Pointers

what are pointers?

- Up until now, we have only considered variables that represents values
  - A variable is a named box in memory that contains a value

what are pointers?

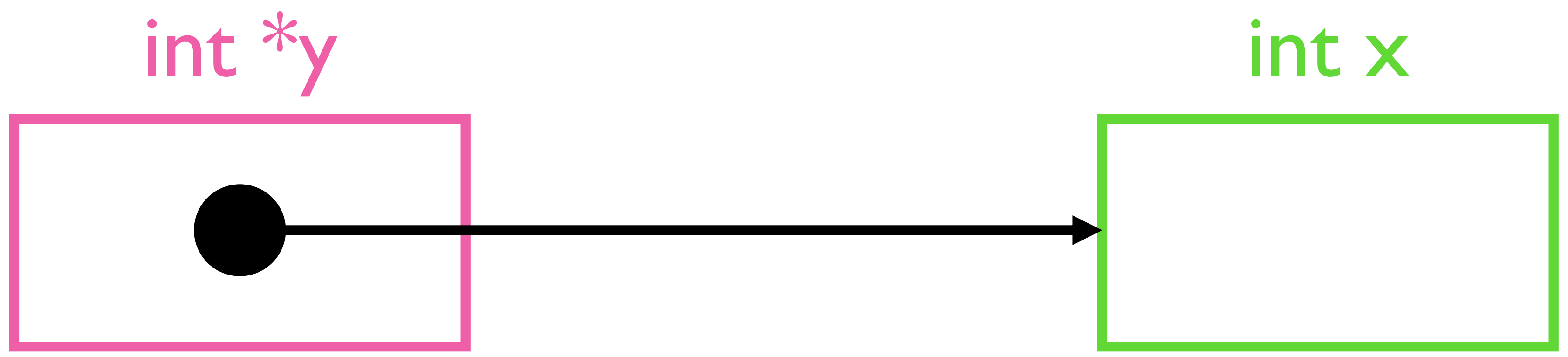
- Up until now, we have only considered variables that represents values
  - A variable is a named box in memory that contains a value

- But what if the box can contain the address of another box?

what are pointers?

- Up until now, we have only considered variables that represents values
  - A variable is a named box in memory that contains a value

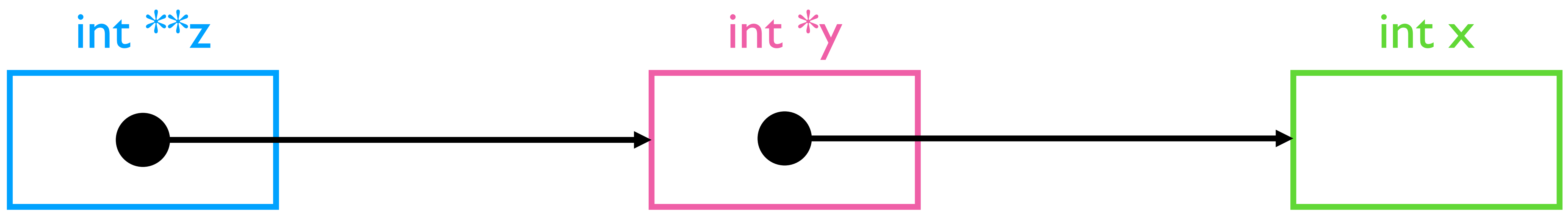
- But what if the box can contain the address of another box?



what are pointers?

- Up until now, we have only considered variables that represents values
  - A variable is a named box in memory that contains a value

- But what if the box can contain the address of another box?



pointers vs references

- A **pointer** is a variable that holds an *address*
- That address can be treated as a *value* that can be computed over

```
int * p = &a //p gets the address of a
int * q = p + 1 //q gets 4 + the address of a
```

- Some languages only have references instead of pointers
  - A reference refers to another memory location (under the hood: holds the address of another location in memory)
  - But cannot do pointer arithmetic

two key operations

- Two new unary operations:
- & : address-of operation
  - Returns the address of a variable

```
p = &a //store the address of a in p
```

- * : pointer dereference operation:
  - Let you load from, or store to, a pointer

```
* p = 7 //store to the address stored in p
x = * p //load from the address stored in p
```

pointer types

- How do we build pointers into our type system?
- Can thin of types as being defined by a grammar!

$$T \rightarrow \text{int} \mid \text{float}$$
$$T \rightarrow *\, T$$

- Type is either a **base type** or a **pointer to** another type

typing * and &

- What are the type rules for our two unary operations?

- * expr :

  If expr has type *T then * expr has type T

- & expr :

  If expr has type T then & expr has type *T next: code generation for pointers