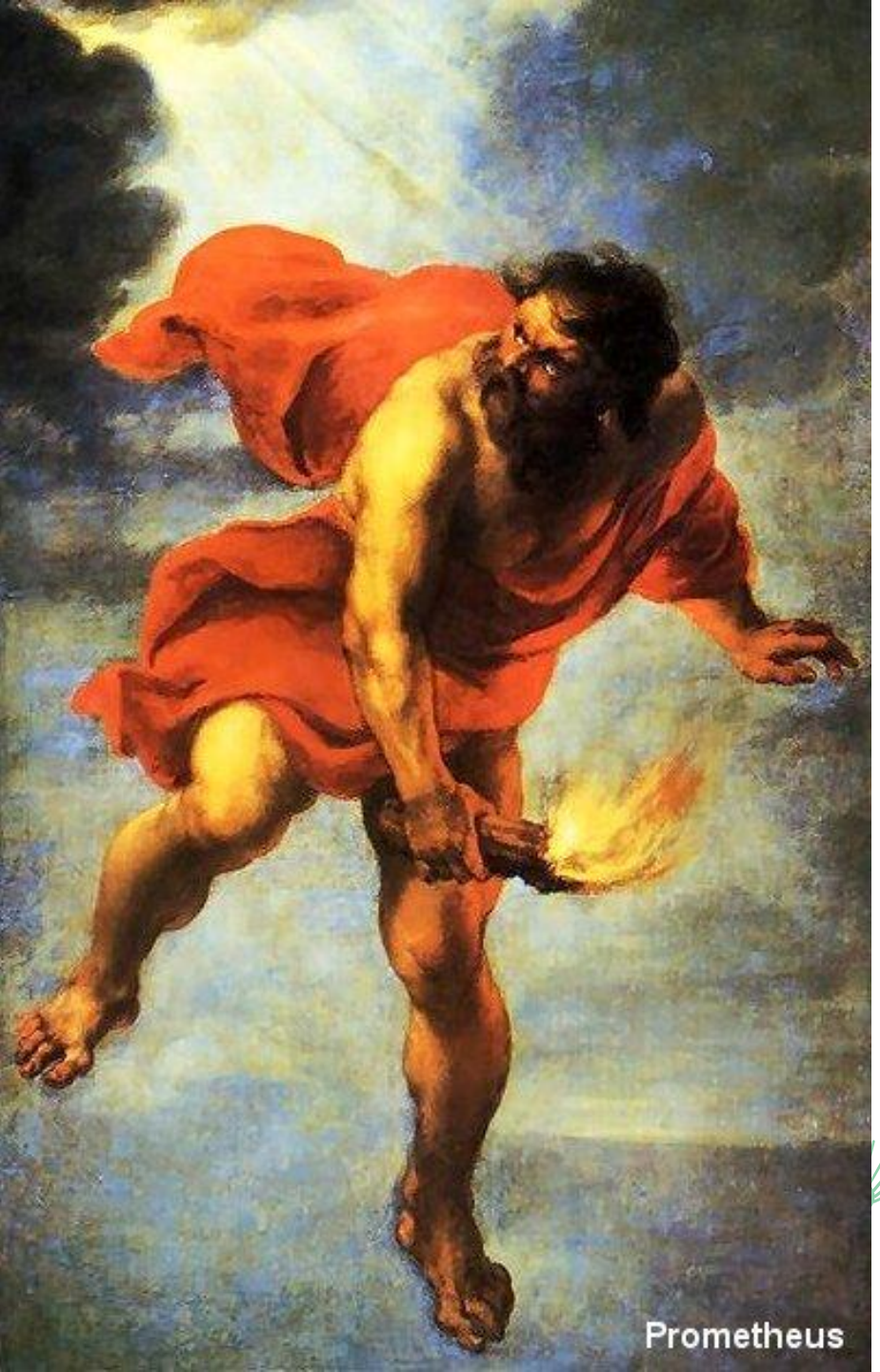
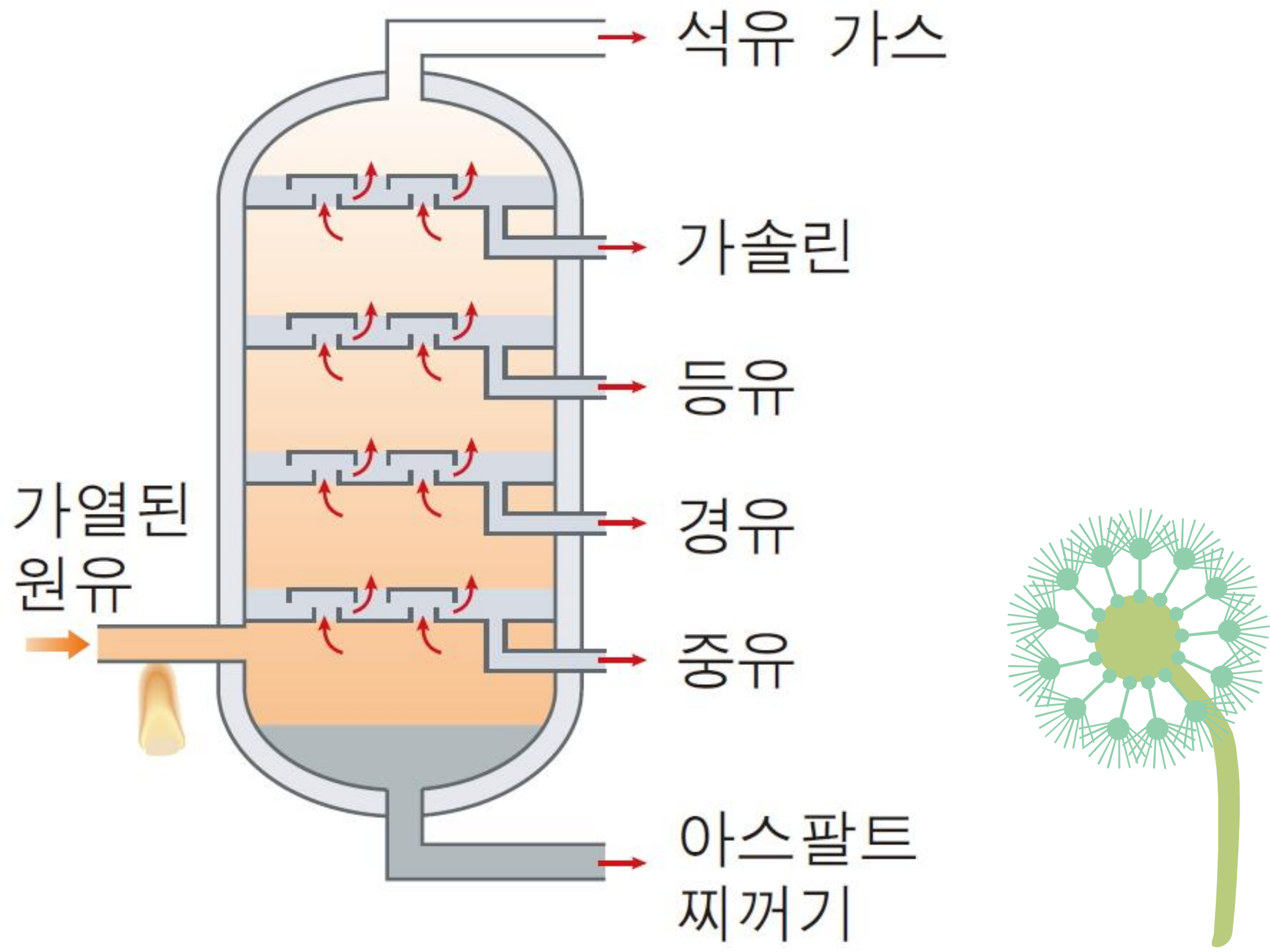
Go lang 하나부터 백만까지

