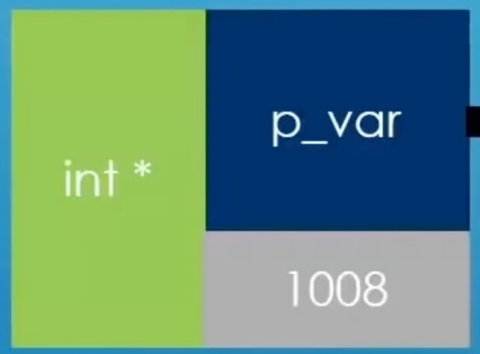
# Declaring and Using Pointers




| address | Variable name | value |
|---|---|---|
| 1000 | | |
| 1004 | | |
| 1008 | var | 22 |
| 1012 | | |
| 1016 | | |
| 1020 | | |
| 1024 | | |
| …. | | |

## Declaring Pointers

```
//Declaring pointers
int * p_number {};  // can only store an address of
                    //a variable of type int
double * p_fractional_number{}; // can only store an address of
                                //a variable of type double


//Explicitely initialize to nullptr
int * p_number1{nullptr};
int * p_fractional_number1 {nullptr};
```

## All pointer variables have the same size

```
std::cout << "Size of number pointer : " << sizeof(p_number)
          << ", size of int : " << sizeof(int) <<  std::endl;

std::cout << "Size of  fractional_number pointer : " << sizeof(p_fractional_number)
          <<", size of double : "  << sizeof(double) << std::endl;

std::cout << "Size of number1 pointer : " << sizeof(p_number1)
          << ", size of int : " << sizeof(int) <<  std::endl;

std::cout << "Size of  fractional_number1 pointer : " << sizeof(p_fractional_number1)
          <<", size of double : "  << sizeof(double) << std::endl;
```

## Position of the * doesn't matter

```
int*  p_number2{nullptr};
int * p_number3{nullptr};
int  *p_number4{nullptr};    // All work the same
                             // int *p_number4 format is easier
                             //to understand when you have multiple
                             //variables declared on the same line


int *p_number5{}, other_int_var{};
int* p_number6{}, other_int_var6{}; // Confusing as you wonder if other_int_var6
                                    //is also a pointer to int. It is not
                                    // The structure is exactly the same for the
                                    //previous statement

//It is better to separate these declarations on different lines though
int *p_number7{};
int other_int_var7{}; // No room for confusion this time
```

## Assigning data to pointer variables

```
//Initializing pointers and assigning them data

//We know that pointers store addresses of variables
int int_var {43};
int *p_int{&int_var};// The address of operator (&);

std::cout << "Int var : " << int_var << std::endl;
std::cout << "p_int (Address in memory)  : " << p_int << std::endl;


//You can also change the address stored in a pointer any time
int int_var1 {65};

int_var1 = 126;

p_int = &int_var1; // Assign a different address to the pointer
std::cout << "p_int (with different address)  : " << p_int << std::endl;
```

Pointer only stores the type for which it was declared

```
int *p_int1{nullptr};
double double_var{33};

//p_int1 = &double_var; // Compile error
```