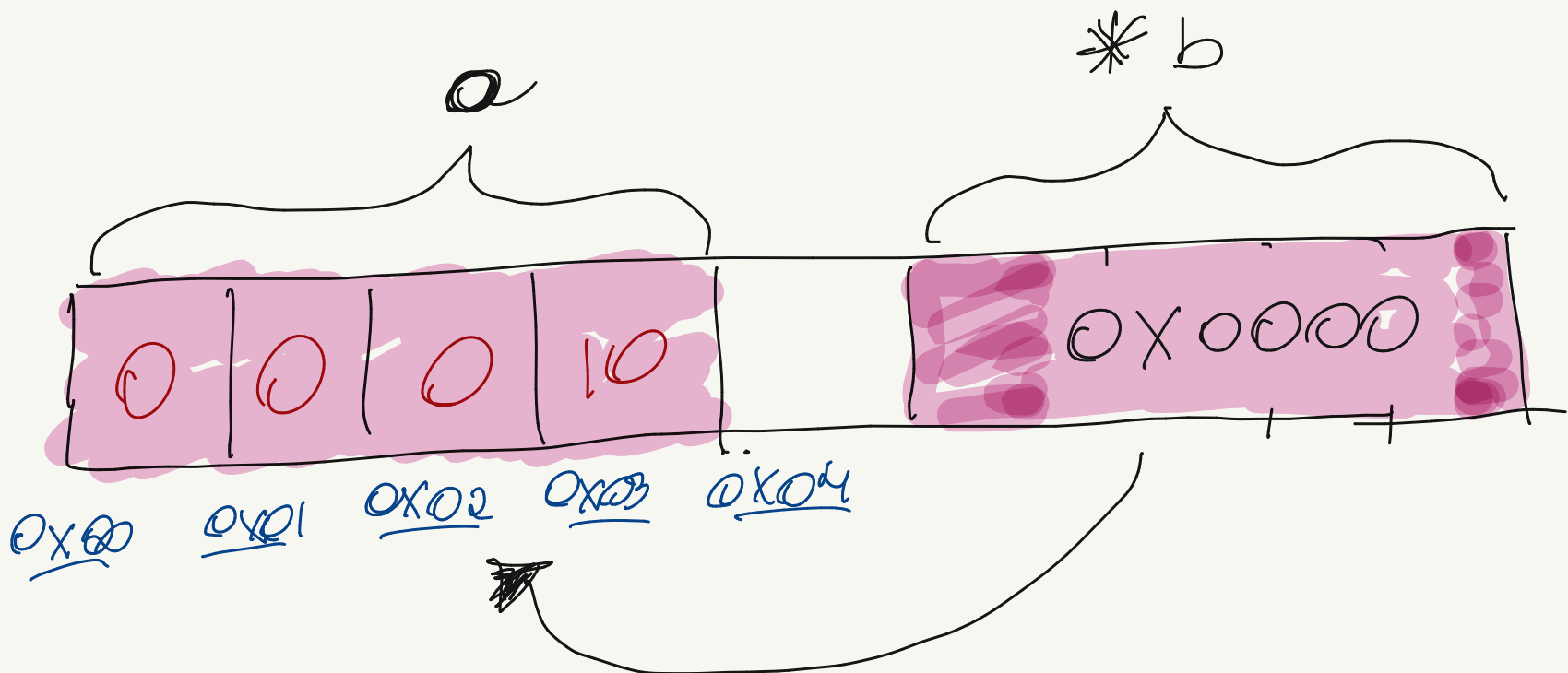
} números com
parte fracionária

int a = 10;

nome: a

valor: 10

endereço: 0x0012



uint a = 10;

uint *b = &a;

(a)
&a

[b] = 0x0000 \Rightarrow endereço a

&b = 0x0005 \Rightarrow endereço b

*b = 10 \Rightarrow conteúdo que b aponta

vetor de caracteres

char palavra [10];

1 Byte

↳ Reservado uma região de memória para 10 caracteres.

palavra [0] = 'a';

