

# Agile Software Development

- 
- 
- 
- 
- ref

## Boilerplate code

- boilerplate code( ) boilerplate( )

## Brute force attack

- ID

## Local Brute Force Protection

가

## Network Brute Force Protection

가 IP

## Cloud Native

- ,
- 가 ,



# Functional Programming

- 함수형 프로그래밍은, 가변성 없이, 순수 함수를 사용하여 데이터를 처리하는 프로그래밍 패러다임이다.
- 함수형 프로그래밍의 특징은, 함수를 일급 객체로 취급하고, 불변 데이터를 사용하며, 선언적 코드를 작성하는 것이다.
- 함수형 프로그래밍은,  $f(x)$  가  $( : )$  x  $f$  ,

# GraphQL

- Graph Query Language
- Server API
- GraphQL API Endpoint
- GraphQL API Query Query Language 가

# Hang

- hang freeze System hang ,
- 가 ( ) ,
- System Hang Panic

# HTS

- Home Trading System
- PC

# IEEE 754

- IEEE 가 +0, -0 (Infinity), NaN ( Not a Number )

# In-Memory Computing

- 
- (system of record)
- , (stash)
- stash[ - : ( ) [ ]
- 가

## Kanban[ , ]

```
> Kanban .\
```

(看板) , ( 가 “

Kanban , “ ” ,

kaizen( , 改善( )), keiretsu( , 系列( )) 3-K ,

.( : Naver )

\* Ref Lnk [Kanban](#) ?

## log [ , ]

1. , , , , ,
2. , .

## logging [ ]

## Interpreter [ ]

(interpreter) 가 .

## meme

- ( Internet Meme ) ( Meme )
- 

## MTS

- Mobile Trading System
- 

## Proxy Server [ ]

- 가
- ( Proxy ) 가
- 가
- 가
- 가
- ( Proxy Server)

## pseudo-code[ ]

- (疑 / 似 | Pseudo : 가 ) 가 (
- )
- 

## QD, Quantum Dot [ ]

- [量子點] wlfmadl 2~10nm( : 10 1m ]
- 

## Refactoring [ ]

-

- 가 가

## REST, RESTful

- REST : REpresentational State Transfer
- Resource( , User,...) Endpoint , Endpoint Resource
- REST API RESTful API , API
- RESTful
- GraphQL RESTful API

## Sandbox [ ]

- ITWorld | (SandBox)

## Snippet

- 가 가
- 가 , , .( ) 가

## SOLID

- 가 가
1. S : SRP [Single Responsible Principle, ]
    - 가
  2. O : OCP [Open/Closed Principle, - ]
    -
  3. L : LSP [Liskov-substitution Principle, ]
    -
  4. I : ISP [Interface Segregation Principle, ]
    - 가

5. D : DIP [Dependency Inversion Principle, ]  
o “ ”

## Statically typed Language ( )

(Statically typed Language)

가

가

Java, C, C++, C#, Scala, Fortran, Haskell, ML, Pascal

:

가

:

## Dynamically typed Language ( )

( Dynamically typed Language )

### tokenizer [ ]

- tokenization [ ] :
- Token( )
- Token 가

### Use Case

- - 
  - 
  - ref (Use Case)
- 가

## Upper Case, Lower Case

- Uppercase [ ] : = capital letter [ ]
- Lowercase [ ] : = small letter [ ]
- [upper case](#), [lower case](#)

## Wireframe

- 
- [UI](#)
- [PENCIL PROJECT](#)

, [IT](#),

From:

<http://rwiki.repia.com/> -

. - 2023.12

Permanent link:

<http://rwiki.repia.com/doku.php?id=wiki:user:emblim98:terms&rev=1666841392>



Last update: **2022/10/27 12:29**