

MOCCA Studio Manual

<목차>

1. MOCCA Studio
2. MOCCA 로봇
3. 프로그램 실행 절차
4. 프로그램 UI
5. 변수
6. 블록의 세부 구조
7. 블록 기능 상세 설명
8. 함수 만들기

1. MOCCA Studio

MOCCA(My Own Cognitive Communication Agent) Studio 는 한성대학교 교육로봇연구실(REEL: Robot in Education & Entertainment Laboratory)에서 개발한 비전문가를 위한 블록 기반의 로봇 콘텐츠 제작 프로그램입니다. 세계적으로 확산되고 있는 코딩 교육의 정규화 및 인공지능 교육 수요에 적합한 새로운 개념의 소프트웨어 제공을 목적으로 인공지능 기반의 하이 레벨 함수를 활용하여 소셜 로봇의 상호적인 시나리오를 프로그래밍 및 교육할 수 있는 통합 솔루션을 제공하기 위해 개발되었습니다. MOCCA Studio 는 Unity Engine 기반으로 개발되었고 클라우드 기반의 통합개발환경을 제공하며, 시나리오 편집기, 시나리오 실행기, 가상(virtual) MOCCA 로봇으로 구성되어 있습니다. 기본적인 프로그래밍에 필요한 함수는 물론, 자연스럽고 지능적인 상호작용을 위해 영상 기반 지능 함수(얼굴 인식, 성별 인식, 연령대 인식 등) 및 자연어 기반 지능 함수(대사에 알맞은 표정, 동작을 자동으로 실행하는 모듈, 긍정/부정 대답 인식 등)가 내장되어 있습니다.

MOCCA Studio 는 세계적으로 유일한 새로운 개념의 소셜 로봇 시나리오 프로그래밍 솔루션인 동시에, 다양한 인공지능 함수들을 통합하여 직접 활용해 볼 수 있는 환경을 제공합니다. 초보자에게는 로봇의 말, 표정, 행동을 활용하여 프로그래밍의 개념을 쉽고 재밌게 익힐 수 있는 도구로 활용될 수 있습니다. 또한 인공지능 모듈을 개발할 수 있는 고급 수준의 사용자에게는 자신이 개발한 인공지능 블록을 직접 편집기에 통합하여 시나리오를 작성할 수 있어 개발한 인공지능 함수의 유용성을 검증할 수 있습니다. 클라우드 기반의 개발 환경이므로 자신이 제작한 콘텐츠와 다른 사용자의 콘텐츠가 공개되어 보다 큰 시너지 효과를 기대할 수 있습니다.

이제부터 MOCCA Studio 그리고 MOCCA 로봇과 함께 로봇 콘텐츠를 제작해 볼까요?



2. MOCCA 로봇

MOCCA Studio에는 virtual twin 로봇인 MOCCA 로봇이 있습니다. 이는 로봇 하드웨어가 없어도 말과 영상을 사용한 상호작용 체험이 가능하도록 해 주는 소프트웨어 로봇으로 실물 로봇과 동시에 제어할 수 있는 가상 로봇입니다. MOCCA는 차동구동형 모바일 베이스, 3자유도 팔 2세트, 2자유도 목으로 구성되며, 3D 프린팅과 아두이노 호환 제어기로 실물 로봇까지 제작할 수 있습니다. MOCCA는 말하기, 얼굴 표정, 동작, 주행이 가능합니다. MOCCA의 외형은 갈색의 곰 인형과 같은 형태로 친숙한 디자인으로 사용자에게 접근성을 높여 줍니다.



말하기를 할 때 MOCCA의 입은 음성출력의 발음기호에 따라 실시간 립싱크 애니메이션 처리가 되어 있습니다. 아래는 각 발음기호에 따른 입 모양입니다.

PP	FF	TH	DD
CH	RR	AA	E
I	O	U	<립싱크 애니메이션 목록>

MOCCA는 감정 표현에 따라 얼굴 표정 애니메이션이 변화합니다. 아래는 표정 애니메이션 목록입니다.

기본 / 없음	행복	화남
		
두려움	쳐다보기(상)	쳐다보기(하)
		
쳐다보기(좌)	쳐다보기(우)	슬픔
		
미소	놀람	윙크(좌)
		
윙크(우)		
		

<표정 애니메이션 목록>

아래는 MOCCA의 동작 목록과 주행 목록입니다.

