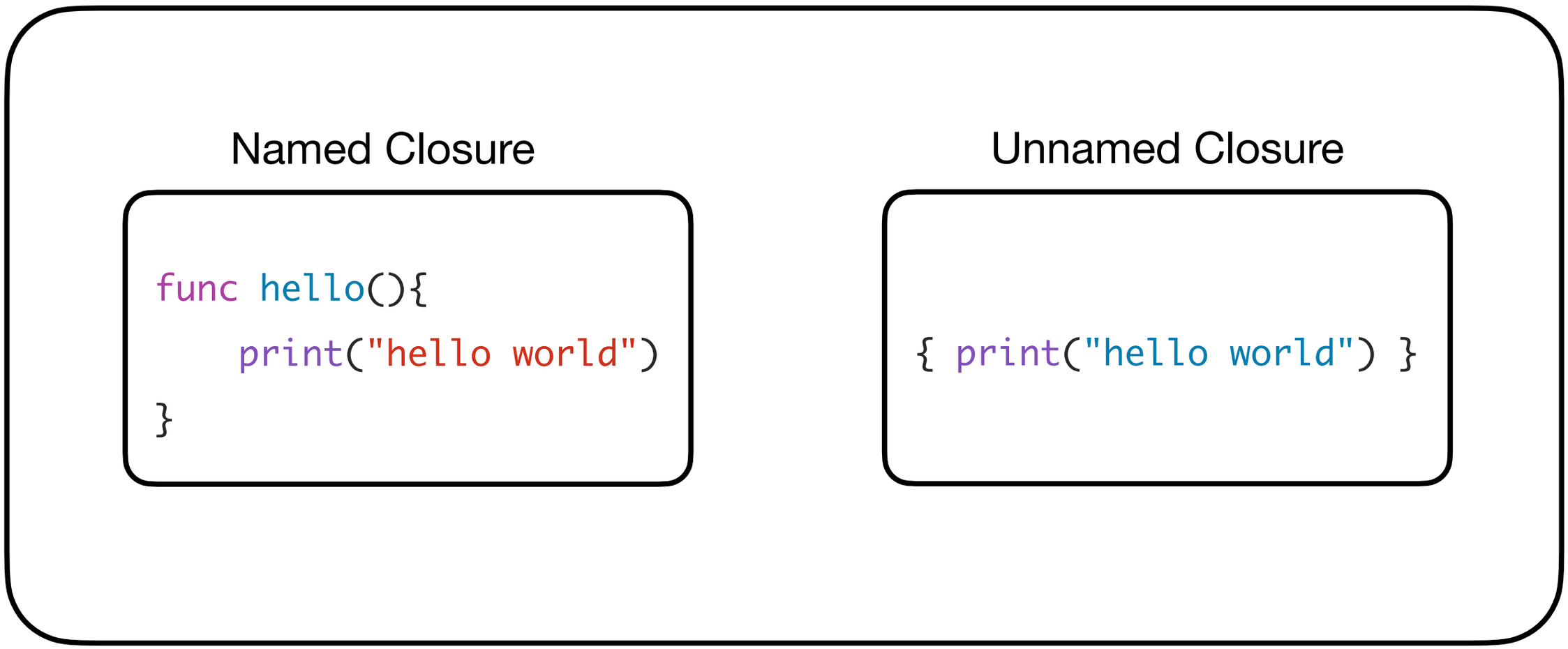
Swift Closure

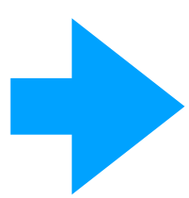
Closure


Closure
Named Closure
func hello(){
print("hello world")
}
Unnamed Closure
{ print("hello world") }

기본 표현 문법

```
{ (parameters) -> return type in
    statements
}
```

```
func sayHello(name:String){
    print("Hello \(name)")
}
```



```
{ (name:String)->Void in
    print("Hello \(name)")
}
```

```
func doSomething(name:String, handler:(String)->Void){
    handler(name)
}
doSomething(name: "홍길동", handler: sayHello)

doSomething(name: "홍길동", handler: {
   (name:String) -> Void in print("Hello \(name)")
})
```

타입 추론에 의한 생략

```
let names = ["Chris", "Alex", "Ewa", "Barry", "Daniella"]

func backward(_ s1: String, _ s2: String) -> Bool {
    return s1 > s2
}

var reversedNames = names.sorted(by: backward)

var reversedNames1 = names.sorted(by: {(s1: String, s2: String) -> Bool in
                        return s1 > s2 })

var reversedNames2 = names.sorted(by: $0 > $1 )

var reversedNames3 = names.sorted(by: > )
```

후행 Closure

```
doSomething(name: "홍길동", handler: { (name:String) -> Void in
    print("Hello \(name)")
})

doSomething(name: "홍길동"){ (name:String) -> Void in
    print("Hello \(name)")
}
```

```
func exec(handler:()->()){
    handler()
}

exec{
    print("Hello")
}
```

Escaping Closure

함수의 매개변수로 전달된 closure
함수의 외부나 함수의 종료 후에 실행될때는 @escaping 이라고 명시해야함
일반적으로 비동기로 실행될때 사용

```
func exec(handler:@escaping ()->())->(()->()){
    return handler
}

let func1 = exec{
    print("Hello")
}

func1()
```

Capturing Values

함수 내부의 값을 Closure에서 사용했을때 함수는 종료되었지만
count값은 소멸되지 않고 캡쳐되어 사용가능한 상태가 됨

```
func captureValue()->(()->()){
    var count = 3
    print(count)
    let closure = {
        count += 1
        print(count)
    }
    return closure
}

let func1 = captureValue()
func1()
```