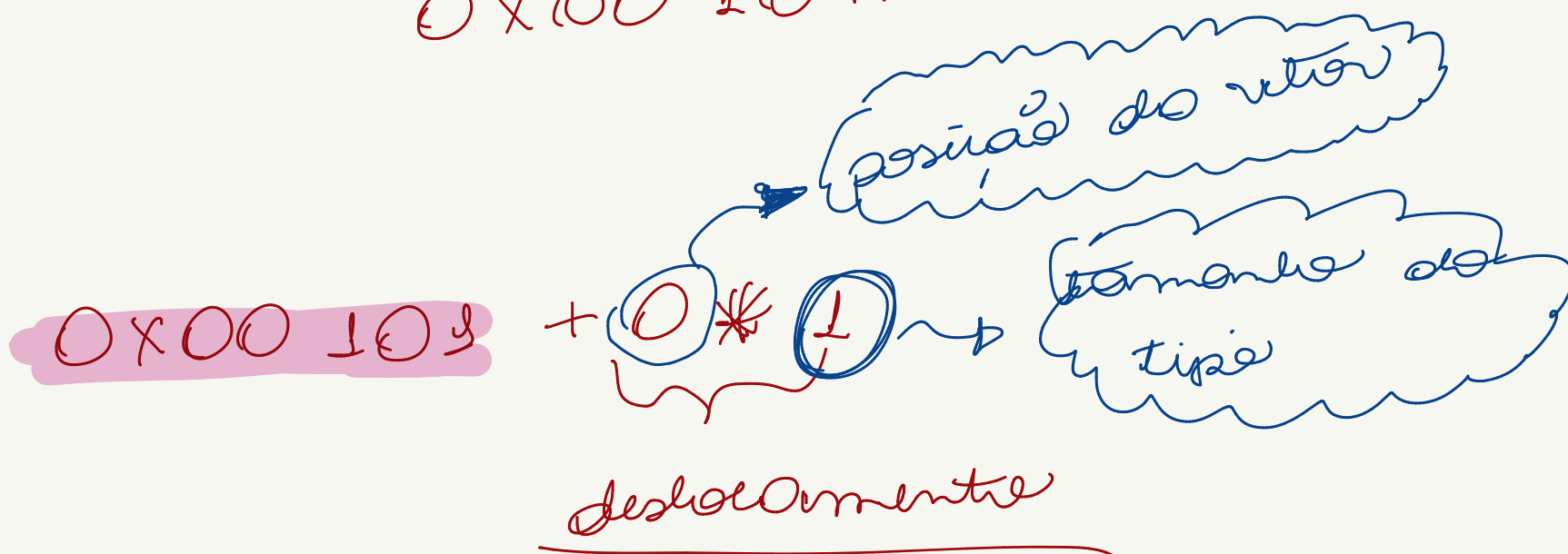
palavra [1] = 'm';

palavra [2] = ~~'o'~~; '\0';

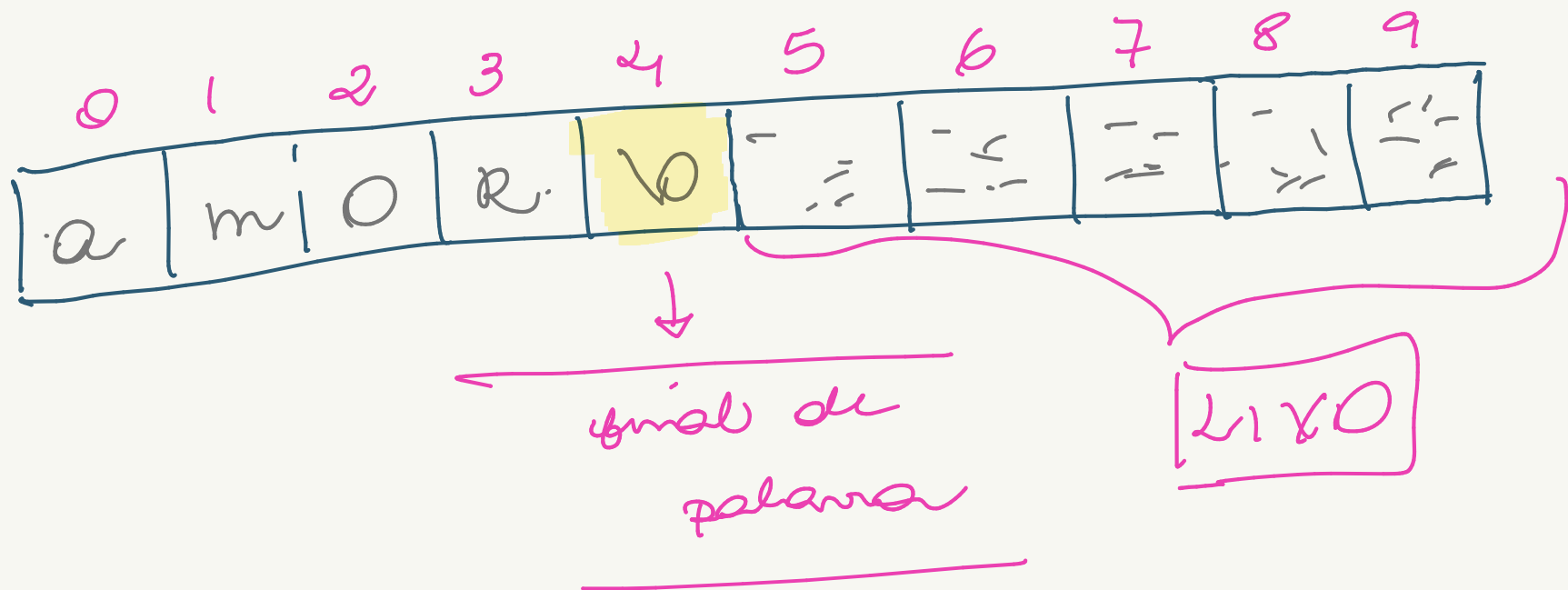
palavra [3] = 'r';