하이	헬로	화남	슬픔	오케이
박수	아니	틀림	행복	끄덕이기(우)
끄덕이기(좌)	고개들기	고개들기(좌)	고개들기(우)	고개돌림(좌)
고개돌림(우)	고개숙임	고개숙임(좌)	고개숙임(우)	없음

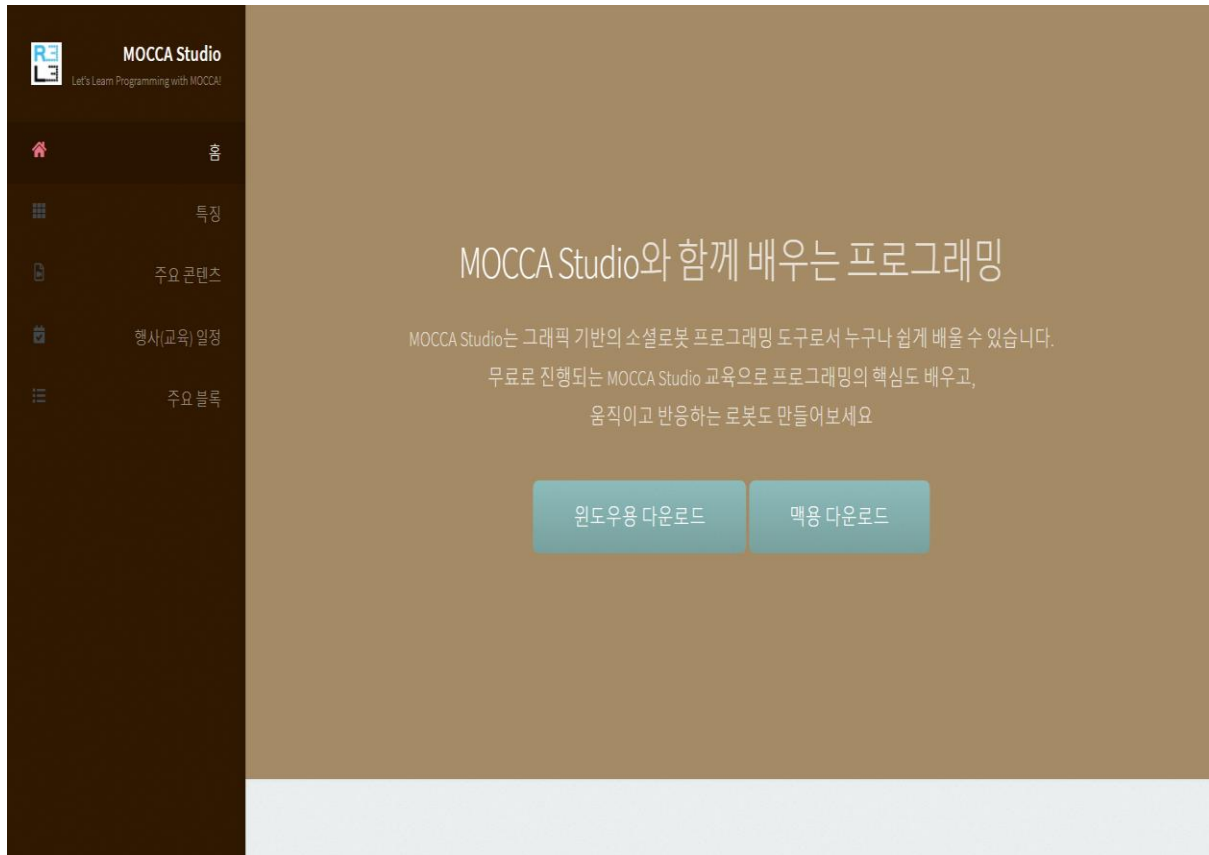
<동작 목록>

전진	후진	좌회전	우회전	정지
----	----	-----	-----	----

<주행 목록>

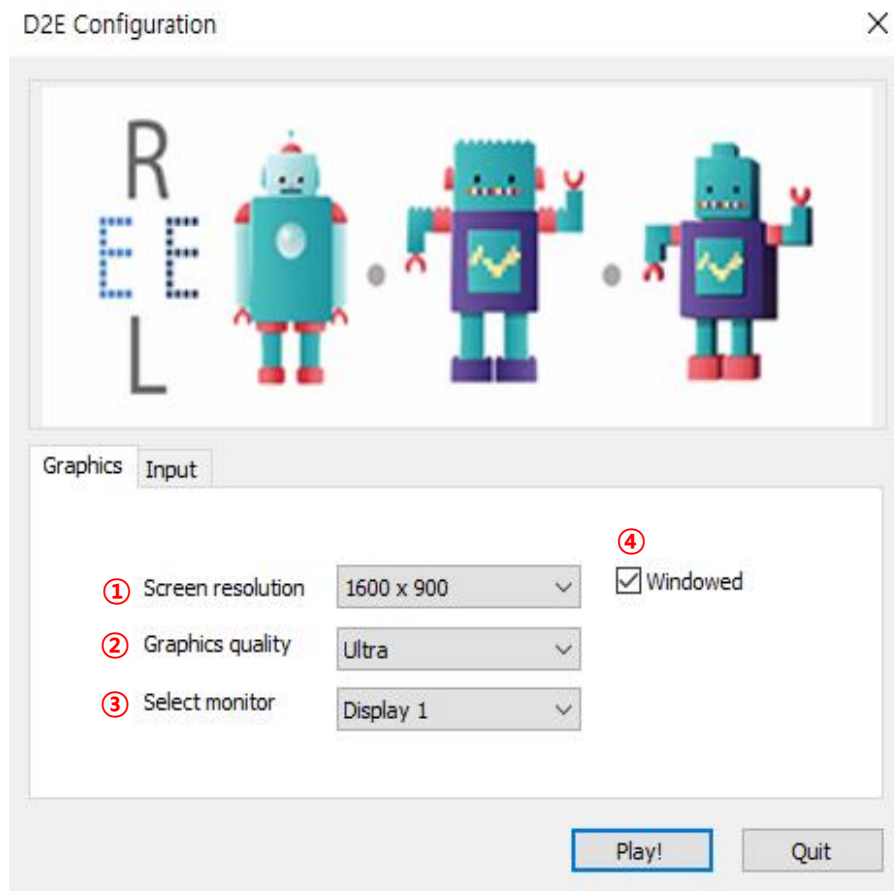
3. 프로그램 실행 절차

1. reel.hansung.ac.kr/mocca.html 사이트에 접속합니다.



2. 사용하는 OS에 해당하는 버튼을 클릭하여 알집 파일을 다운로드 받습니다.
3. 알집 파일을 풀어줍니다.
4. 생성된 폴더에서 D2E_x64.exe 파일을 클릭하여 프로그램을 실행합니다.

5. 팝업창에서 설정할 수 있는 기능에 대해 설명하도록 하겠습니다.



- ① Screen resolution: 프로그램 스크린 창의 크기를 설정합니다.
- ② Graphics quality: 프로그램 그래픽 수준을 설정합니다. 원활한 프로그램 실행을 위해 High 이상을 설정하는 것을 권장합니다.
- ③ Select monitor: 프로그램을 띄울 모니터를 선택합니다.
- ④ Windowed: 창 모드를 설정합니다.
 - 체크가 있으면 창 모드, 체크가 없으면 전체화면으로 프로그램이 실행됩니다.
 - 창 모드에서 전체화면으로 전환하려면 상단바를 더블 클릭하거나 최대화 버튼을 클릭합니다.
 - 전체화면에서 창 모드로 전환하려면 Alt+Enter를 통해 설정합니다.

6. 좌측 상단에 있는 로그인 버튼을 클릭합니다.



7. 사용자 등록을 위해 우측 하단에 있는 회원가입을 클릭한 후 회원가입 해줍니다. 회원가입을 마치면 자동으로 로그인 됩니다.

* 비밀번호는 6글자 이상으로 입력해야 합니다.

Login

이메일

비밀번호

로그인

[비밀번호 찾기](#)
[회원가입](#)

