

# Communication Patterns on iOS

Advanced Topics in iOS & Swift



# Agenda

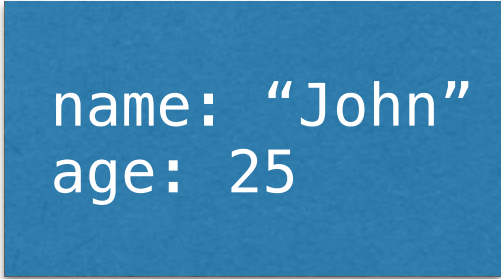
1. Objects & Messages
2. Tight vs Loose Coupling
3. One-To-One Communication
  1. Direct Method Invocation
  2. Delegates
  3. Closures
4. One-To-Many Communication
  1. NotificationCenter
  2. Key-Value-Observing

# Objects & Messages

```
class Person {  
    let name: String  
    var age: Int  
}
```

```
let person = Person(name: "John", age: 25)
```

person



```
name: "John"  
age: 25
```

# Objects & Messages

```
class Person {  
    let name: String  
    var age: Int  
  
    func hasBirthday() {  
        age = age + 1  
    }  
}
```

person

hasBirthday()

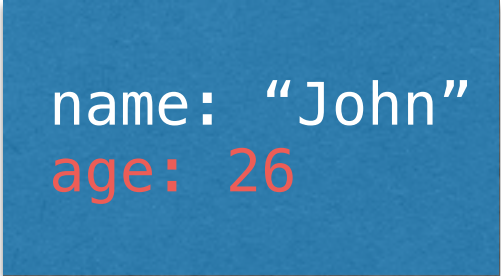


name: "John"  
age: 25

# Objects & Messages

```
class Person {  
    let name: String  
    var age: Int  
  
    func hasBirthday() {  
        age = age + 1  
    }  
}
```

person



name: "John"  
age: 26

# Objects & Messages

```
class PersonViewController: UIViewController {  
    let person = Person(name: "John", age: 25)  
  
    func updatePersonAge() {  
        person.hasBirthday()  
    }  
}
```

viewController



hasBirthday()



person



# Tight vs Loose Coupling

Q: How much do the sender of a message and the receiver of that message know about each other?

**Loose:** Sender and receiver don't know anything about each other

**Tight:** Sender and receiver know everything about each other

**The less the sender and the receiver know about each other the better!**

One-To-One

One-To-Many

Loose Coupling

???

???

Tight Coupling

???

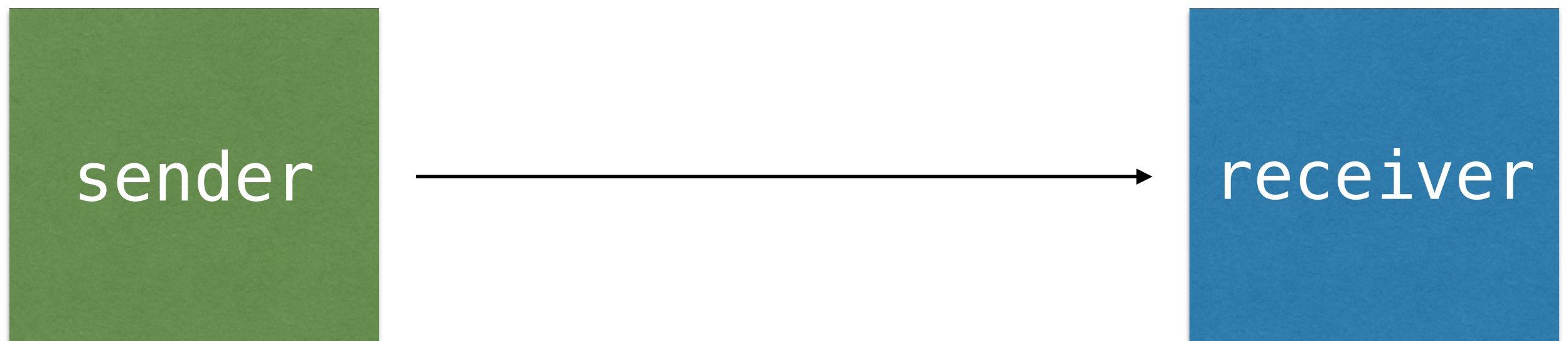
???