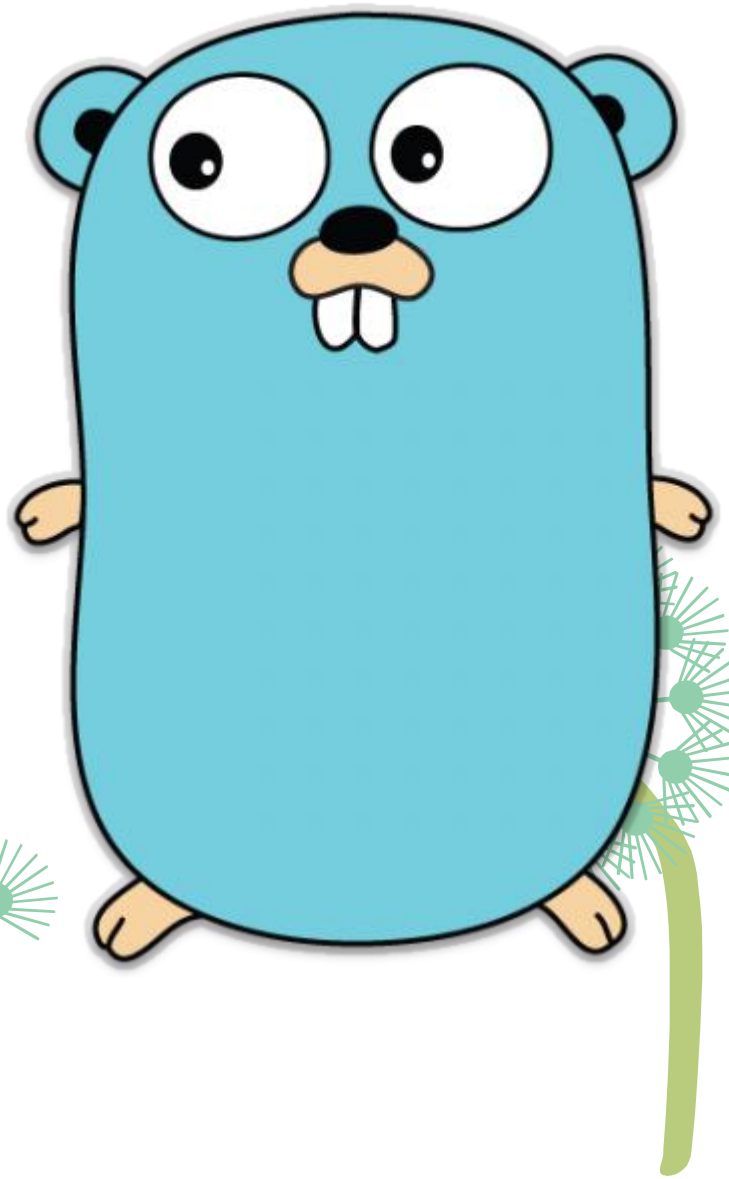
112121 배진수


Prometheus


석유 가스
가솔린
등유
경유
중유
아스팔트
찌꺼기
가열된
원유

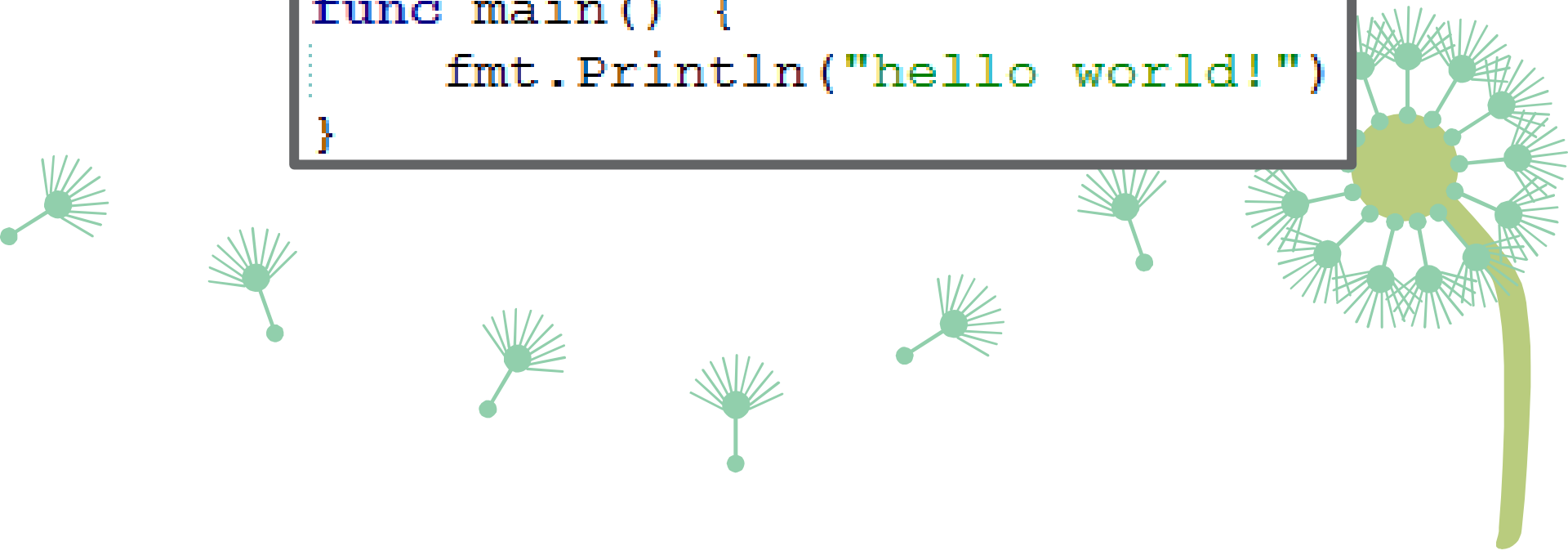
Go 언어?

- Made by Google
- Open source
- Simple
- Reliable
- Efficent
- General Purpose lang



Hello world!

```
package main

import (
    "fmt"
)

func main() {
    fmt.Println("hello world!")
}
```



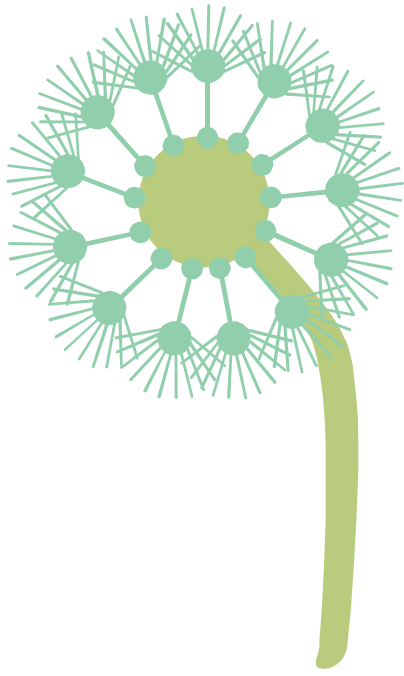
변수 선언법

- var name type

```
package main

import "fmt"

func main() {
    var a int
    a=1
    b:=1
}
```

