

Park 3. 프로그래밍 기법.

Keyword 정리.

◦ 각 세션 사이마다 시스템 세미터넷 저장

이동체지침, 함수형 언어

프로그래밍에서, 필요한 데이터를 추상화시켜 상태라. 행위.

종가인 객체제를 만들고, 객체들 간의 유기적 상호작용을 통해 구조를 구성.

장점: 코드 재사용, 유지보수, 대형 프로젝트

단점: 느린 처리속도, 용량, 설치시간.

기법: 클래스 + 인스턴스; 추상화를 거쳐, 속성(attribute), 행위(behavior)를 변수라. 매서드라.

추상화 → 인스턴스; 클래스에서 정의한 것을 실제로 실제 메모리에 할당된 것. 즉 데이터.

행위화 → 동작의 속성, 기능을 위해 이름을 붙인 것.

상속 → 기능과 속성의 묶음을 "클래스"라. "행위"에 대해

타입성 상속; 부모클래스의 속성과 기능을 그대로 이어받아, 사용 및 수정.

overriding, overloading

- R, 메모리 할당과라이더' 주. 결과값을, (객체: object 에 저장),

- 함수형 언어: 병렬 프로그래밍.

패러디: 원시인; 우리가 직접 > 정해진 코드의 모방.

목적론: 변역

함수, 주제별, 간략화

① source ("소스"), 스크립 파일 - 코드 실행

② 평가하기, if  $a = a-b-c$ , then

③ sink; (append, split), 시작과일에 새로운 내용 +

④ Paste(a.c(e', f'))

Console에 출력

$= a-b-c, e / a-b-c, f$  any other.

⑤ 매치

⑥ Paste(a, 10, sep = \*)

$= a-b-c * 10$

⑦ cat, "var list it. 행과열 4개.

⑧ <, local variable 인 <라 해서, Global variable로, Global env에서 사용 가능한 함수 체.

⑨ is.isstr(): 변수유형변기.

⑩ C: 숫자, 문자, 리얼, 복소점



## ① Matrix 생성 2가지 Method

ex)  $m = 1:6$ ,  $\dim(m) \leftarrow C(2, 3)$

$= \begin{bmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \end{bmatrix}$

2x)  $m = \text{matrix}(\text{hrow} = 2, \text{ncol} = 3, 1:6)$

$\dim(m) = 2 \ 3$

or,  $\text{attributes} = \dim \ 2 \ 3$

## ② 행렬의 용.

% \* % ; 0행 행렬의 용, 1행 행렬의 용의 용. 곱셈 가능.

## Row와 Column의 파일 저장

; Text > Tab-delimited text, Comma-separated text.

; Excel, JSON, HTML/XML, Database, Statistical SW.

## 2행 저장.

① list.files(dir, dot) ① recursive ; 리치 하위 디렉토리에서 검색함, all.files ; 숨겨진 파일도 포함함.  
path Pattern

\*  
② \, // ; inverted-slash > slash or double inverted slash

③ read.fwf ("자료.txt", widths = C(x, y) - 2, col.names)

④ read.table ; sep 문자, String As Factor : T/F > 엑셀도 로딩으로 인식함  
na.string, header : T/F

⑤ read.csv ; header = T/F, as.is = T/F 비구조적 데이터를 로딩으로 해석하기 가능함.

⑥ write.csv ; col.names = T/F, row.names = T/F

⑦ html of table 읽기.

library(XML) > url 지정 > t <- readHTMLTable(url)

⑧ 데이터 읽어 행 데이터 읽기, scan 중 list로 읽기

데이터 저장

scan ("1000100", what = list(), n = num, skip = num, na.string)

각 Column에 대한 dx type 지정      데이터 저장 row num



### 자료형 데이터의 구조 변환

```
1. with : with(iris, {  
  print(mean(Sepal.Length))  
  print(mean(Sepal.Width))})
```

```
2. attach : > attach(iris)  
           > head(Sepal.Width)  
           > detach(iris)  
           > Sepal.Width ... Error
```

자료형 변환의 as.logical

데이터의 구조 변환 as.data.frame, as.list, as.matrix, as.vector

데이터의 구조 변환