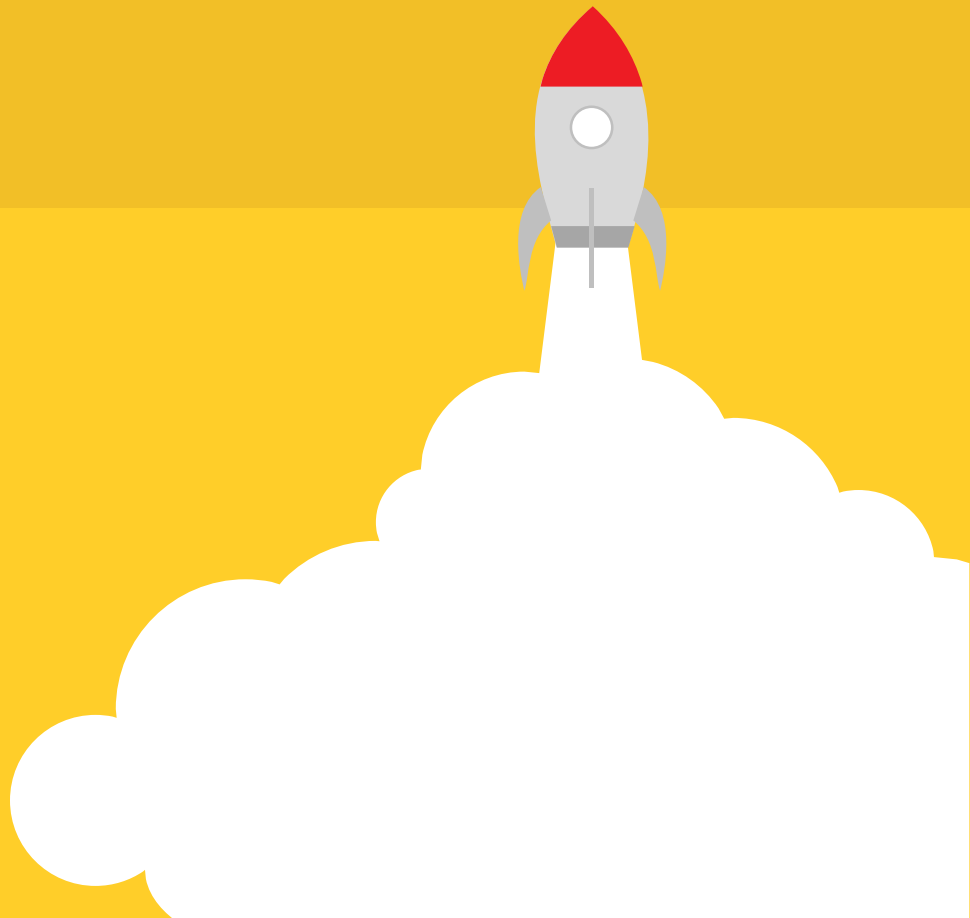


C# 7강. 프로젝트

스마트 밥솥

프로그래밍 강사 : 이태성





목차

01

프로젝트 목표와 개요
필수 클래스와 메서드

02

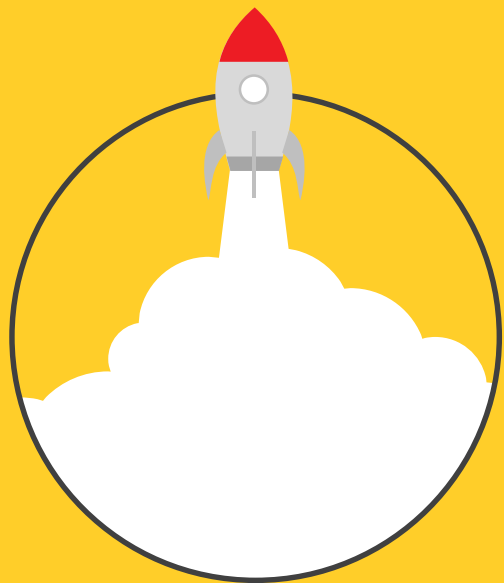
기획서 작성
프로그램 화면

03

메뉴와 전체 구조
전체 구조

04

코드 보기
정리



01 프로젝트

목표와 개요

프로젝트 목표

- C# 기본 문법 활용
- 프로그래밍에 대한 이해
- 필요한 부분을 MSDN에서 검색
스스로 찾아 학습하는 능력



프로젝트 개요

- 전기 밥솥에서 스마트 밥솥으로 업그레이드
- 쌀, 물, 씻기 등을 자동화
- 가전 제품 동작을 프로그래밍으로 구현
- 역동적인 프로그래밍 구현
→ 시각 + 청각

