(1) Today:

- Topics today:
  - The pointer variable
  - Function parameters
    - pass-by-value
    - pass-by-reference parameters (new to C++)
    - pass-by-pointer
    - pass-by-array

(2) The Pointer Variable

0x0fa059C

What is a pointer?

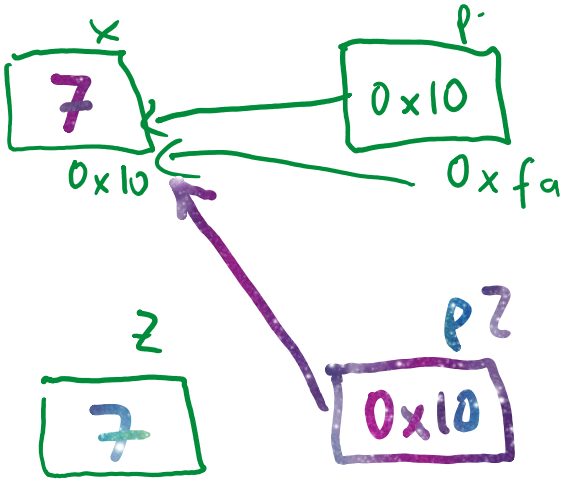
- A variable that stores a memory address
- We specify what data type is stored *at that address*
- * is used in declaration

```
int * p;
double * ptr;
char * p0;
```

var name
pointer type
type pointer points to

How do we get the address?
Using the address-of-operator: **&**

```
int x; // regular int variable
p = &x;
double y;
ptr = &y;
```



How to use a pointer to access the original variable?

```
*p = 7;
int z;
z = *p;
```

Now:

1)
```
cout << *p;   // 7
cout << x;    // 7
cout << z;    // 7
```

What about ?

2)
```
cout << p;    // 0x10
```

3)
```
cout << &p; // ? 0xfa
```

What if we want to have two pointers point to the same variable?

```
int *p2 = p;
```

(3) Pointer as function parameter

Why use a pointer?

- when we want to give a function access to a caller's local variable

Let's look at a function that sets an integer variable to zero.

```
1. // declaration: same rules as always
   void foo(int * x);

2. int main(){
3.     int y = 8;
4.     int *p;
   p   &y;
5.     foo(p);
6.     cout << y; // what is printed?
       return 0;
7. }

8. void foo(int * x){
9.     *x = 0;
10. }
```

(4) Function Parameter: pass-by-value

Pass-by-value

```
void myPBV(int x){
    x =-7; //
}
//
```

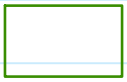
Example:

```
main:

int a = 2;
cout << a << ", ";
myPBV(a);
cout << a;
```

```
void myPBV(int x){
    x =-7;

}
```



Output:

$??

(5) Function Parameter: pass-by-reference

Pass-by-reference
e.g. function:

```
void myPBR(int &x){
   x =-7; // will it modify the caller's variable?
}
```

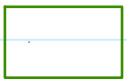
```
main():

int a = 2;
cout << a ;
myPBR(a);
cout << a;
```

Output:
$

```
void myPBR(int &x){
   x =-7;

}
```



(6) Function Parameter: pass-by-pointer

Pass-by-pointer

e.g. function:

```
void myPBP(int * x){
    *x =-7;
}
```

```
// cout << x; // ?
```

```
main()

int a = 2;
int * p = &a;
cout << a ;
myPBP(a);
myPBP(p);
cout << a;
```

console result:
$ 2 -7

```
void myPBP(int * x){
    *x =-7;
}
```

(7) Function Parameter: pass-by-array

Pass-by-array

e.g. function:

```
void myPBA(int arr[]){
   arr[0] = -7;
}
```

main:

```
int arr[] = {5,6};
cout << arr[0] ;
myPBA(arr);
cout << arr[0];
```

console result:

$

```
void myPBA(int arr[]){
   arr[0] = -7;
}
```

So what is an array, really?

(8) Arrays and pointers

```
int arr[] = {3,6};
int * p;
p = arr;

// allowed?
cout << p[0] << " " << p[1] << endl;


// allowed?
int * p1 = {7,8};
```