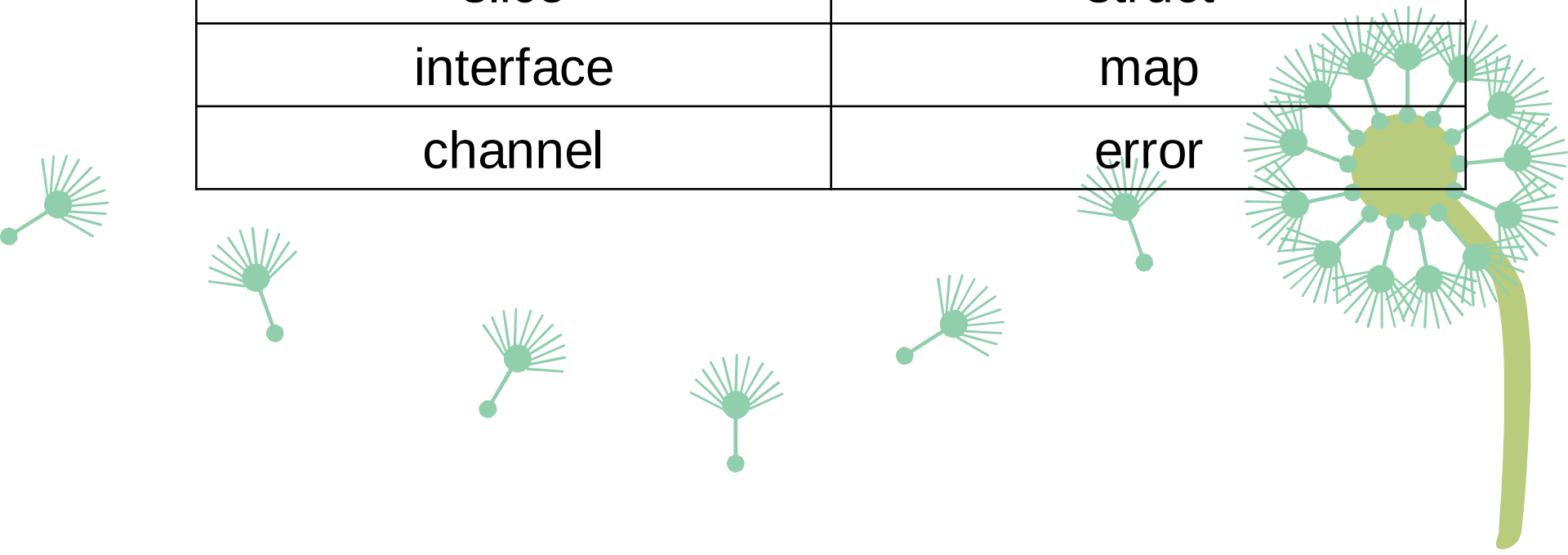






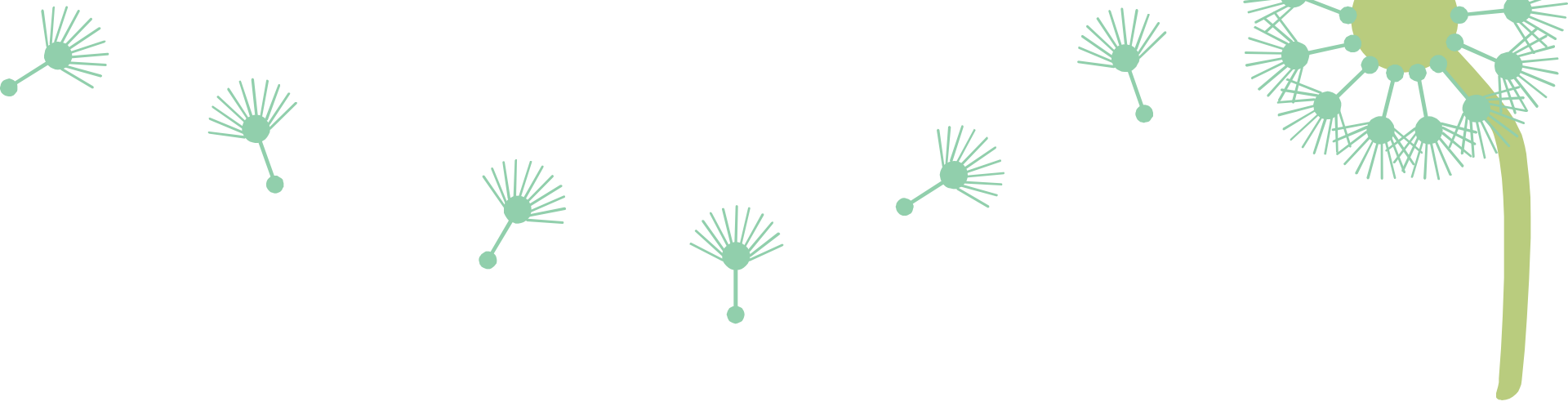
Data type

Bool	Numeric
Array	Pointer
string	function
slice	struct
interface	map
channel	error



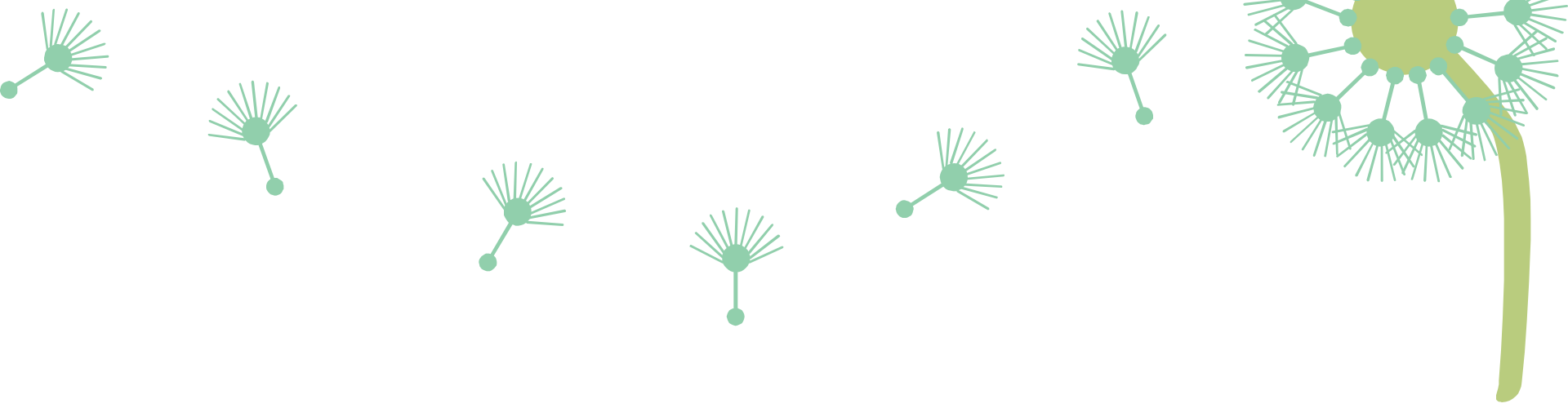
=, == 연산자

- 대입, 비교 연산자
- Array
- string
- struct
- Slice,map



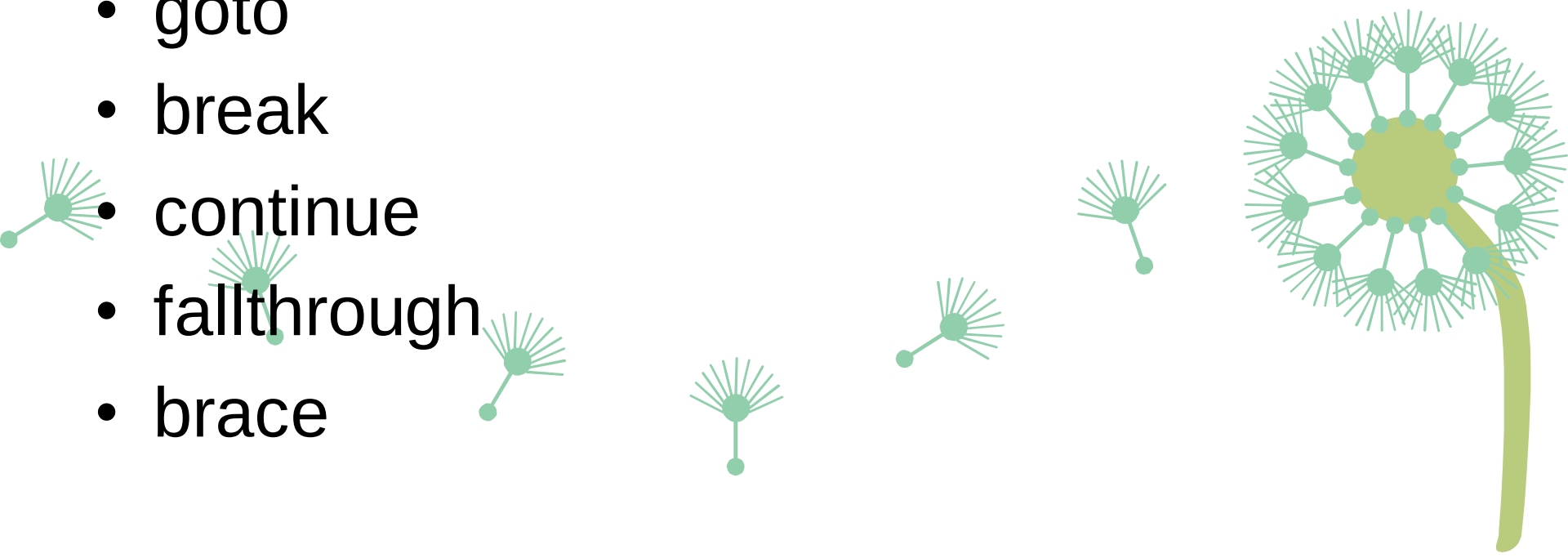
=, == 연산자

- 대입, 비교 연산자
- Array
- string
- struct
- ~~Slice,map~~



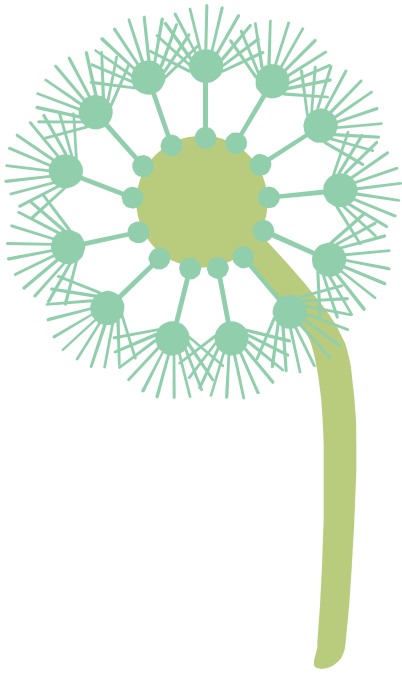
제어문

- for
- If-else
- switch-case
- goto
- break
- continue
- fallthrough
- brace



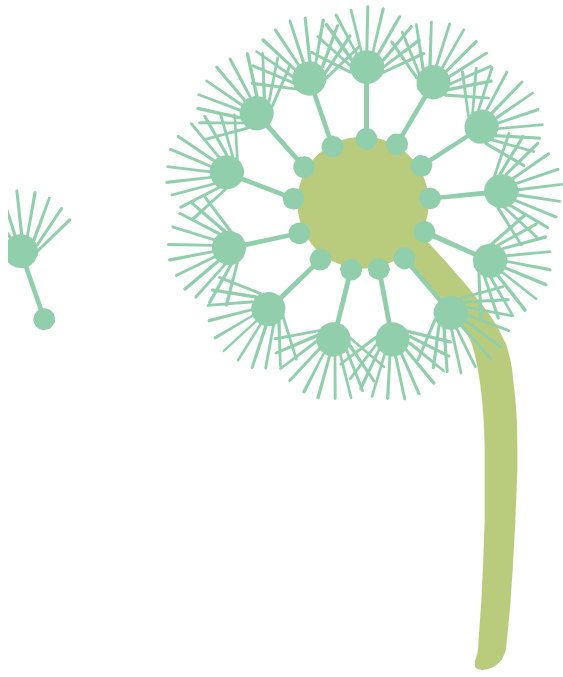
for

```
for i := 0; i < 5; i++ {
    fmt.Println(i)
}

i := 0
for i < 5 {
    fmt.Println(i)
    i = i + 1 // i++
}

for {
    fmt.Println("Hello, world!")
}
```