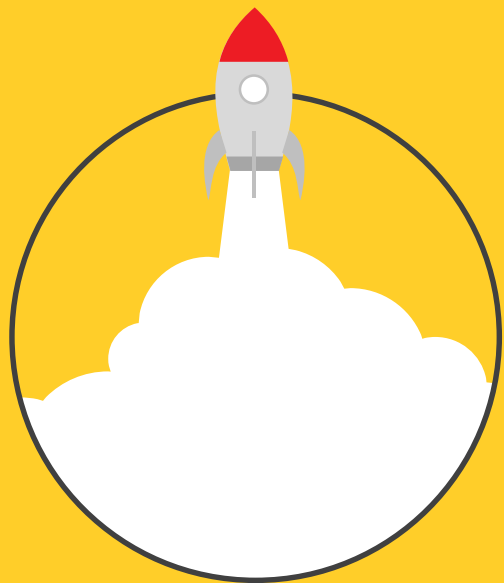


프로젝트 개요

- 전기 밥솥에서 스마트 밥솥으로 업그레이드
- 쌀, 물, 씻기 등을 자동화
- 가전 제품 동작을 프로그래밍으로 구현
- 역동적인 프로그래밍 구현
→ 시각 + 청각







02 프로젝트를 위한 필수 클래스와 메서드

콘솔창 관련 클래스와 메서드

- 화면 출력 클래스

Console

- 화면 출력 메서드

Console.Beep()

Console.Clear()

Console.SetCursorPosition()

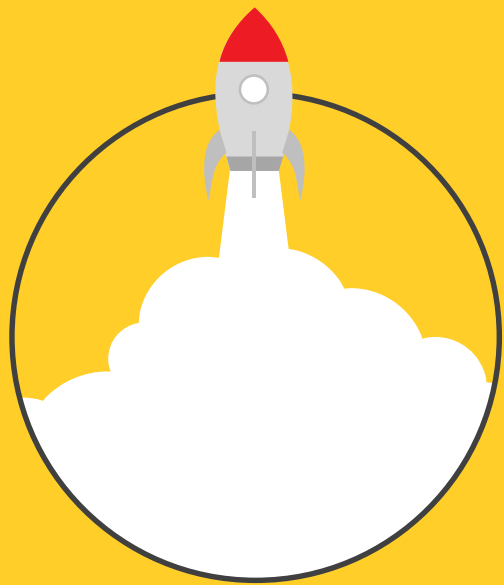
Console.SetWindowSize()

Thread.Sleep()

콘솔창 관련 클래스와 메서드

- 콘솔 색상관련 속성
Console.BackgroundColor
Console.ForegroundColor
- 임의의 위치로 문자열을 출력해 주는
프로그램

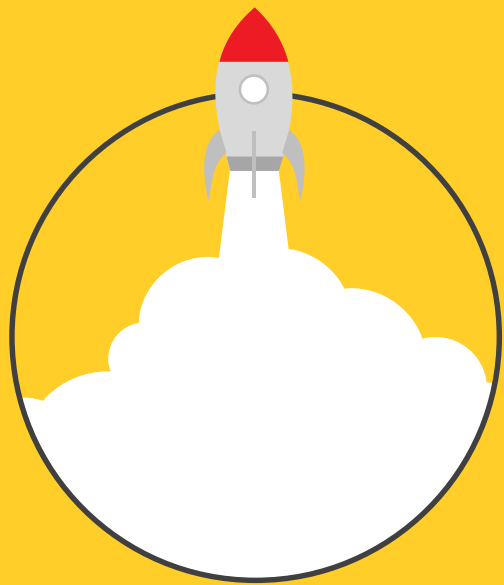
[소스 보기](#)



