

# Entry 블록 커스터마이징 가이드

## 1. 블록 메타데이터 JSON 파일 수정

### 1.1 when\_run\_button\_click.json 예시

경로: `data/blocks/start/when_run_button_click.json`

```
json

{
  "name": "시작하기 버튼을 클릭했을 때",
  "keywords": [
    "시작", "실행", "처음", "게임시작", "프로그램시작",
    "시작버튼", "실행버튼", "초록깃발", "그린플래그",
    "런", "run", "start", "begin", "초기화"
  ],
  "common_questions": [
    "게임을 시작하려면",
    "시작 버튼을 누르면",
    "처음 실행할 때",
    "프로그램이 시작될 때",
    "실행 버튼 클릭하면"
  ]
}
```

### 1.2 move\_direction.json 예시

경로: `data/blocks/moving/move_direction.json`

```
json

{
  "name": "() 만큼 움직이기",
  "keywords": [
    "이동", "움직임", "앞으로", "전진", "걸기",
    "move", "forward", "walk", "step"
  ],
  "common_questions": [
    "캐릭터를 움직이려면",
    "앞으로 가게 하려면",
    "이동시키고 싶어"
  ]
}
```

## 2. RAG 검색 매핑 테이블 (background.js)

### 2.1 블록별 직접 매핑 코드

javascript

```
const BLOCK_DIRECT_MAPPINGS = {  
  // 시작 카테고리  
  "게임.*시작|프로그램.*시작|실행.*버튼|시작.*버튼": {  
    file: "when_run_button_click",  
    category: "start"  
  },  
  "스페이스.*키|스페이스바": {  
    file: "when_some_key_pressed",  
    category: "start"  
  },  
  "클릭했을.*때|마우스.*클릭": {  
    file: "when_object_click",  
    category: "start"  
  },  
  
  // 움직임 카테고리  
  "앞으로.*움직|전진|이동하기": {  
    file: "move_direction",  
    category: "moving"  
  },  
  "회전|돌기|각도.*바꾸기": {  
    file: "rotate_relative",  
    category: "moving"  
  },  
  
  // 흐름 카테고리  
  "반복|계속|여러.*번": {  
    file: "repeat_basic",  
    category: "flow"  
  },  
  "만약|조건|경우": {  
    file: "_if",  
    category: "flow"  
  }  
};
```

### 3. 블록별 오류 패턴 수정

#### 3.1 자주 발생하는 잘못된 응답 교정

javascript

```
const BLOCK_CORRECTIONS = {  
  "when_run_button_click": {  
    wrong: ["이벤트 카테고리", "컨트롤 카테고리", "시작하기 블록"],  
    correct: "시작 카테고리의 '시작하기 버튼을 클릭했을 때' 블록"  
  },  
  "move_direction": {  
    wrong: ["모션 카테고리", "동작 카테고리"],  
    correct: "움직임 카테고리의 '() 만큼 움직이기' 블록"  
  },  
  "_if": {  
    wrong: ["조건 카테고리", "판단 카테고리"],  
    correct: "흐름 카테고리의 '만약 () 라면' 블록"  
  }  
};
```

---

### 4. 우선순위 20개 블록 집중 관리

#### 4.1 우선순위 블록 리스트

javascript

```
const PRIORITY_BLOCKS = [  
  // 최우선 10개 (가장 자주 사용)  
  { file: "when_run_button_click", category: "start", priority: 10 },  
  { file: "move_direction", category: "moving", priority: 10 },  
  { file: "_if", category: "flow", priority: 10 },  
  { file: "repeat_basic", category: "flow", priority: 10 },  
  { file: "set_variable", category: "variable", priority: 10 },  
  
  // 차순위 10개  
  { file: "when_some_key_pressed", category: "start", priority: 8 },  
  { file: "wait_second", category: "flow", priority: 8 },  
  { file: "rotate_relative", category: "moving", priority: 8 },  
  { file: "change_variable", category: "variable", priority: 8 },  
  { file: "show", category: "looks", priority: 8 }  
];
```

## 4.2 테스트 케이스 자동 생성

javascript

```
function generateTestCases() {  
  const testCases = [];  
  
  PRIORITY_BLOCKS.forEach(block => {  
    testCases.push({  
      input: `${block.category} 관련 질문`,  
      expected: block.file,  
      category: block.category  
    });  
  });  
  
  return testCases;  
}
```