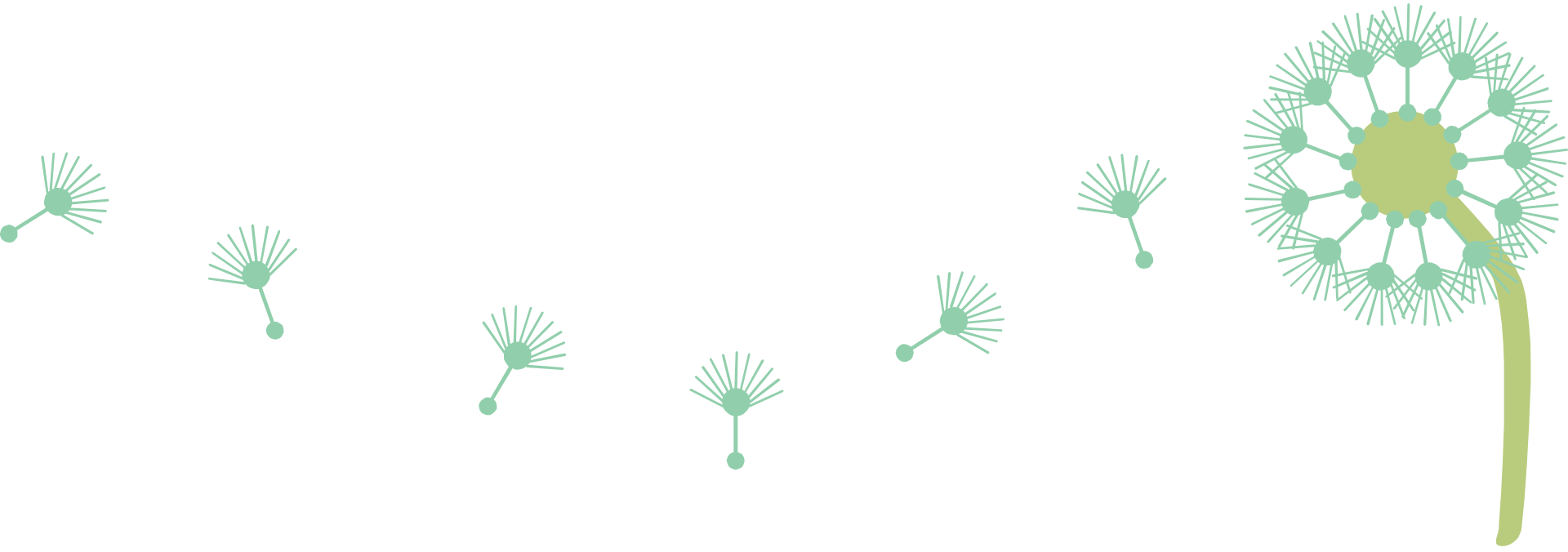
switch-case

```
s:="hello"
switch(s){//가장 기본적인 switch 형태
    case "hello":
        fmt.Println(s);fallthrough
    case "world":
        fmt.Println(s)
}

switch(s){//여러 조건식을 한번에 처리
    case "hello", "world":
        fmt.Println(s)
}

switch {//case문에서 논리식을 지원
    case s=="hello" || s=="world":
        fmt.Println(s)
}
```



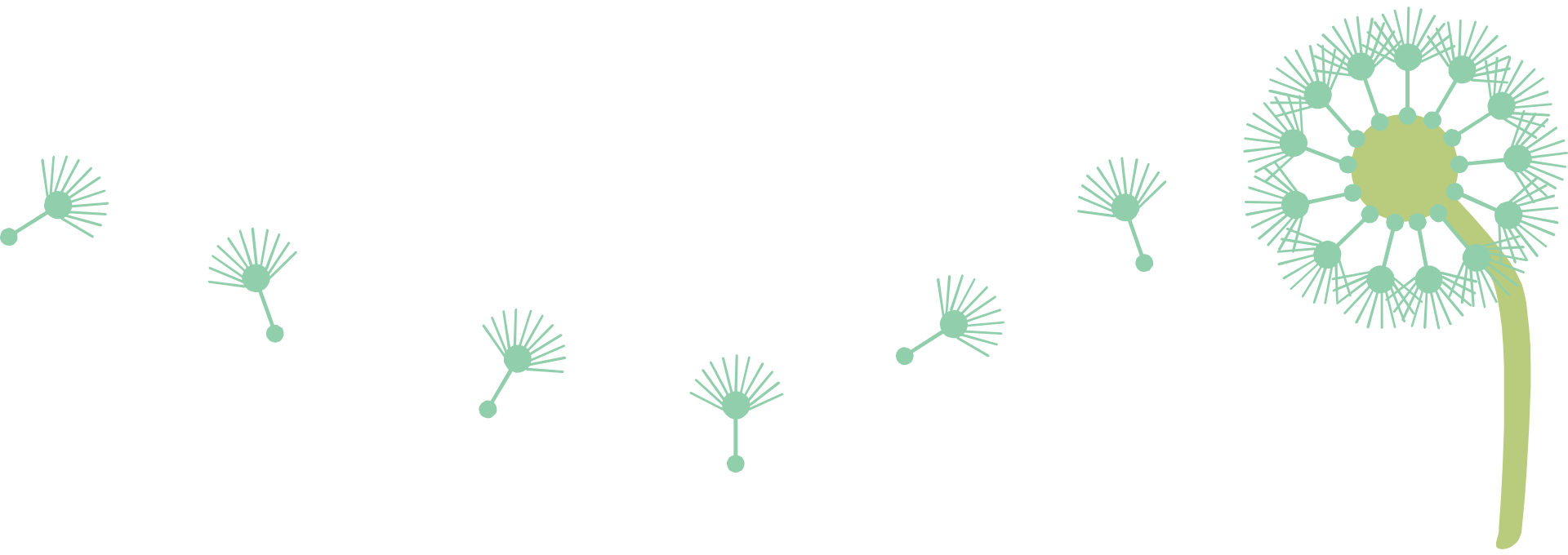
지원하지 않는 제어구조

- try catch 블록구조
- #define ... 등 proprocessor



지원하지 않는 제어구조

- try catch 블록구조->함수 형태로 지원
- #define ... 등 proprocessor

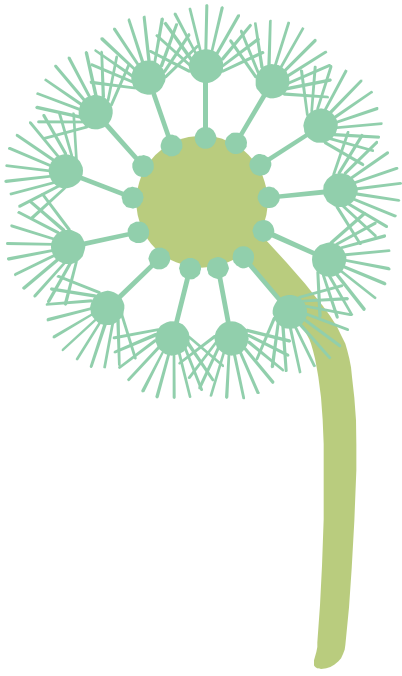


Function and method

- Multiple return values
- defer
- go
- lambda
- Closure
- Interface
- Struct embedding