→

Sign Up

이름

이메일

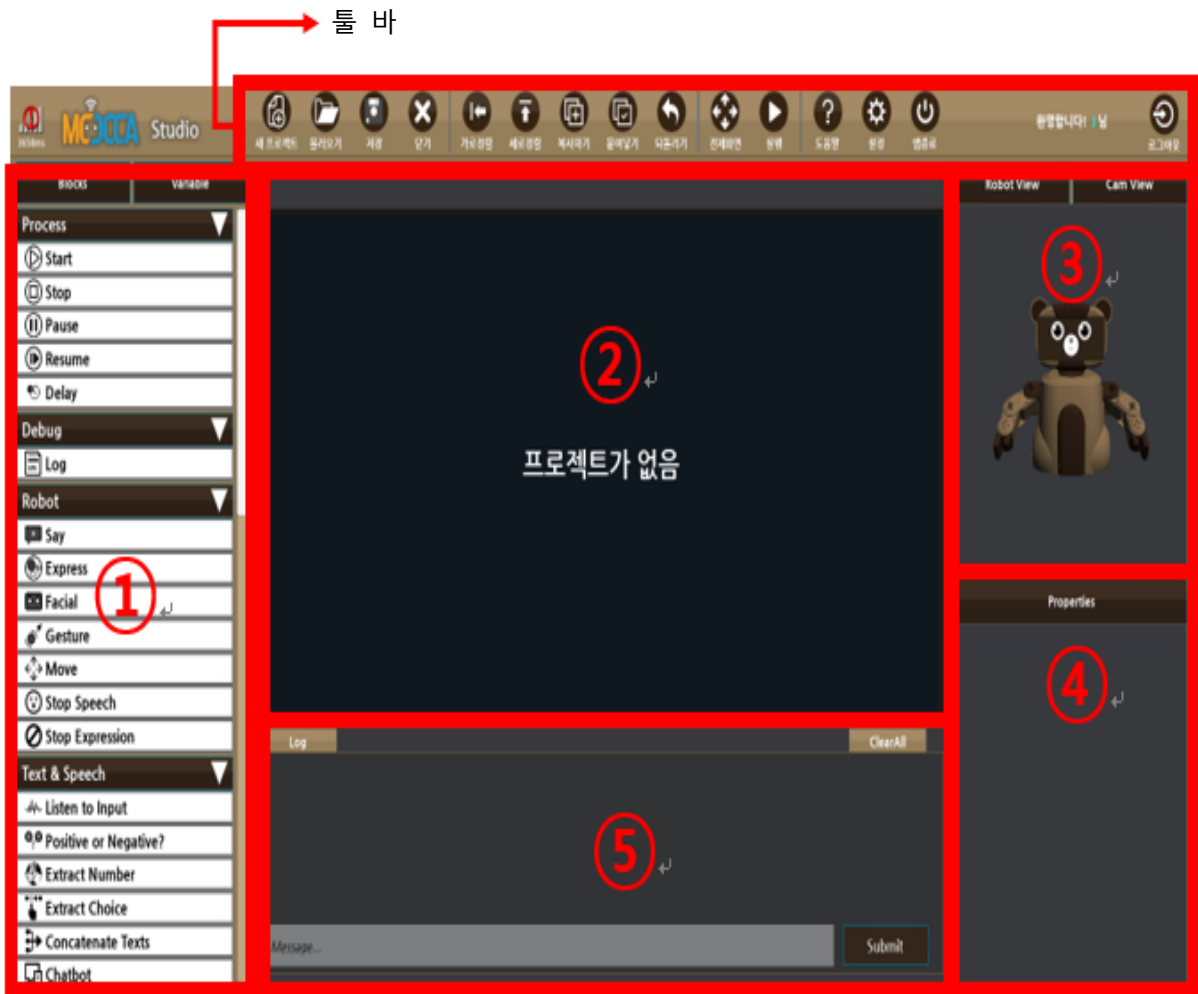
비밀번호

회원가입

여기까지 따라오셨다면 모카 스튜디오로 로봇 콘텐츠를 제작하기 위한 모든 준비가 끝났습니다.

4. 프로그램 UI

MOCCA Studio의 UI는 다음과 같이 툴 바와 5개의 창으로 구분할 수 있습니다.



- ① 블록 & 변수 창(Blocks & Variables Window)
- ② 시나리오 창(Scenario Window)
- ③ 뷰 창(View Window)
- ④ 프로퍼티 창(Property Window)
- ⑤ 로그 창(Log Window)

1. 툴 바

프로그램 사용을 위해 필요한 필요한 도구들이 나열되어 있습니다. 왼쪽부터 순차적으로 그 기능에 대해 설명하겠습니다.