03 기획서 작성

기획서

- 생각을 정리하고 문서화
- 세부적인 기획서는 코드를 간결
- 업그레이드 되는 프로그램
- 기획서 항목
개요, 기능, 개선 사항
- www.nowcampus.co.kr 참고



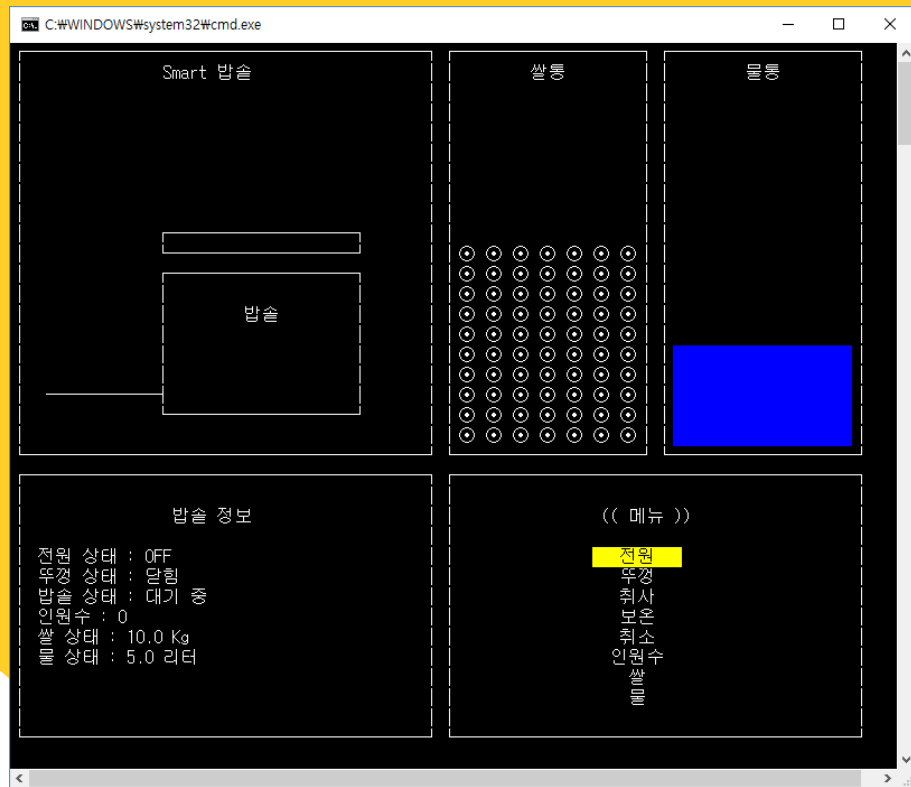
04 프로그램 화면

프로그램 화면

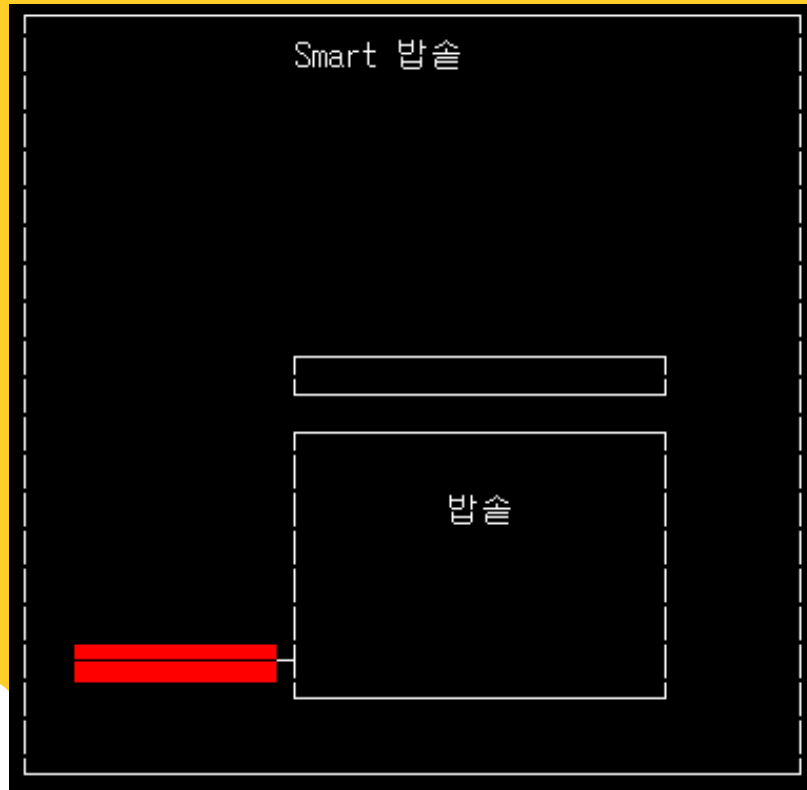
- 화면 디자인이 바뀌면 코드도 바뀜
- 화면 디자인은 사용자 인터페이스임
- 간결하지만 시각적으로 충분한 화면 구성