Multiple return values

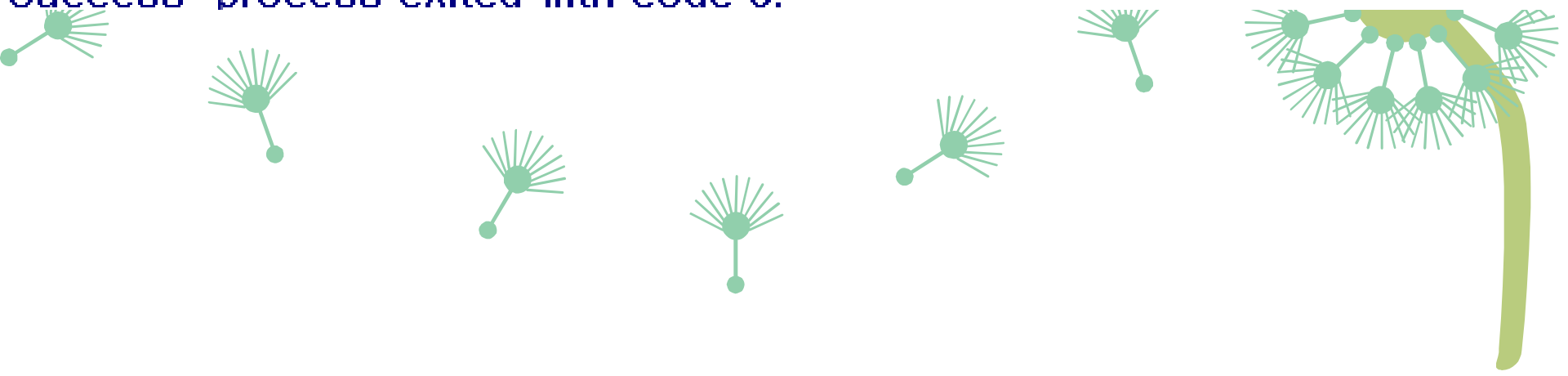
```
package main

import "fmt"

func swap(a int,b int) (int, int) {
    return b, a
}

func main() {

    a:=1
    b:=2
    a,b=swap(a,b)
    fmt.Println(a,b)
}
```



defer

```
func main() {
    defer fmt.Println("world")
    fmt.Println("hello")
    fmt.Println("hello")
}
```

```
C:/Go/bin/go.exe build -i [C:/Users/b/Documents/Go/test]
Success: process exited with code 0.
C:/Users/b/Documents/Go/test/test.exe  [C:/Users/b/Documents/Go/test]
hello
hello
world
Success: process exited with code 0.
```



go

```
func hello() {
    for i:=0;i<3;i++ {
        fmt.Println("hello")
        time.Sleep(1*time.Second)
    }
}
func main() {
    go hello()
    for i:=0;i<3;i++ {
        fmt.Println("world")
        time.Sleep(1*time.Second)
    }
}
```

```
C:/Go/bin/go.exe build -i [C:/Users/b/Documents/Go/test]
Success: process exited with code 0.
C:/Users/b/Documents/Go/test/test.exe  [C:/Users/b/Documents/Go/test]
world
hello
hello
world
world
hello
Success: process exited with code 0.
```

Lambda & Closure

```
//                ↓ 리턴 값이 익명 함수
func calc() func(x int) int {
    a, b := 3, 5            // 지역 변수는 함수가 끝나면 소멸되지만
    return func(x int) int {
        return a*x + b // 클로저이므로 함수를 호출 할 때마다 변수 a와 b의 값을 사용할 수 있음
    }
    // ↑ 익명 함수를 리턴
}

func main() {
    f := calc() // calc 함수를 실행하여 리턴값으로 나온 클로저를 변수에 저장

    fmt.Println(f(1)) // 8
    fmt.Println(f(2)) // 11
    fmt.Println(f(3)) // 14
    fmt.Println(f(4)) // 17
    fmt.Println(f(5)) // 20
}
```

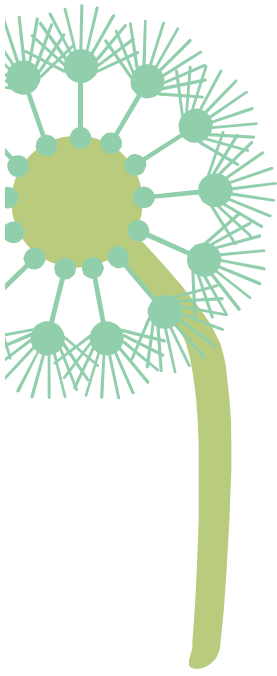
```
C:/Go/bin/go.exe build -i [C:/Users/b/Documents/Go/test]
Success: process exited with code 0.
C:/Users/b/Documents/Go/test/test.exe  [C:/Users/b/Documents/Go/test]
8
11
14
17
20
Success: process exited with code 0.
```

Struct embedding

```
type pikachu struct { // 피카츄 구조체 정의
    name string
    age  int
}
func (p *pikachu) thunderbolt() {
    fmt.Println("백만 볼트!")
}
type poketmon_traniner struct {//트레이너 구조체
    pikapika pikachu        // has-a 관계
    school string
    grade  int
}
func main() {
    var traniner poketmon_traniner
    traniner.pikapika.thunderbolt()
```

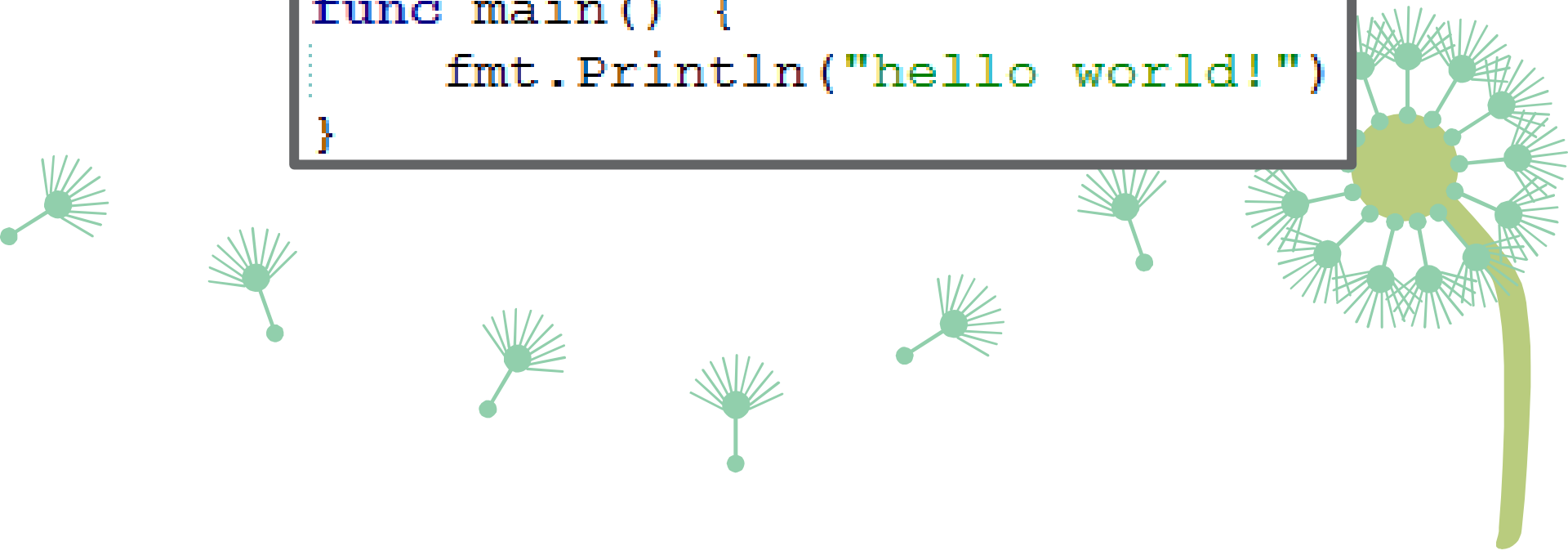
Struct embedding

```
type pikachu struct { // 피카츄 구조체 정의
    name string
    age  int
}
func (p *pikachu) thunderbolt() {
    fmt.Println("백만 볼트!")
}
type raichu struct {//라이츄 구조체
    pikachu         // is-a 관계
    school string
    grade  int
}
func main() {
    var rai raichu
    rai.thunderbolt()
```