1) 새 프로젝트

 **프로젝트 생성**

저 자: 정예지

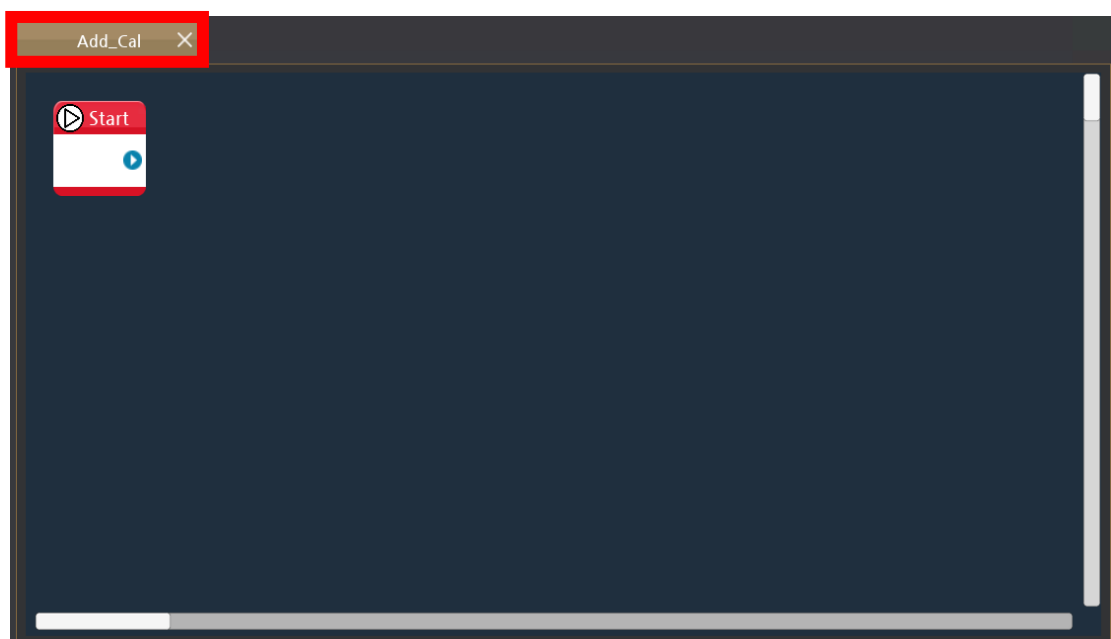
프로젝트 제목: Add_Cal

프로젝트 설명: 덧셈 계산을 위한 계산기 프로젝트입니다.

최초 생성일: 2020-02-20

생성 취소


- 로봇 콘텐츠를 제작하기 위해서는 첫 번째로 프로젝트를 생성해야 합니다.
- 해당 버튼을 클릭하면 위 사진이 팝업 창으로 뜹니다. 사용자 자신만의 프로젝트 생성을 위해 프로젝트 제목, 설명은 필수로 입력해야 합니다.
- 생성 버튼을 클릭하면 아래 사진(▲그림 1)과 같이 시나리오창(②)에 프로젝트가 새 탭으로 열립니다










▲그림 1

2) 불러오기

● 내 프로젝트

 프로젝트 불러오기

제목

제 목	공 유	삭 제	작성일
 OXQuiz_with CamerAPI	<input checked="" type="checkbox"/>	<input type="button" value="삭제"/>	2020-02-17
 test project	<input type="checkbox"/>	<input type="button" value="삭제"/>	2020-02-14
 updowngame(with userAPI)	<input checked="" type="checkbox"/>	<input type="button" value="삭제"/>	2020-02-06
 UpDownGame(User Question	<input checked="" type="checkbox"/>	<input type="button" value="삭제"/>	2019-11-13
 가위바위보_테스트	<input type="checkbox"/>	<input type="button" value="삭제"/>	2019-11-10
 행맨게임	<input checked="" type="checkbox"/>	<input type="button" value="삭제"/>	2019-11-12
 HandMan	<input type="checkbox"/>	<input type="button" value="삭제"/>	2019-11-11

프로젝트 설명:

- 자신이 저장한 프로젝트 목록을 확인하고, 불러올 수 있습니다. 체크 박스를 통해 해당 프로젝트의 공유 여부를 설정할 수 있고, 삭제 버튼을 통해 프로젝트 삭제가 가능합니다.

● 공유 프로젝트

 프로젝트 불러오기


제목

제 목	저 자	작성일
 SpeedQuiz_합수	김도연	2019-12-09
 SpeedQuiz(블록정리)	장세운	2019-12-09
 SpeedQuiz_수정	김도연	2019-12-09
 SpeedQuiz(복사)	장세운	2019-12-09
 QuizUsingAI	이동훈	2020-01-16
 심심이	한성	2020-01-16
 TestUserApi	이동훈	2020-01-16


프로젝트 설명:

- 다른 사용자들이 공유한 프로젝트 목록을 확인하고, 새 탭으로 불러올 수 있습니다.