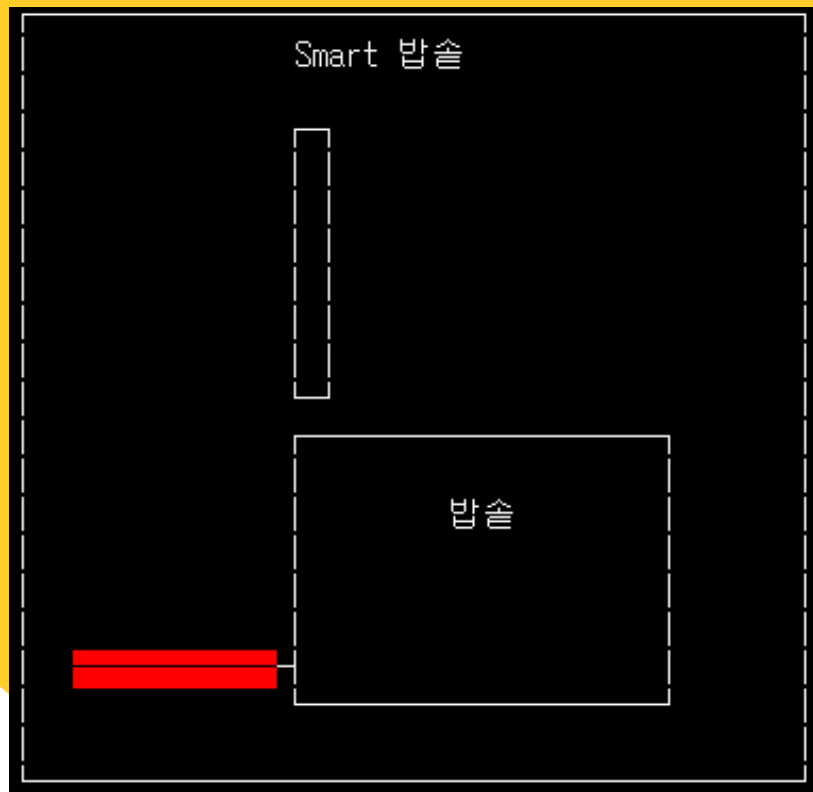
전체 화면



전원 On/Off

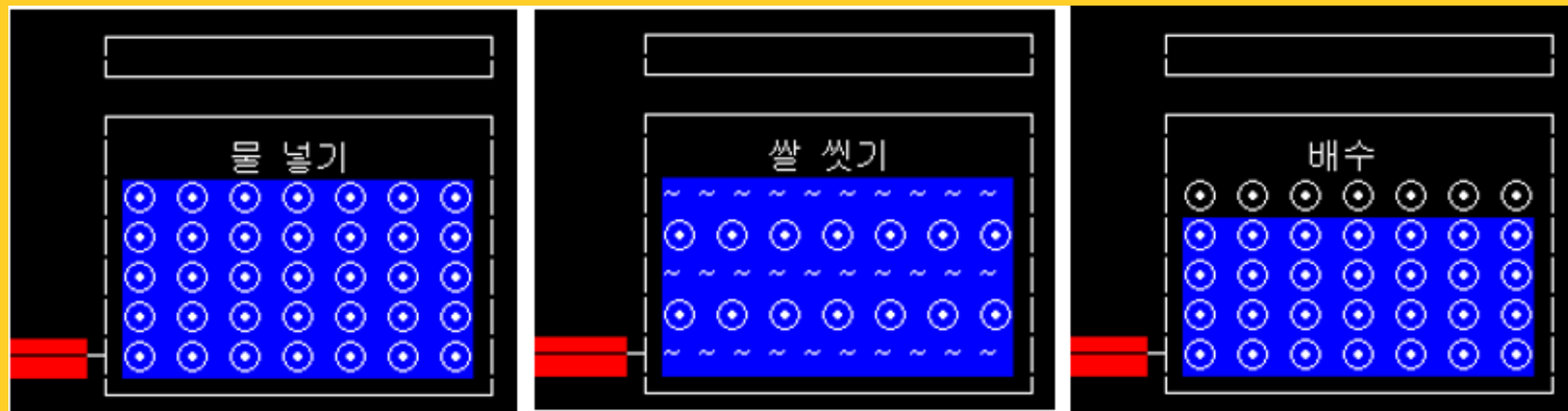


뚜껑 열기/닫기

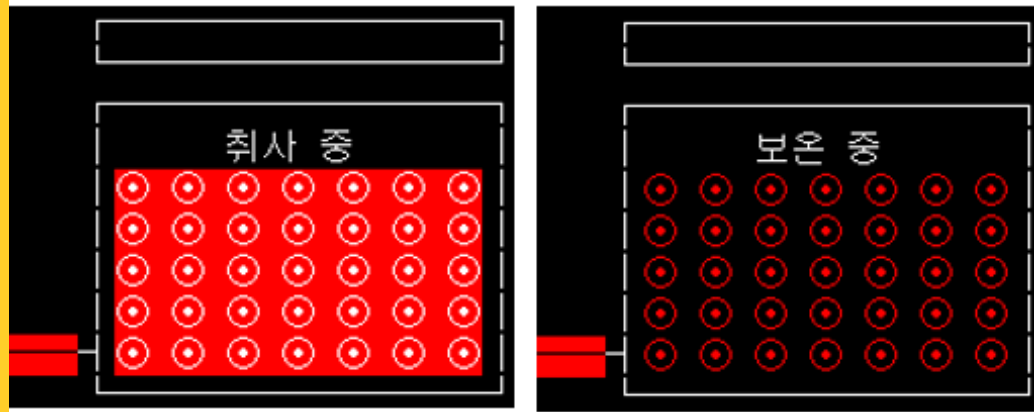


밥솥 취사 전 동작

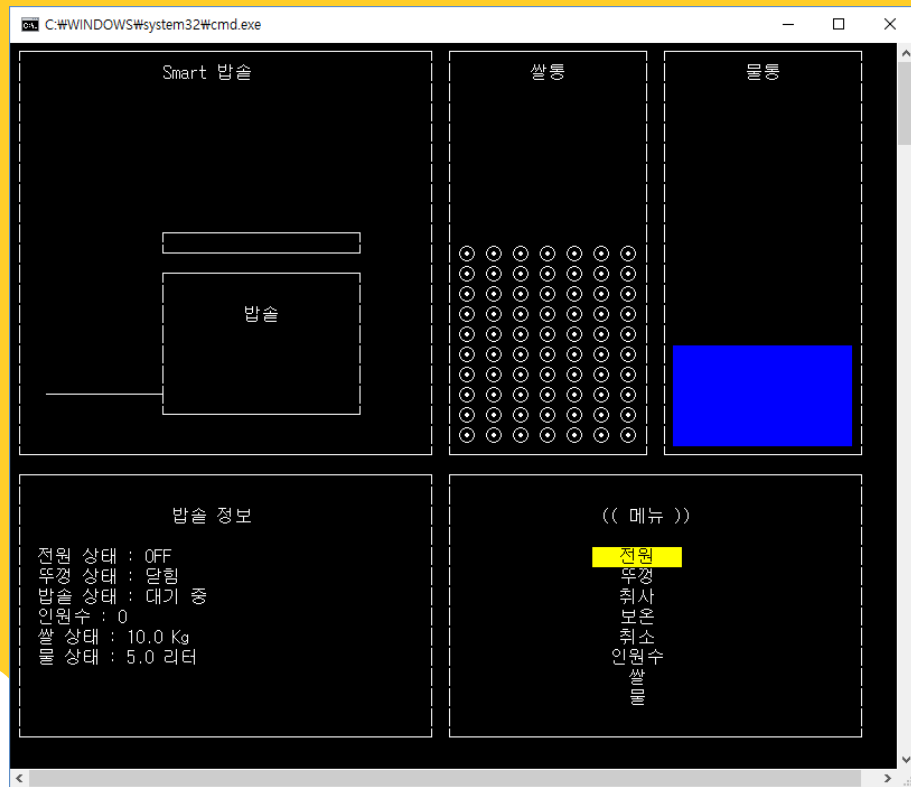
- 물 넣기, 쌀 씻기, 배수

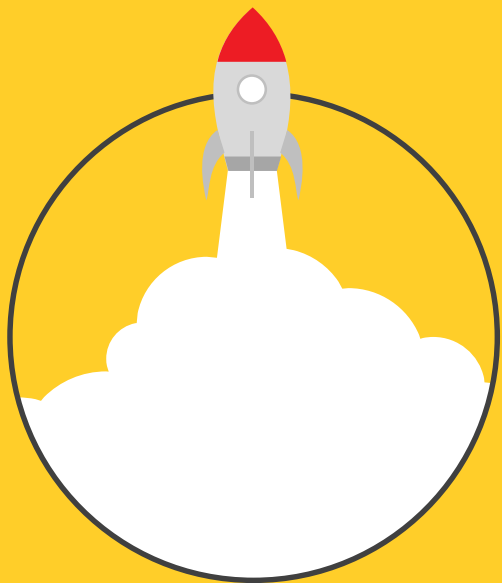