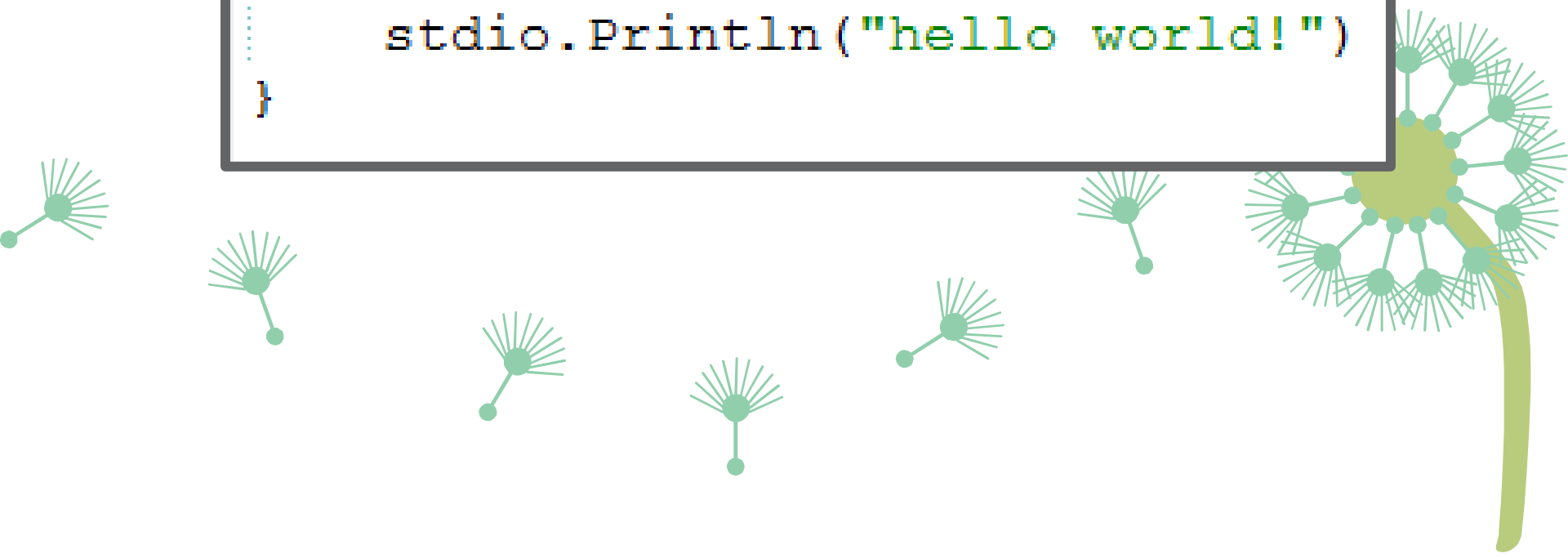
Package

```
package main

import (
    "fmt"
)

func main() {
    fmt.Println("hello world!")
}
```



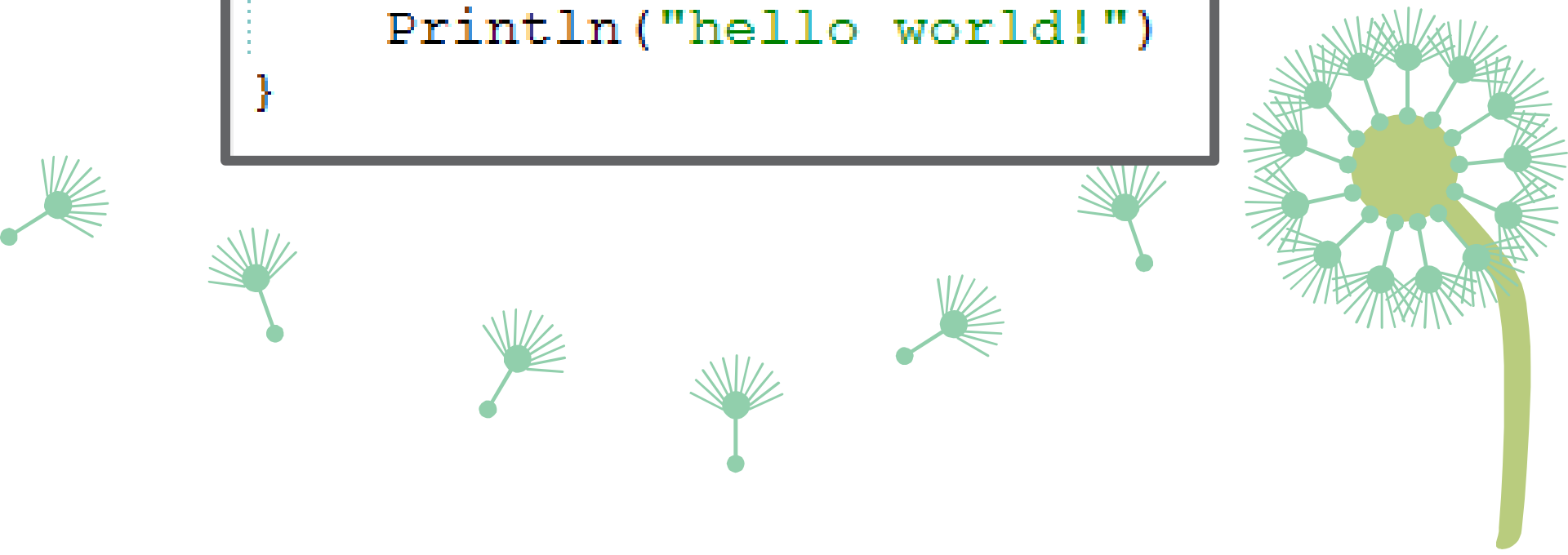
Package

```
package main

import  stdio "fmt"


func main() {
    stdio.Println("hello world!")
}
```



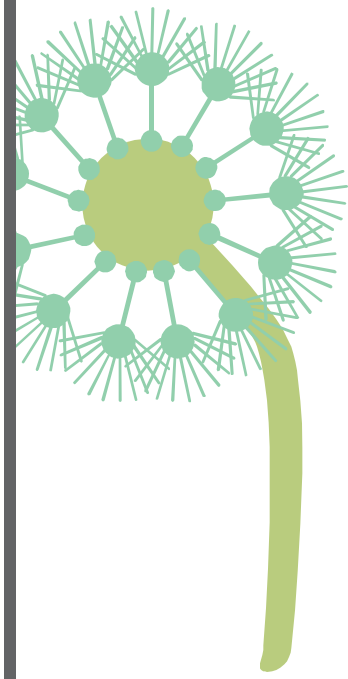
Package

```
package main

import ."fmt"


func main() {
    Println("hello world!")
}
```



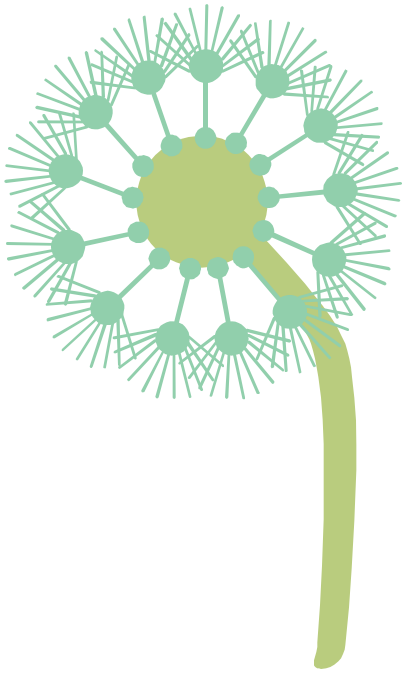
Package

```
package db

import (
    "fmt"
    "os"
    "github.com/ziutek/mymysql/mysql"
    _ "github.com/ziutek/mymysql/thrsafe"
    "MyHttp/Header_date"
    "MyHttp/Msg_Encoding"
)

//db 객체 선언
var DB mysql.Conn

//db 스키마 구조체
type User struct {
    Id string
    Pw string
    Mail string
    Number string//아이디마다 할당되는 고유 번호
}
```



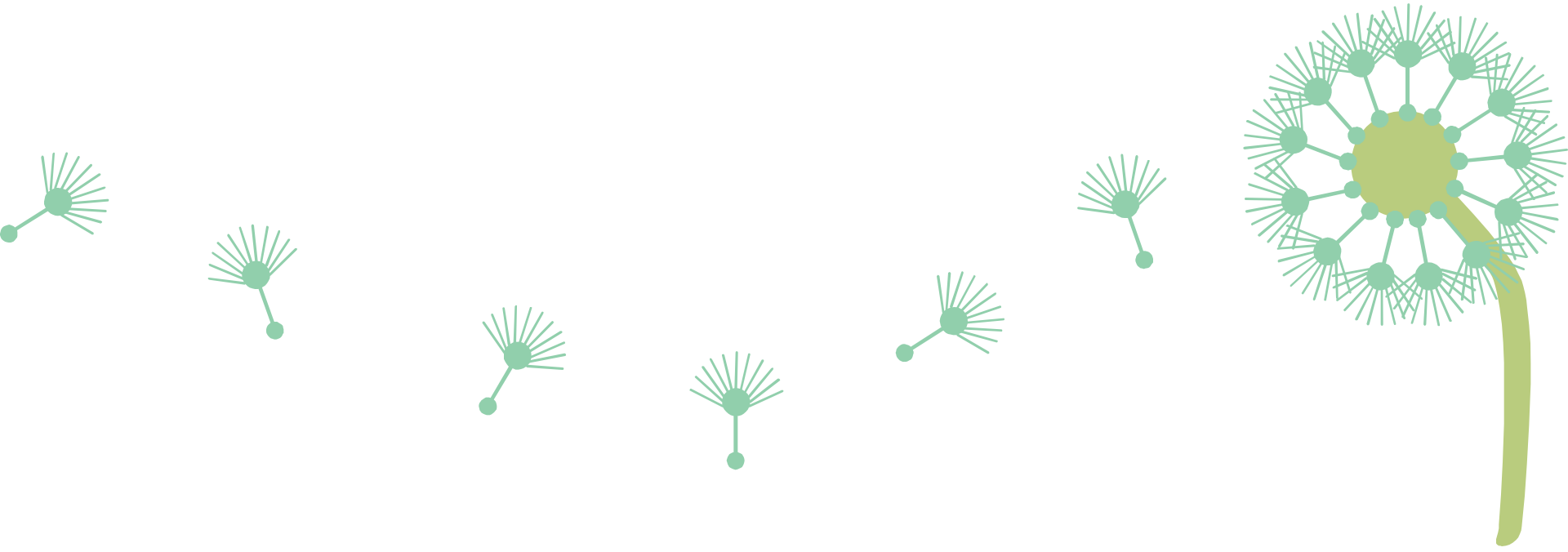
Package

```
func fetch_one_user_test(){
    var dat User
    dat.Id="test5"
    Fetch_one_user(&dat)
}

func fetch_one_page_test(){
    var dat Page
    dat.Addr="testpage6"
    Fetch_one_page(&dat)

}
```



