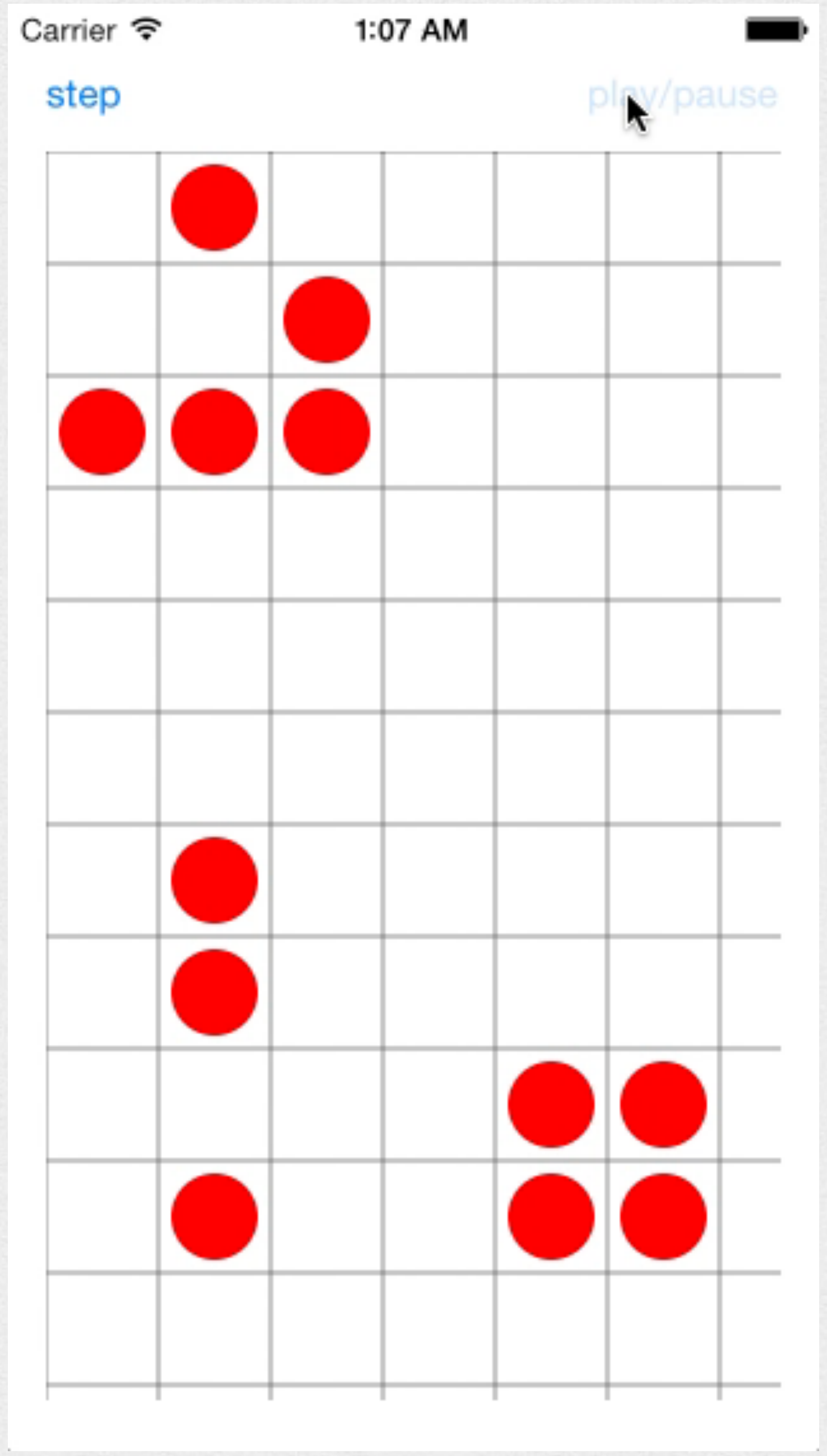
204: Swift Functional Programming

Part 1: Overview

raywenderlich.com

Conway's Game of Life



- Grid of live/dead cells
- Current liveness of cell and its neighbors determines future liveness