취사와 보온



프로그램 시연





05 메뉴

메뉴 형태

- | | |
|------|---|
| 1. 입 | 력 |
| 2. 수 | 정 |
| 3. 삭 | 제 |
| 4. 삽 | 입 |
| 5. 저 | 장 |
| 6. 종 | 료 |

((메뉴))

전원
뚜껑
취사
보온
취소
인원수
쌀물

메뉴 번호를 입력해 주세요! []

메뉴 형태

- 배경색과 전경색 활용

`Console.BackgroundColor`

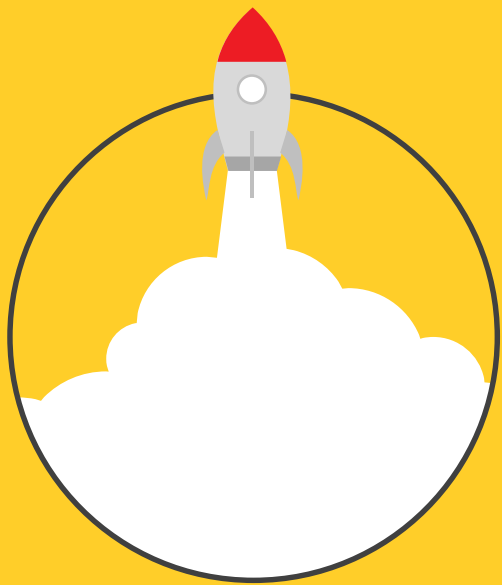
`Console.ForegroundColor`

[소스 보기](#)

- 키 입력에 따라 메뉴바 이동

[소스 보기](#)





06 전체 구조

화면 디자인 코드

- 코드

소스 보기



메뉴에 따른 전체 처리 구조

- 메뉴 처리 ➔ 선택

- while(true)

```
{
```

```
    MenuIndex = Menu(); // 메뉴 출력
```

```
    switch(MenuIndex)
```

