Pointers Intro

What is ?
working with memory
Declaration
initilization

Dreferance *
referance &
Adress Vs value

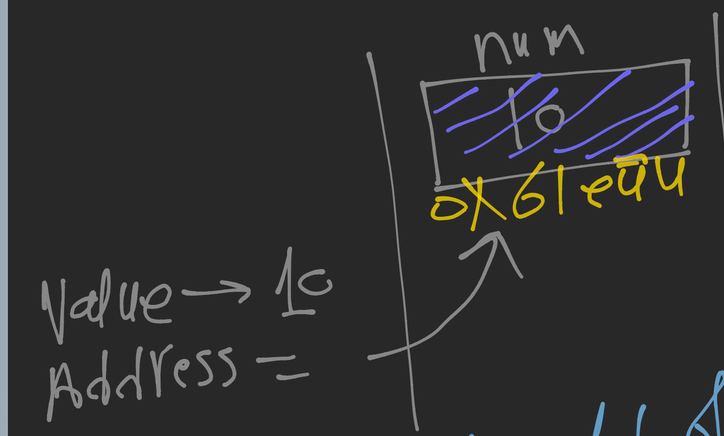
types — simple / structured / pointers

syntaxflexdec

Pointer is a variable that store the memer address of another variable

```
1  // @m_abrazeg
2
3  int num = 10;
4  cout << &num << "\n" ; // 0x61fea4
5
6  string str = "Koko";
7  cout << &str << "\n"; // 0x61fe8c
8
9  const double pi = 3.14;
10 cout << &pi << "\n"; // 0x8c7f88
11
12 cout << 0x61fea4 << "\n"; // 6422180
13 cout << 0x61fe8c << "\n"; // 6422156
14 cout << 0x8c7f88 << "\n"; // 9207688
```

int num = 10



Value → 10
Address =

(&) → get addr of any var.

Hexadecimal
0 — 9
A — f

T * name ;

4 byte

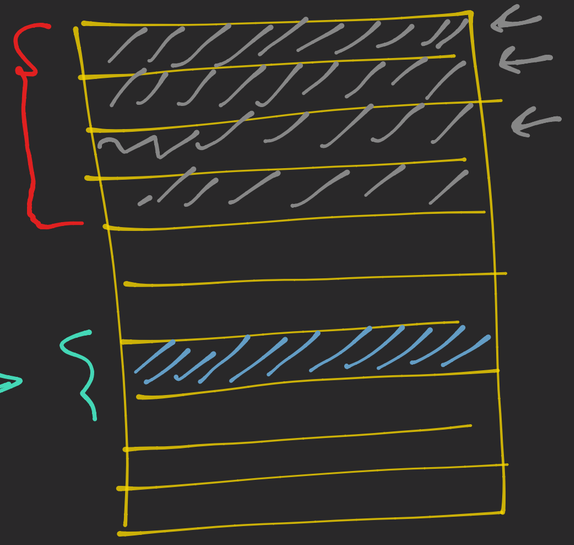
int x ;

1 byte

int *p = &x ;

bool b = true ;

bool *p2 = &b ;



```cpp
// How To Declare Pointers
int* ptr , N , *p;
string * str_ptr;

// Initialize Pointers [ NULL , address of variable ]
ptr = NULL;
str_ptr = 0;

bool isValid = true;
bool * bl_ptr = &isValid;

// How To Access Pointers
cout << &bl_ptr <<" "<< bl_ptr <<" "<< *bl_ptr << "\n";
// 0x61feb0 0x61feb7 1
```

pointer to integer

invalid memory

*b

bl_ptr

0x61feb7

pointer to

isValid

true

0x61feb7

value = true

Add = 0x61feb7

0x61feb6

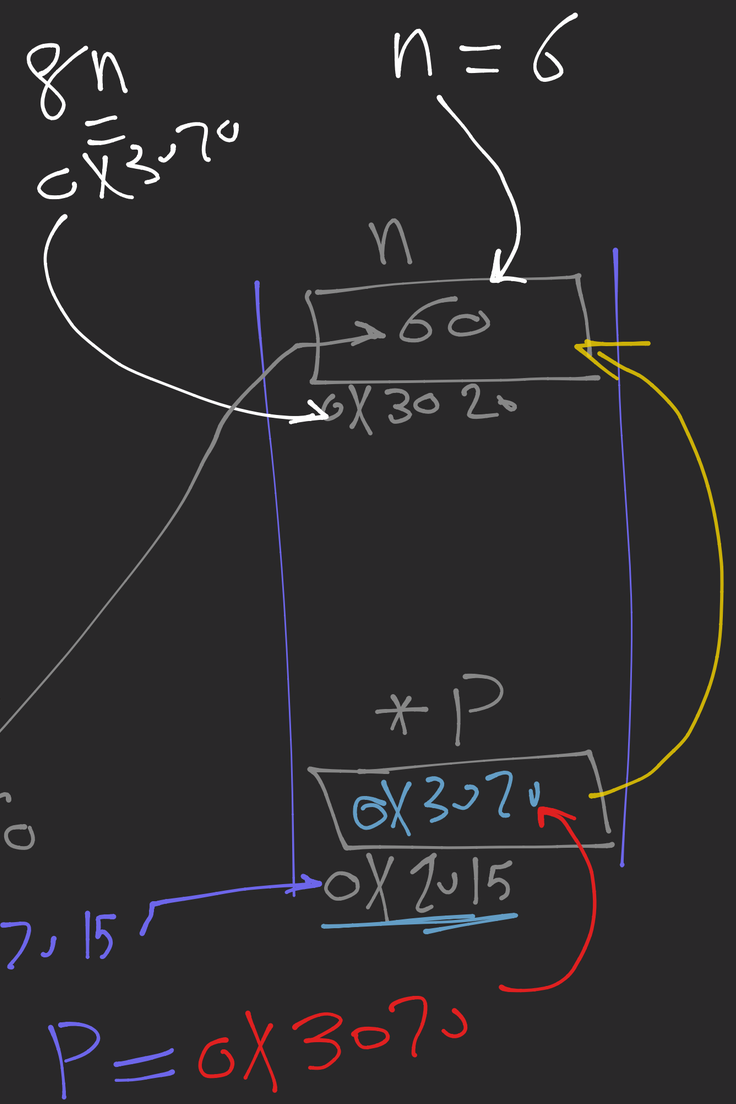
0x1feb7

*bl_ptr

0x61feb8

int n = 0

T * p = NULL, nullptr, 0



int n = 60

int *p = nullptr

P = &n ;

*P = 60

&P = 0x2b15

P = 0x3070

```cpp
// @m_abrazeg

// arrow operator ( -> )
string handle = "@m_abrazeg";
string* h = &handle;
cout << h << " " << &handle << "\n"; // 0x61fee4 0x61fee4

cout << h->at(0) << " " << handle[0] << "\n"; // @ @

cout << h->size() << " " << handle.size() << "\n"; // 10 10

cout << (*h).size() << endl; // 10
```

h.size() X

h→size();

(*h).size()