Agenda: Elements of FP style

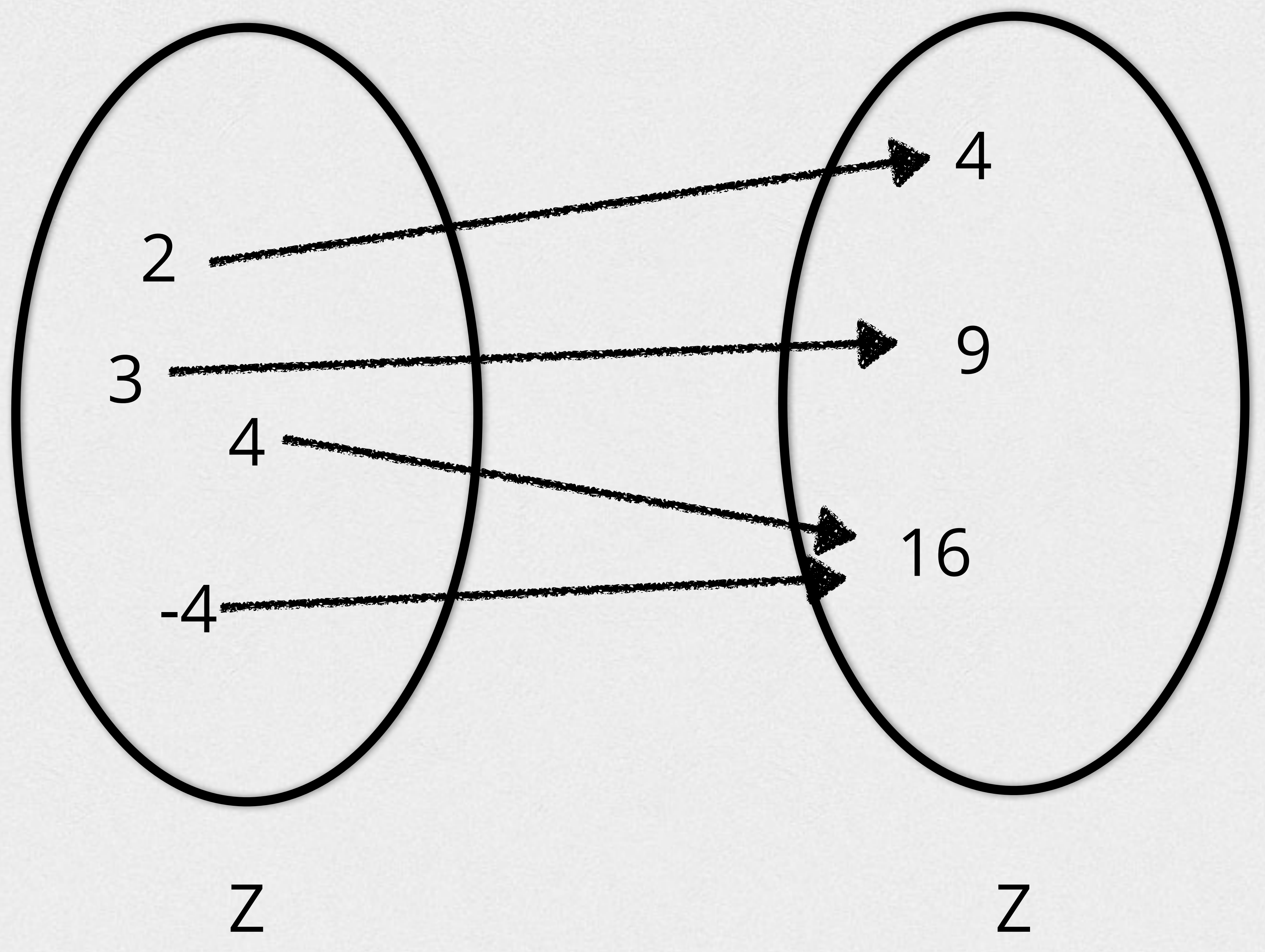
- higher-order functions (map, filter, reduce)
- pure functions
  - (input determines output)
- values
  - references < value types < immutable values

Functional Programming

- A *style* of programming ...
- ... treating the *mathematical function* as the primary unit of abstraction