- 예제 프로젝트

 프로젝트 불러오기


제목

제 목	저 자	작성일
 DEMO_2019	이동훈	2019-12-11
 평가1 템플릿	이동훈	2019-11-13

프로젝트 설명:

- 대표 예제 프로젝트를 확인하고, 불러올 수 있습니다.

3) 저장

 프로젝트 저장 ✕

저 자: 정예지

프로젝트 제목:

프로젝트 설명:

최초 생성일:

최종 수정일:

웹 공유: ☐ 프로젝트 공유하지 않음

- 프로젝트 저장을 위해 해당 버튼을 클릭하면 위와 같은 팝업창이 뜹니다.

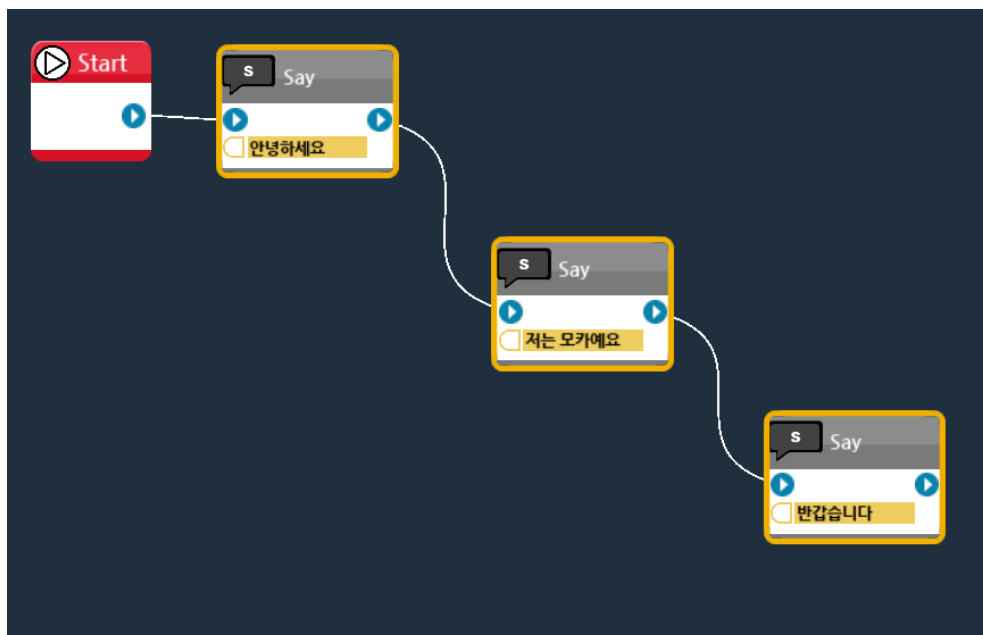
- 프로젝트의 제목은 우측의 변경 버튼을 통해 수정할 수 있고, 설명 또한 수정할 수 있습니다.
- 웹 공유의 체크 박스를 통해 프로젝트 공유 여부를 설정할 수 있습니다.
- 웹 공유 체크를 하지 않고 저장 버튼을 클릭하면 해당 프로젝트가 '내 프로젝트'에 추가됩니다. 체크를 하면 '내 프로젝트'와 '공유 프로젝트'에 추가됩니다.
- 다른 사용자의 공유 프로젝트를 수정하였다면 프로젝트 제목을 수정한 후 복사본 저장을 클릭합니다. 이를 통해 수정된 내용의 프로젝트가 수정된 제목으로 '내 프로젝트'에 추가됩니다.