Others

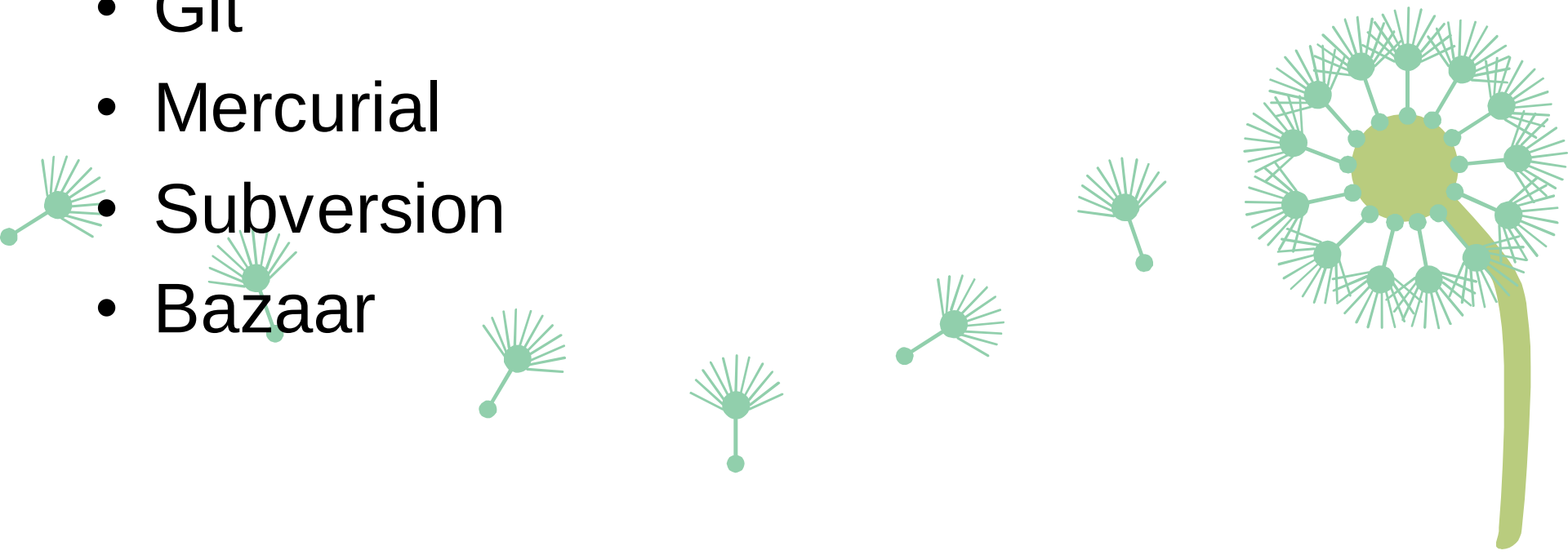
- install package from web storage
- Garbage collection



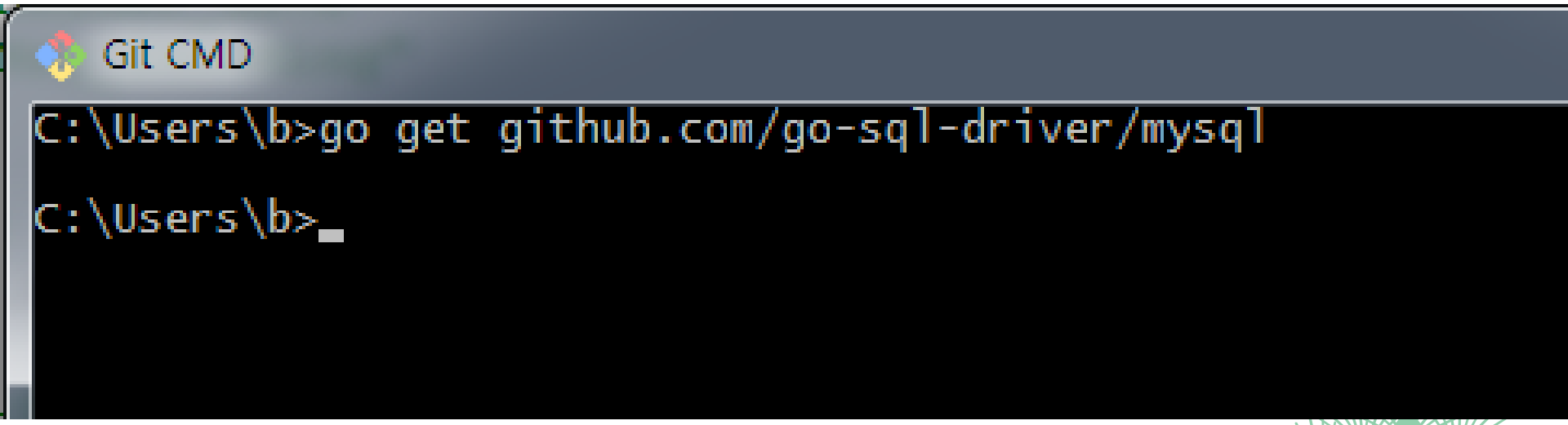
install package from web storage

```
import (
    "fmt"
    "os"
    "github.com/ziutek/mymysql/mysql"
```