Mathematical function

- maps values to values
- mathematical value
  - immutable
  - no identity, only equality



f: Z -> Z

f(3) = 9

f(x) = x * x

Values vs Variables

$$t_0 = 5$$

```
NSNumber * t0 = @5;
```

$$t_0 = 5$$

```
t0 = @6;
```

$$t_0 = 5$$

```
[t0 setIntegerValue:@7];
```

$$f(t_0) = 50$$

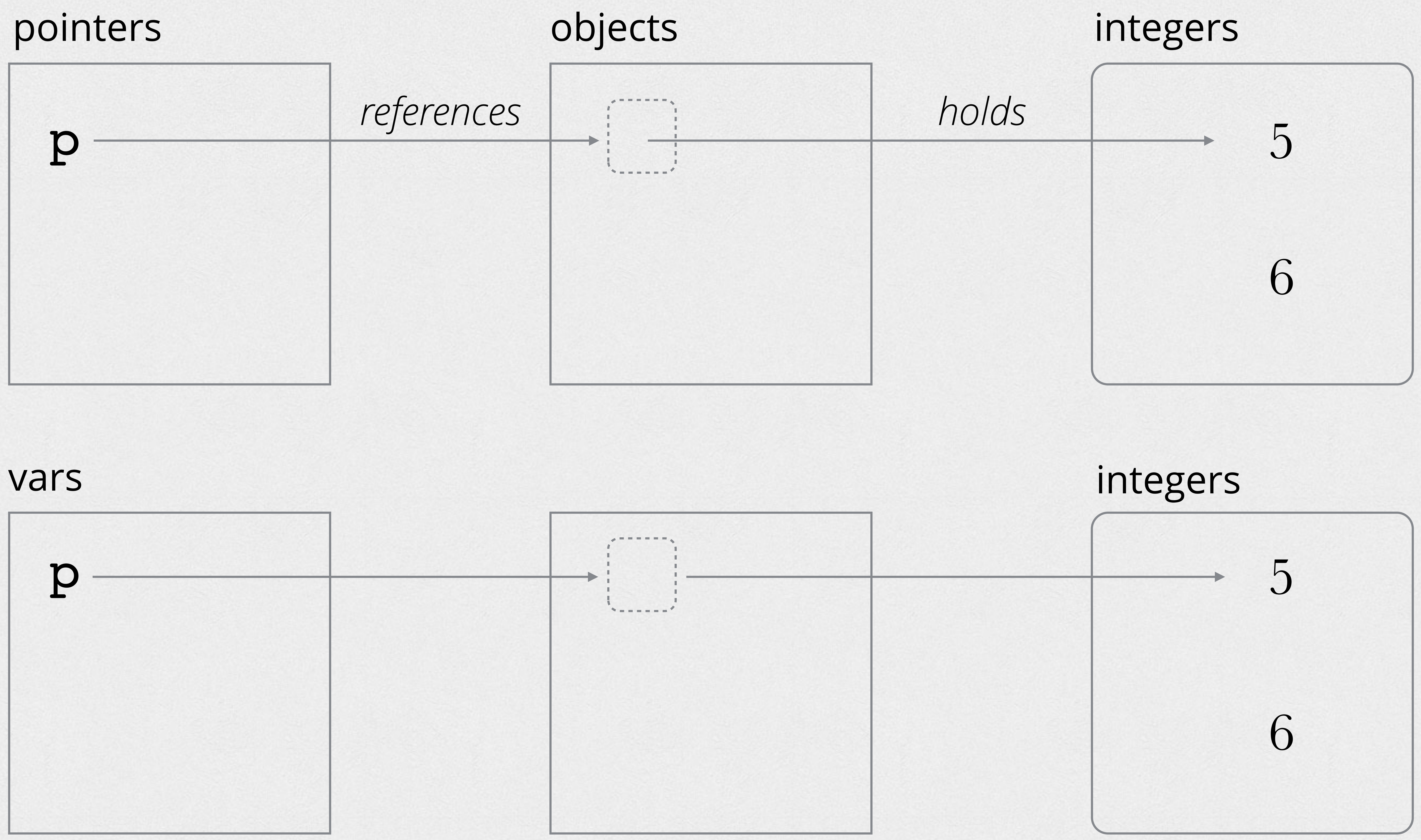
```
f(t0); // => 70
```

$$f(t_0) = 50$$

```
f(t0); // => 80
```

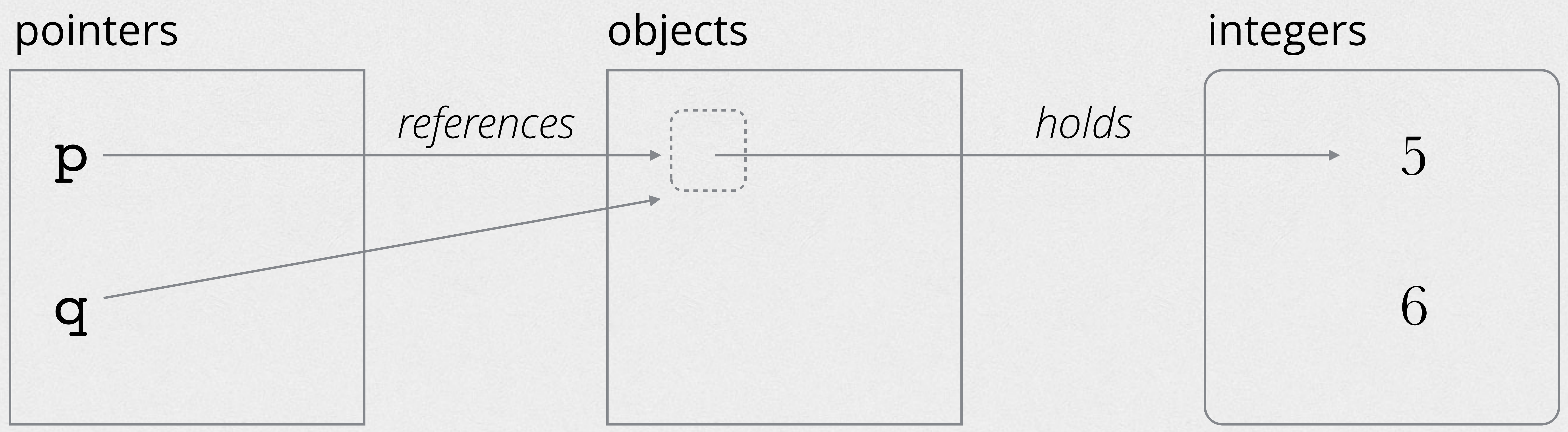
reference types
(i.e., classes)

value types
(i.e. structs)



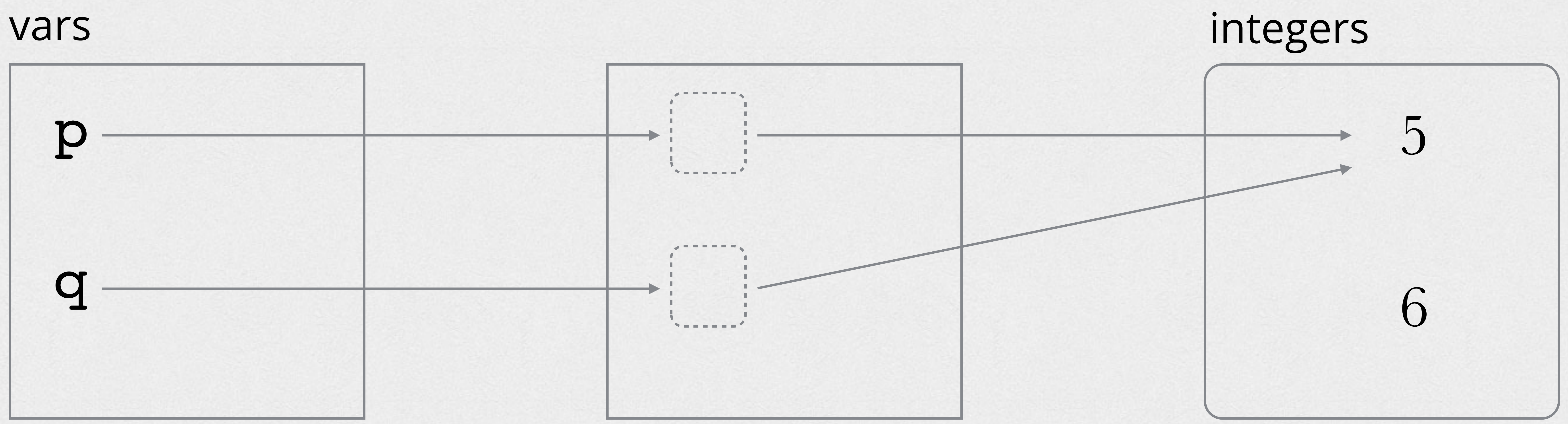
reference types
(i.e., classes)

```
q = p
```