4) 닫기

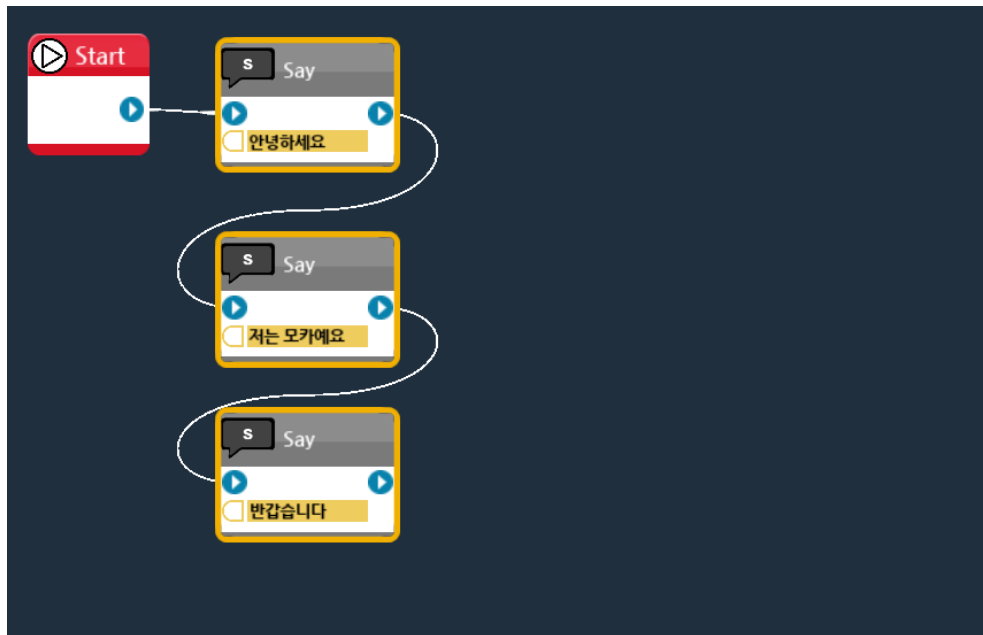
- 현재 시나리오 창(②)에 열려 있는 프로젝트를 닫습니다.
- 프로젝트를 저장하지 않고, '닫기' 버튼을 클릭하면 현재 프로젝트의 블록들이 자동으로 저장되지 않습니다. 이 점을 유의하시기 바랍니다.

5) 가로정렬

- 원하는 블록들을 드래그로 선택한 후 해당 버튼을 클릭하면 각 블록의 좌우가 서로 같아지도록 정렬합니다.