palavra [4] = '\0'; → fim de palavra

palavra → 0x00101 } 10 posições
0x0010A }



char palavra [10];



strlen(palavra) = 4

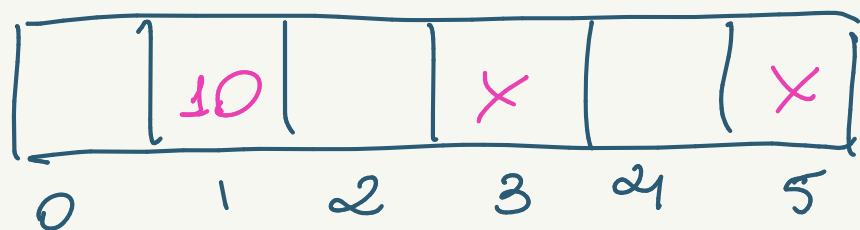
↳ tamanho da minha
palavra até o '\0'

palavra[0] = 'a'

palavra[pos] =

Arranjos ou Vetores

→ forma de agrupar um conjunto de dados do mesmo tipo.

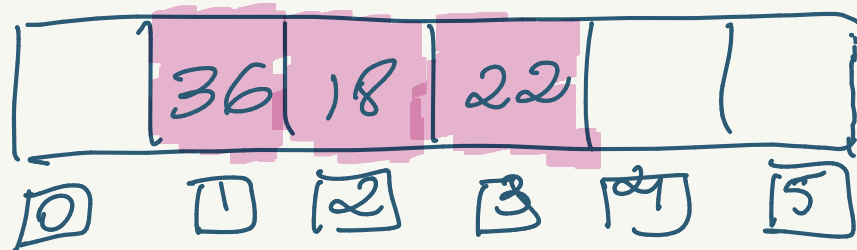


→ int char
float short
double

short a = 10;

* Agrupa elementos de um mesmo tipo para um mesmo propósito.

→ estruturas de declarações sequenciais



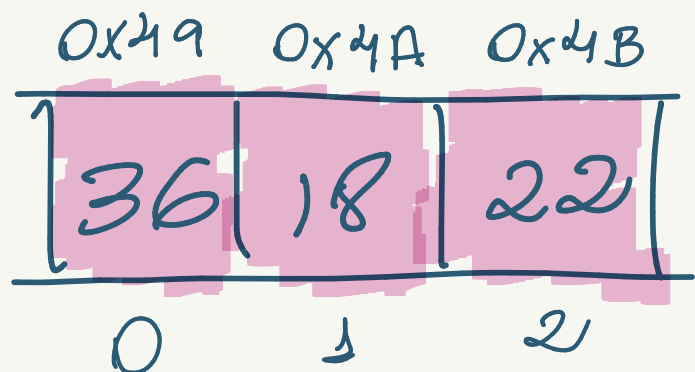
short idade [3];

+ variável

+ do tipo short

+ com 3 posições

endereço de idade: 0x49



$$\text{idade}[0] = 0x49 + 0 * 1$$

$$\text{idade}[1] = 0x49 + 1 * 1$$

$$\text{idade}[2] = 0x49 + 2 * 1$$