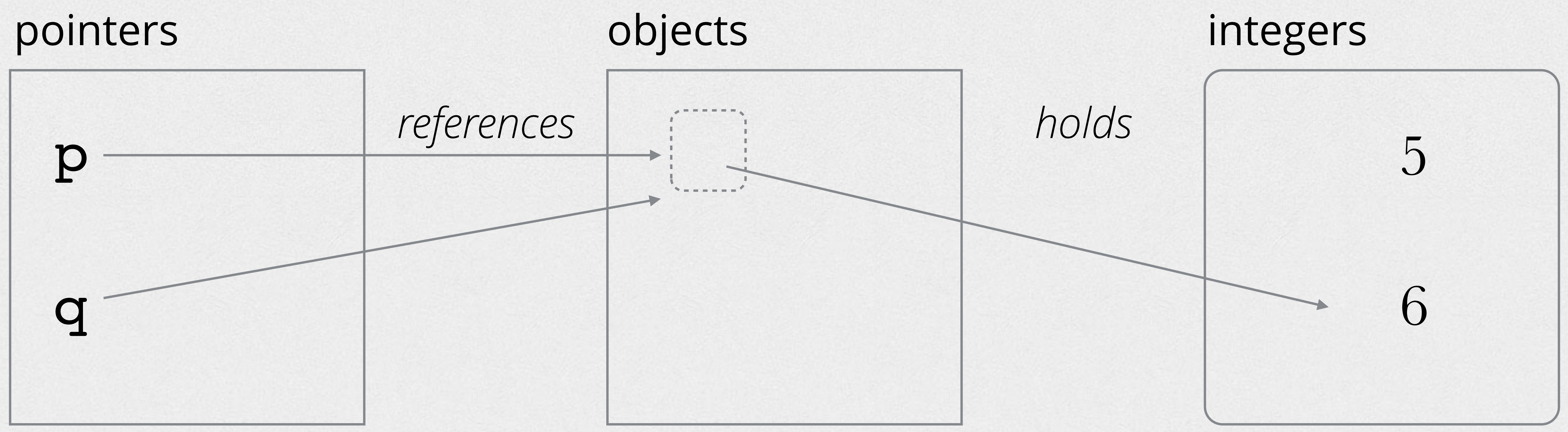
value types
(i.e. structs)

```
q = p
```



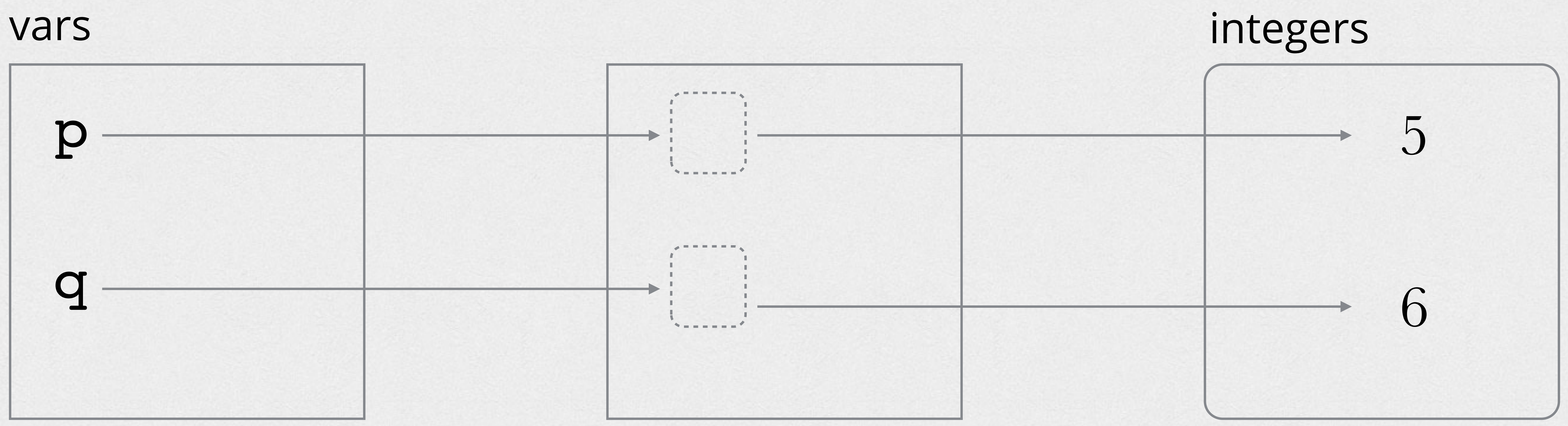
reference types
(i.e., classes)

```
q = p
q = 6
```