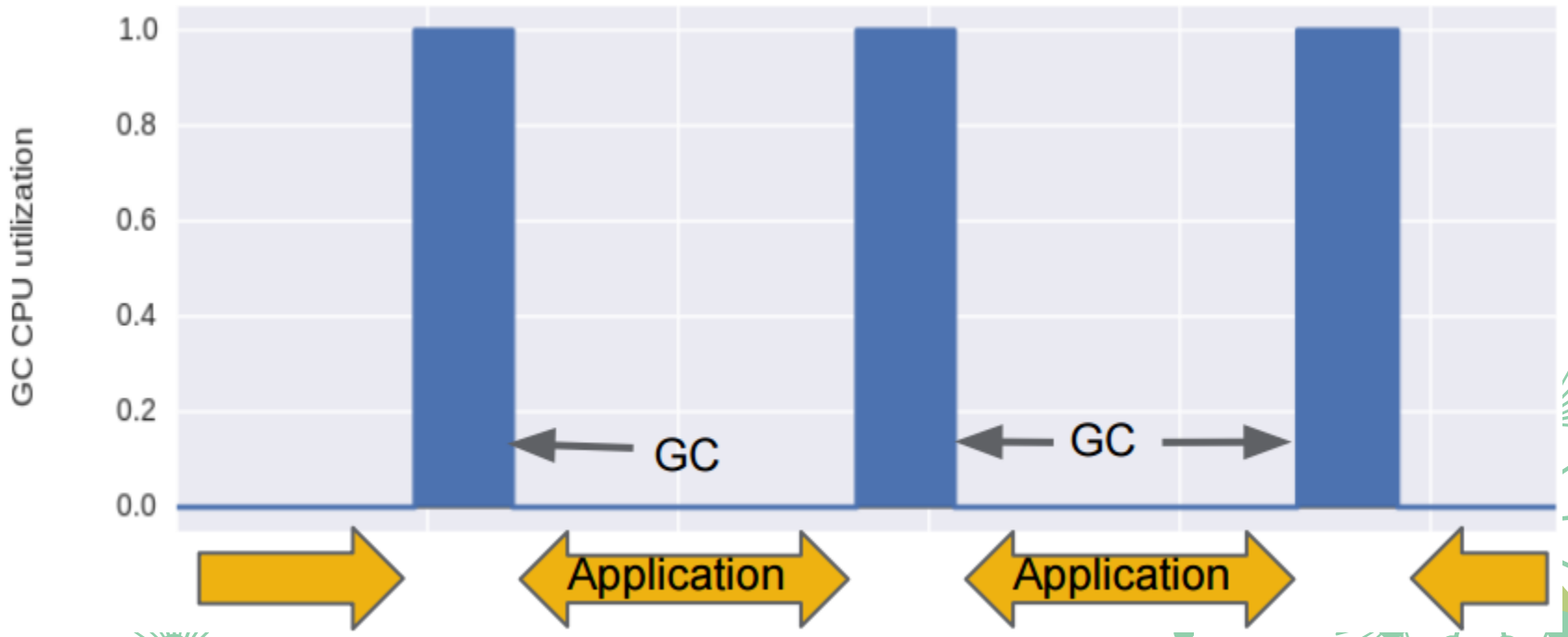
- Git
- Mercurial
- Subversion
- Bazaar



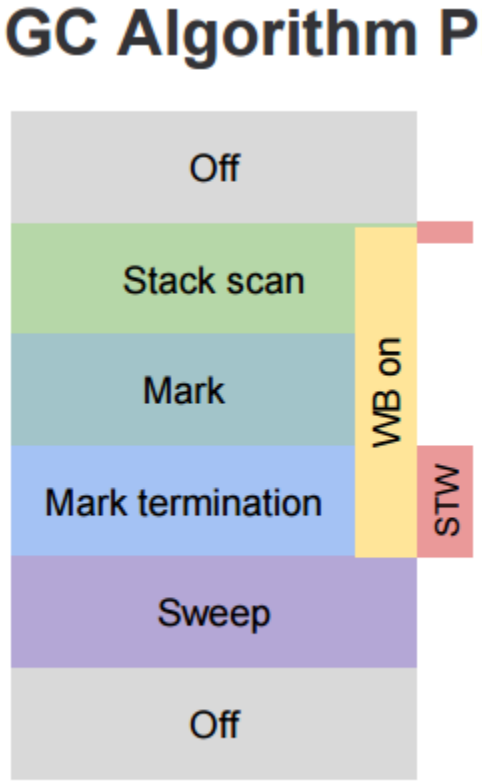
install package from web storage



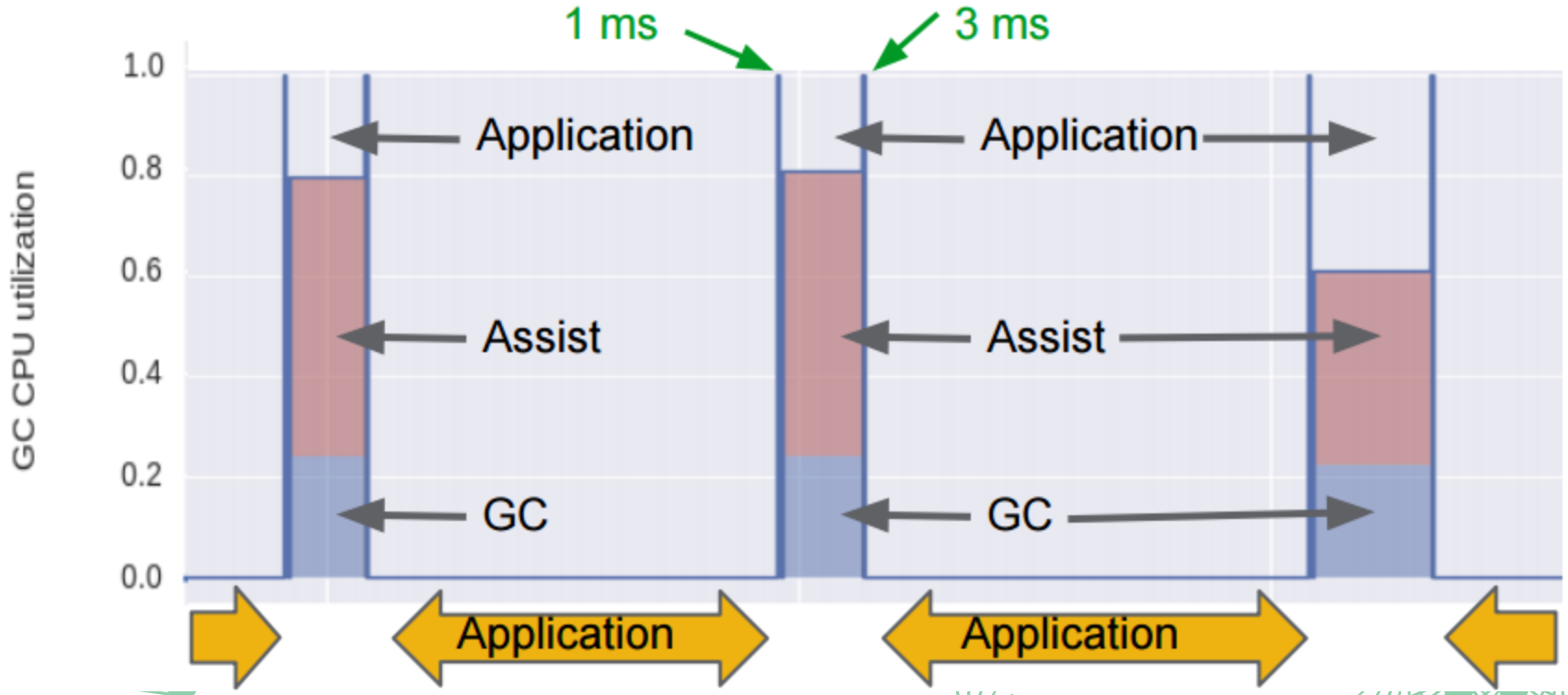
Garbage collection



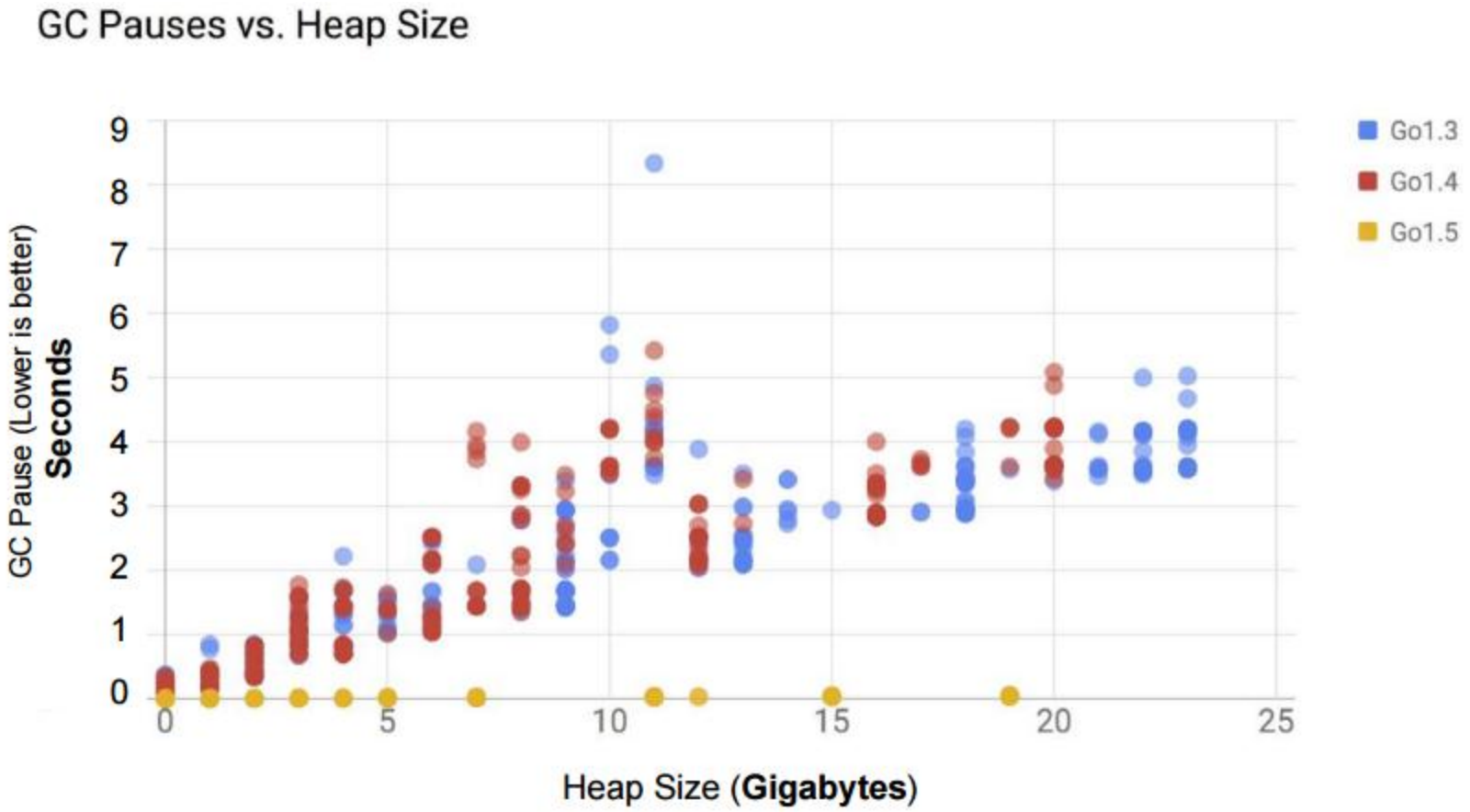
Garbage collection



Garbage collection



Garbage collection



Q&A