# One-To-One

Exactly **2 objects** involved (one **sender**, one **receiver**)

**Sending message = Calling method**



One-To-One

One-To-Many

Loose Coupling

???

???

Tight Coupling

Direct method  
calls

???

One-To-One

One-To-Many

Loose Coupling

Delegates

**???**

Tight Coupling

Direct method  
calls

**???**

One-To-One

One-To-Many

Loose Coupling

Delegates  
&  
Closures

**???**

Tight Coupling

Direct method  
calls

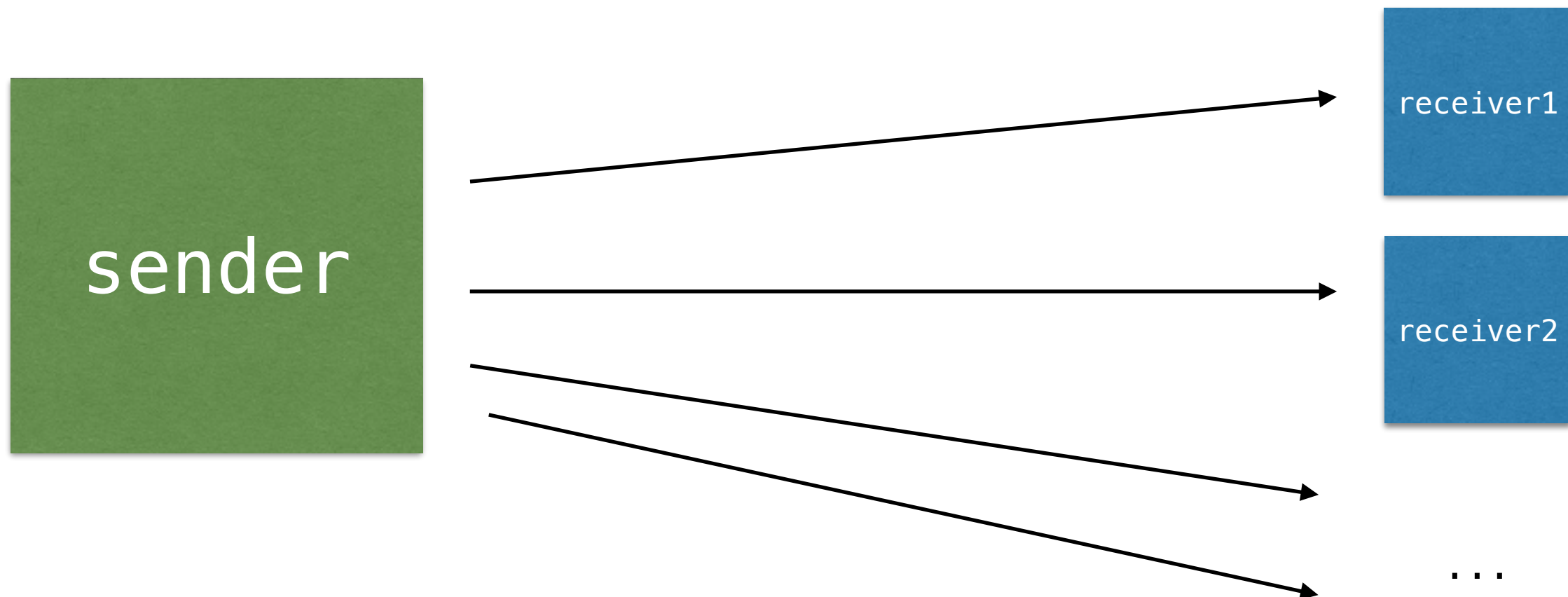
**???**

# One-To-Many

**At least 1 object** involved (one **sender**, 0..n **receivers**)

**Sending message = Posting notification**

Receivers must **subscribe** to receive the notification



One-To-One

One-To-Many

Loose Coupling

Delegates  
&  
Closures

NotificationCenter

Tight Coupling

Direct method  
calls

**???**

One-To-One

One-To-Many

Loose Coupling

Delegates  
&  
Closures

NotificationCenter

Tight Coupling

Direct method  
calls

Key-Value-Observing  
(KVO)

# Tight vs Loose Coupling

NSNotificationCenter  
Center

Delegates

KVO

Direct method  
invocation

Closures



Loose

Tight