

자기주도 온라인 프로젝트

Day 7.

Collection 사용

최인국 컨설턴트



Day 7

CONTENTS

Samsung Software Academy For Youth



Collection 사용

최 인 국 Project consultant

- 부산 경성대학교 컴퓨터 공학 학사 졸업, 99학번
- Cymer (2006, 10 ~)
(반도체 장비 모니터링 시스템 개발)
- Future System (2013, 05 ~)
(네트워크 보안 장비 개발)
- ESTMOB (2014, 06 ~)
(기기간 파일 전송 솔루션)
- 프리랜서 (2017, 03 ~)
(카메라 모듈 제어, 출퇴근 시스템,
자재 관리 시스템, 동영상 녹화기)



Collection 적용



- 동일한 속성의 데이터들을 반복적으로 비교 연산
- 비교 연산 후 결정되는 동작이 비슷한 패턴으로 반복
- 모든 랭귀지에서 적용 가능

Collection 이란?



- Stack, Queue, Array,
List, Hash,
Dictionary..

```
public String getFruit(){  
  
    int random = RandomInt();  
  
    int i = random % 4;  
  
    switch(i) {  
        case 0:  
            return "Mango";  
        case 1:  
            return "Apple";  
        case 2:  
            return "Orange";  
        case 3:  
            return "Banana";  
    }  
}
```

```
public string getFruit(){  
  
    String fruits[] = {"Mango", "Apple", "Orange", "Banana"};  
  
    int random = RandomInt();  
  
    int i = random % fruits.getLength();  
  
    return fruits[i];  
}
```

Collection 적용



- 동일한 속성의 데이터들을 반복적으로 비교 연산
- 비교 연산 후 결정되는 동작이 비슷한 패턴으로 반복

```
function post() {  
  
    if(this.user == '') {  
        alert('로그인 해주세요');  
        return;  
    }  
    else if(this.title == '') {  
        alert('제목을 입력해 주세요');  
        return;  
    }  
    else if(this.body == '') {  
        alert('내용을 입력해 주세요');  
        return;  
    }  
    else if(this.image == '') {  
        alert('이미지를 첨부하세요');  
        return;  
    }  
  
    return postCore(this.user, this.title, this.body, this.image)  
}
```



```
function post() {  
  
  const checkList = [  
    [this.user, '로그인 해주세요'],  
    [this.title, '제목을 입력해 주세요'],  
    [this.body, '내용을 입력해 주세요'],  
    [this.image, '이미지를 첨부하세요']]  
  
  for(const v of checkList) {  
    if(v[0] == '') {  
      alert(v[1]);  
      return;  
    }  
  }  
  
  return postCore(this.user, this.title, this.body, this.image)  
}
```

```
def forward():  
    print('move forward')  
  
def left():  
    print('move left')  
  
def right():  
    print('move right')  
  
def backward():  
    print('move backward')  
  
def move(key):  
  
    if(key == 'w'):  
        forward()  
    elif(key == 'a'):  
        left()  
    elif(key == 'd'):  
        right()  
    elif(key == 's'):  
        backward()
```

```
def forward():  
    print('move forward')  
  
def left():  
    print('move left')  
  
def right():  
    print('move right')  
  
def backward():  
    print('move backward')  
  
def move(key):  
  
    dict = {'w':forward, 'a':left, 'd':right, 's':backward}  
  
    func = dict.get(key)  
  
    if func is not None:  
        func()
```

Collection



- Collection 을 만들기 위한 CPU/메모리 비용 발생
- 가독성, 메모리/CPU 사용량 등 종합적 고려하여 상황에 따라 적절하게 적용
- collection 을 초기에 한번 만들고, 사용을 여러 번 할 경우, 성능이 크게 개선

```
public class A {  
  
    public String getFruit() {  
  
        int random = RandomInt();  
  
        int i = random % 4;  
  
        switch(i) {  
  
            case 0:  
                return "Mango";  
            case 1:  
                return "Apple";  
            case 2:  
                return "Orange";  
            case 3:  
                return "Banana";  
            default:  
                return null;  
        }  
    }  
}
```

```
public class B {  
  
    String fruits[] = {"Mango", "Apple", "Orange", "Banana"};  
  
    public String getFruit() {  
  
        int random = RandomInt();  
  
        int i = random % this.fruits.length;  
  
        return this.fruits[i];  
    }  
}
```

```
const date = new Date()
const year = date.getFullYear()
const month = date.getMonth()
var days = null

switch(month){
  case 0:
  case 2:
  ....
  case 11:
    days = 31;
    break;

  case 3:
  ....
  case 10:
    days = 30;
    break;
  case 1:
    if((year % 4 == 0) && (year % 100 != 0) || (year % 400) == 0)
      days = 29;
    else
      days = 28;
    break;
}

console.log(days + ' days for ' + year + '-' + (month + 1))
```

마무리



코로나19 조심 하세요.

감사 합니다.