## 5. 블록별 수정 체크리스트

### 핵심 20개 블록 작업 현황

#### ☒ 완료된 블록

#### ☒ when\_run\_button\_click (시작하기 버튼을 클릭했을 때)

- keywords 추가 완료
- common\_questions 추가 완료
- 잘못된 응답 패턴 수집 완료
- 테스트 케이스 5개 작성 완료

#### ☐ 작업 예정 블록

#### 시작 카테고리

- ☐ when\_some\_key\_pressed (키를 눌렀을 때)
- ☐ when\_object\_click (오브젝트를 클릭했을 때)
- ☐ when\_scene\_start (장면이 시작되었을 때)

#### 움직임 카테고리

- ☐ move\_direction (() 만큼 움직이기)
- ☐ rotate\_relative (() 도 회전하기)
- ☐ locate\_xy (x() y() 위치로 이동하기)

#### 흐름 카테고리

- ☐ \_if (만약 () 라면)
- ☐ if\_else (만약 () 라면 아니면)
- ☐ repeat\_basic (() 번 반복하기)
- ☐ repeat\_inf (계속 반복하기)
- ☐ wait\_second (() 초 기다리기)

#### 자료 카테고리

- ☐ set\_variable (변수 () 를 () (으)로 정하기)
- ☐ change\_variable (변수 () 를 () 만큼 바꾸기)
- ☐ get\_variable (변수 ())

#### 판단 카테고리

- ☐ boolean\_and\_or (() 그리고/또는 ())
- ☐ boolean\_basic\_operator (() = ())
- ☐ is\_press\_some\_key (() 키를 눌렀는가?)

#### 생김새 카테고리

- ☐ show (보이기)
  - ☐ hide (숨기기)
- 

## 6. 작업 순서 가이드

### Step 1: JSON 파일 수정 (블록당 30분)

1. `keywords` 배열에 관련 키워드 10-15개 추가
2. `common_questions` 배열에 자주 나오는 질문 패턴 5-10개 추가
3. `description` 필드에 상세 설명 추가

### Step 2: 매핑 테이블 업데이트 (블록당 10분)

1. `BLOCK_DIRECT_MAPPINGS`에 정규식 패턴 추가
2. `BLOCK_CORRECTIONS`에 잘못된 응답 패턴 추가

### Step 3: 테스트 케이스 작성 (블록당 20분)

1. 일반적인 질문 3개
2. 엣지 케이스 2개
3. 예상 응답 검증

Step 4: 실제 테스트 (블록당 20분)

- 1. Entry 사이트에서 실제 질문 테스트
- 2. 잘못된 응답 기록
- 3. 수정사항 반영

7. 예상 작업 시간

작업 내용	블록당 시간	20개 총 시간
JSON 수정	30분	10시간
매핑 테이블	10분	3.3시간
테스트 케이스	20분	6.7시간
실제 테스트	20분	6.7시간
총 작업 시간	80분	약 27시간

8. 팁과 주의사항

효율적인 작업을 위한 팁

- 1. 카테고리별로 묶어서 작업: 같은 카테고리 블록들은 유사한 패턴을 가짐
- 2. 템플릿 활용: 첫 번째 블록을 완벽하게 만들고 복사-수정
- 3. 점진적 개선: 완벽하게 만들려 하지 말고 80% 수준에서 출시 후 개선

주의사항

- 1. Entry 공식 용어 사용: "이벤트", "컨트롤" 같은 잘못된 카테고리명 사용 금지
- 2. 한국어 표현 통일: "시작하기" vs "스타트" → "시작하기"로 통일
- 3. 테스트 데이터 백업: 수정 전 원본 파일 백업 필수

9. 성과 측정 지표

정량적 지표












- 정확도: 20개 블록에 대한 정답률 85% 이상
- 응답 시간: 평균 2초 이내
- 오류율: 잘못된 카테고리 언급 5% 이하

## 정성적 지표

- 사용자 만족도: 5점 만점 4점 이상
  - 학습 효과: 단계별 힌트로 자기주도 학습 유도
  - 오류 감소: "이벤트 카테고리" 같은 오류 0%
- 

## 10. 부록: 빠른 참조

### Entry 공식 카테고리명

-  시작 (start)
-  움직임 (moving)
-  생김새 (looks)
-  소리 (sound)
-  판단 (judgement)
-  흐름 (flow)
-  자료 (variable)
-  함수 (func)
-  계산 (calc)
-  붓 (brush)
-  이벤트 (존재하지 않음)
-  컨트롤 (존재하지 않음)