

Closure

“Closures are self-contained blocks of functionality that can be passed around and used in your code.”

Meaning that a closure is a block of code you can assign to any variable or even pass it around your code to another function to be run at a later point. A closure parameters are not named. You only specify the type. Then inside the closure's expression you can then assign the different parameters to a name. You can also skip the variable naming in the expression and use the short hand “\$0” - meaning the first parameter in the closure. Then if you have more you would just increment the number by one. This is not a good solution because it could make the code harder to understand. You can have closures that returns nothing (void) or any object.

```
let birthday:(String) -> () = { name in
    print("Happy birthday, \(name)!")
}

birthday("Arthur")
```

Here is a closure that take a string as it's first parameter - returns nothing “()” and names the first parameter received “name” - it then just prints out a string.

Closures is really smart for handling all sorts of things. It's mostly used for stuff like error logging or handling lengthy task, like download files with completion handlers.

Weak / Strong

Apple has a mostly automated way of handling memory on the iPhone, and it's quite effective. It's called ARC (Automatic Reference Counting) - in it's basic form it's quite simple. It keeps track on how many times and where a object is reference. And if it's ever hit 0 it's deleted automatically. This is why in most use cases it's best to stick with weak references. As you get more memory freed up automatically if something is not in use anymore by the application. If you use a Strong link. You can end up with the count never going down and therefor always have a small chunk of memory allocated to something that is not really in use by the application or user. But it's only really recommended to references that can at some point become nil without also crashing the application. So you need to find the sweet middle road.