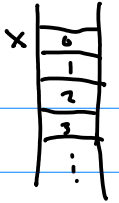
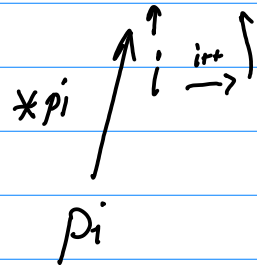
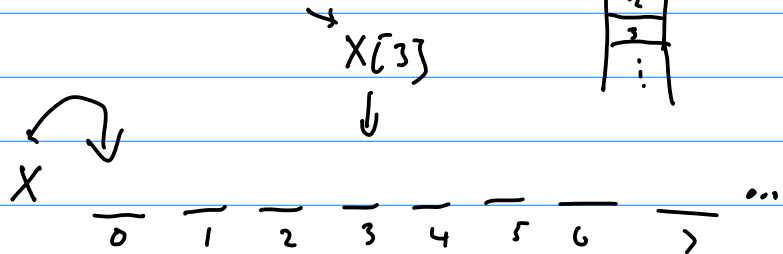


Pointers



```
int x[50];
```

```
i = 3
x[i]
x[i++] = 7
```



SAME

declaration

```
int *pi = x;
int *pi = &x[0];
```

best to initialize a pointer when you declare it

operator
↑
address of the thing to the right

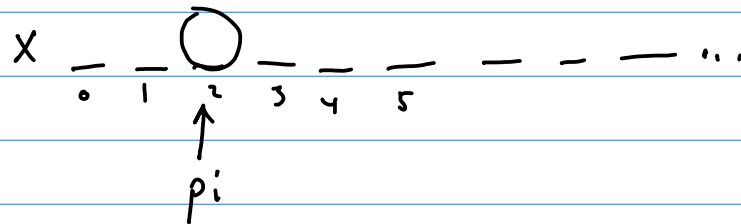
← Operator

operator



*pi dereferences the pointer

```
pi = &x[2]
```



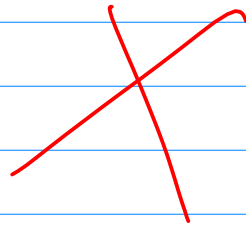
```
x[2] = 7;
*pi = 7;
```

← same

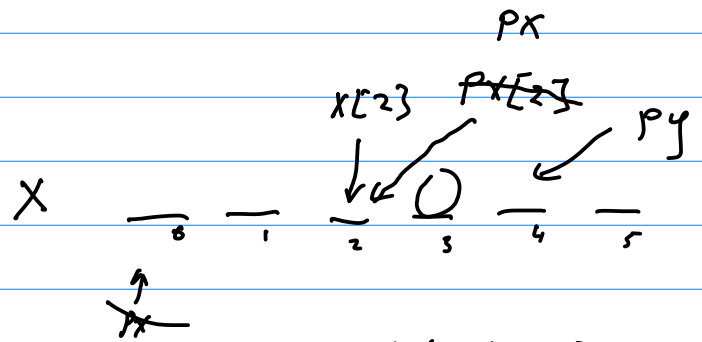
```
* & * & x[2] = 7;
```

function arguments

```
void swap (int x, int y) {
    int t = x;
    x = y;
    y = t;
}
```



```
void swap (int *px, int *py) {
    int t = *px;
    *px = *py;
    *py = t;
}
```



Operators $&$ address of
 $*$ dereference a pointer
 $[]$

```
int x[6];
int *px = x;
      = &x[0].
```

```
px = &x[2];
px[1]
```

① pointer plus int gives a point that is that many elements past the original
 $px++$

② pointer minus pointer \Rightarrow # elements between

```
int *py = px + 2
      py - px
      2
```