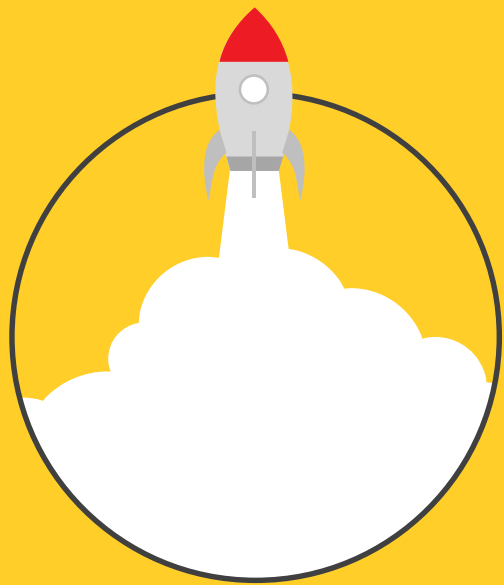
```
{
```

```
    case 0 : // 전원 처리
```

```
        break;
```

```
    case 1 : // 뚜껑 처리
```

```
        break;
```



07 코드 보기

밥솥 데이터 구조

- 밥솥 상태 표현 ➔ 열거형

```
enum CookerProcess { None, Riceing,  
    Watering, Washing, Droping, Cooking,  
    Completion, Keeping  
};
```

밥솥 데이터 구조

- 밥솥 상태 표현 ➡ 열거형

```
enum CookerProcess { None, Riceing,  
    Watering, Washing, Droping, Cooking,  
    Completion, Keeping  
};
```

밥솥 데이터 구조

- 구조체

```
struct RiceCookerInfo
{
    public bool CoverOpenClose; // 뚜껑 열기 닫기
    public bool Power; // 전원 On/Off
    public CookerProcess State; // 밥솥 진행 상태
    public int Rice; // 쌀의 양 g 기준, 출력 때 환산
    public int Water; // 물의 양 미리 리터 기준, 출력때 환산
    public int Number; // 인원수
```

사운드 출력

- SoundPlayer
 - using System.Media
 - wav 파일 출력

- 코드

```
SoundPlayer Sound = new SoundPlayer();  
Sound.SoundLocation = "xxx.wav";  
Sound.Load();  
Sound.Play(); // 또는 Sound.PlaySync()
```

취사 코드

- 취사 루프



- 쌀 넣기 전에 조사하는 코드
전원, 뚜껑, 인원 수, 쌀, 물

[소스 보기](#)

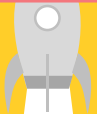
취사 코드

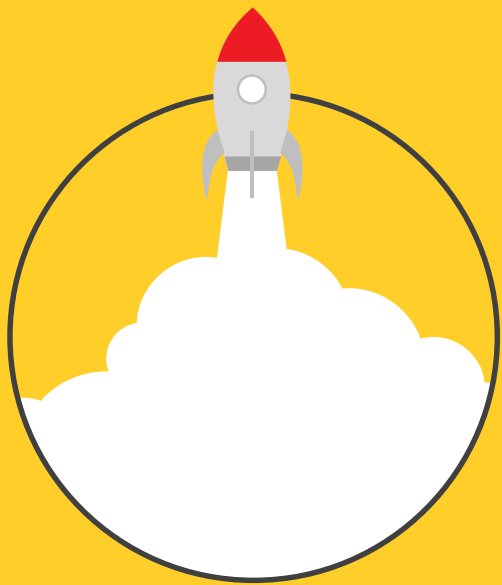
- 쌀 넣기 부분

소스 보기

- 물 넣기, 쌀 씻기, 배수

소스 보기





08 정리

정리

- 프로젝트에서 제외된 항목
예약 취소
- 기초 문법만으로 C# 프로그래밍 경험
- 다양한 소재의 프로젝트
전기 장판, 전기 청소기, 스마트 침대,
일반적인 관리 프로그램

정리

- MSDN 사용과 예제 코드
- C# 프로그래밍?

기본 문법 + 기본 클래스 라이브러리 사용