value types
(i.e. structs)

```
q = p
q = 6
```

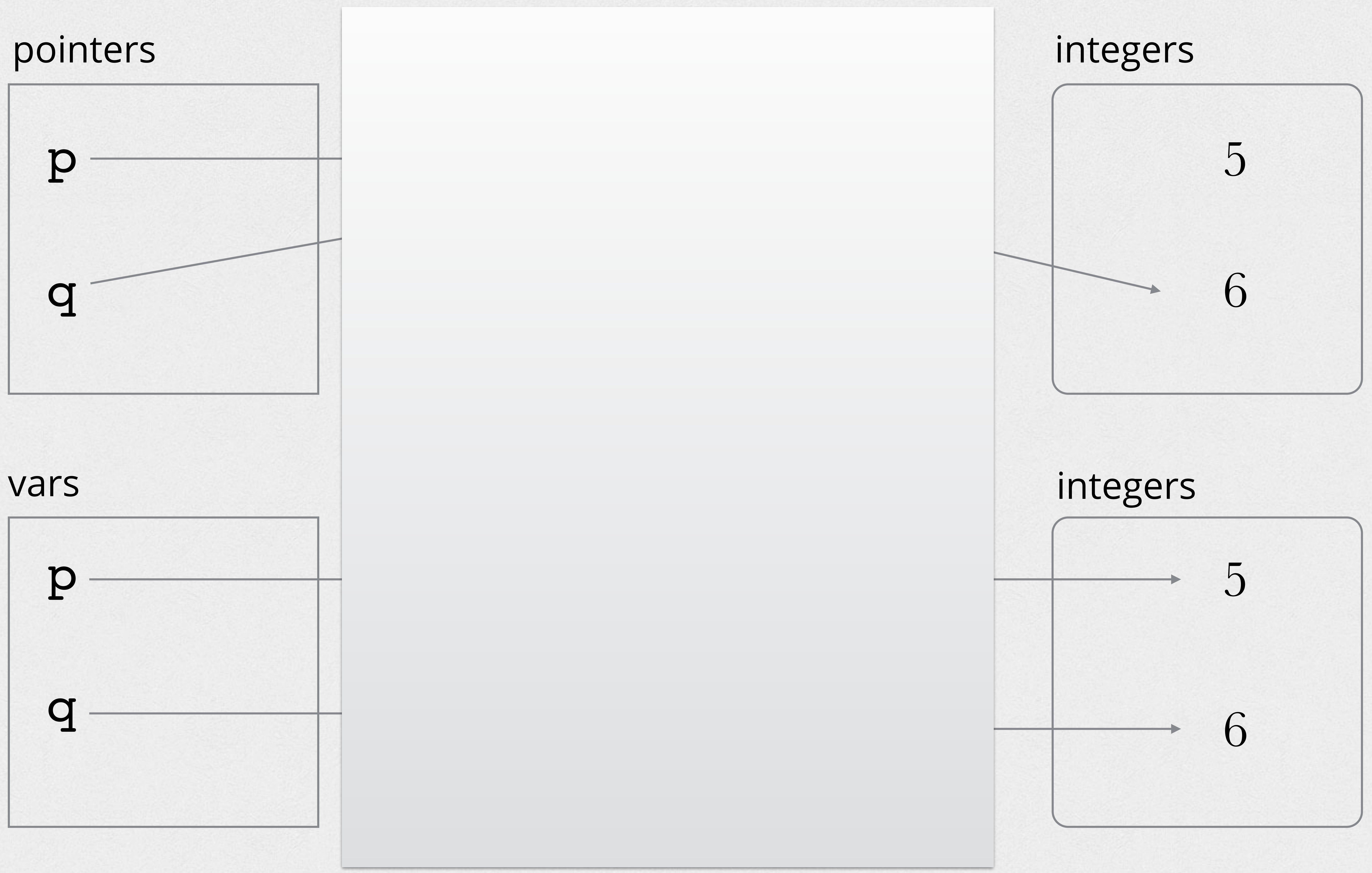


reference types
(i.e., classes)

```
q = p
q = 6
```

value types
(i.e. structs)

```
q = p
q = 6
```


pointers
p
q
integers
5
6
vars
p
q
integers
5
6