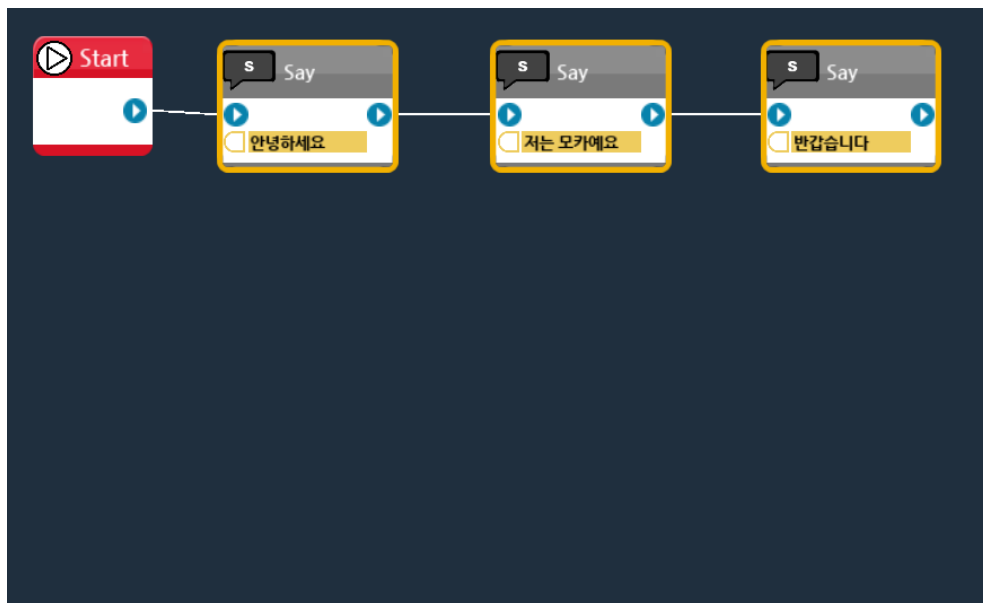
▲그림 2. 변경 전



▲그림 3. 변경 후

6) 세로정렬

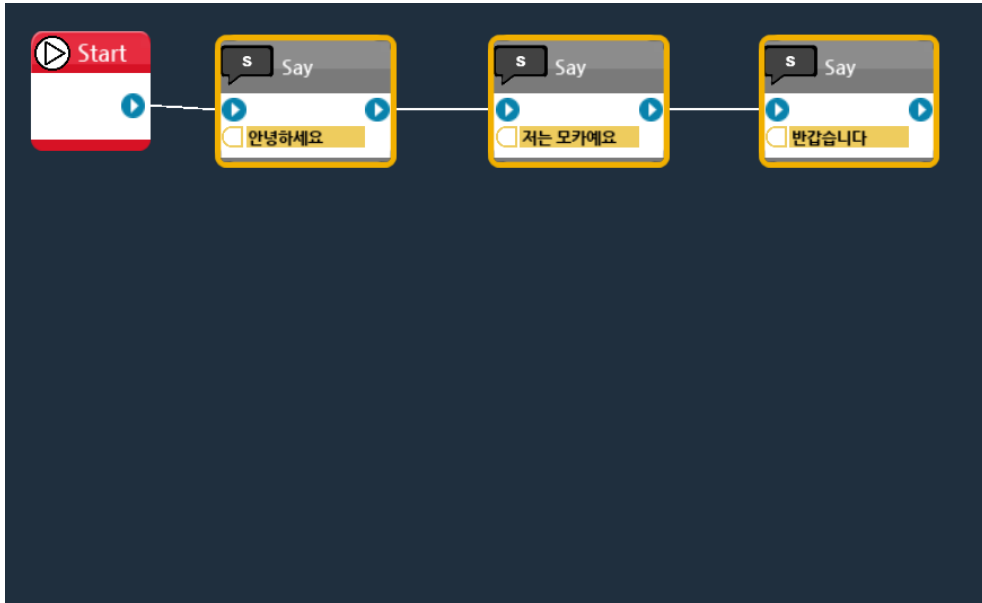
- 원하는 블록들을 드래그로 선택한 후 해당 버튼을 클릭하면 각 블록의 상하가 서로 같아지도록 정렬합니다. 아래 사진은 '▲그림 1' 상태에서 '세로정렬' 버튼을 클릭한 결과입니다.



▲그림 4. 변경 후

7) 복사하기

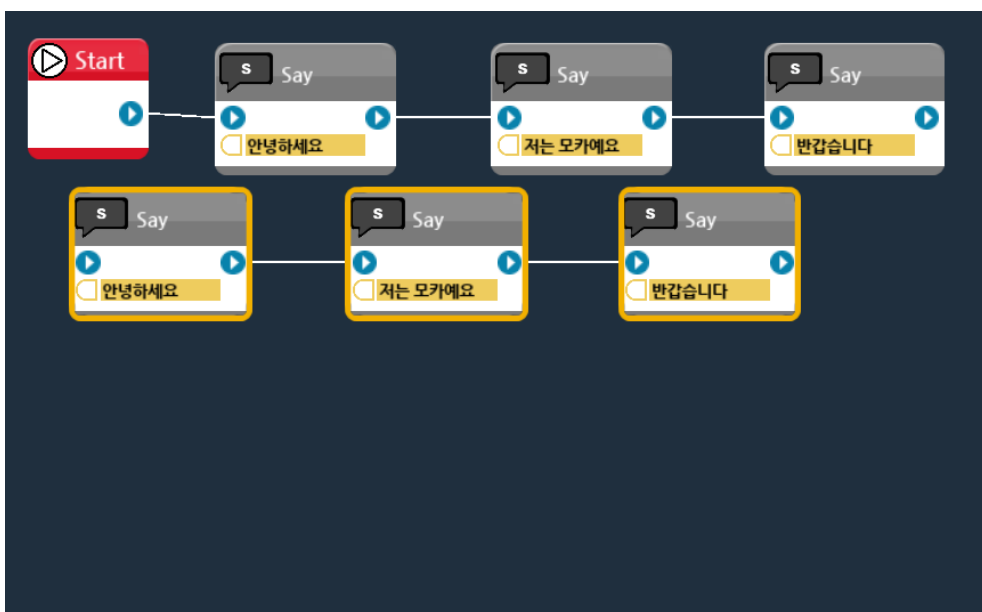
- 원하는 블록들을 드래그를 통해 선택한 후 해당 버튼을 클릭하면 복사됩니다.
- Ctrl+C 단축키를 통해서도 사용할 수 있습니다.



▲그림 5. 복사

8) 붙여넣기

- 해당 버튼 클릭을 통해 복사한 블록들을 시나리오 창(②)에 붙여 넣습니다.
- Ctrl+V 단축키를 통해서도 사용할 수 있습니다.



▲그림 6. 붙여넣기 후

9) 되돌리기

- 해당 버튼 클릭을 통해 작업한 내용을 가장 최근의 이전 상태로 되돌립니다.
- Ctrl+Z 단축키를 통해서도 사용할 수 있습니다.
- '▲그림 5. 붙여넣기 후' 상태에서 '되돌리기' 버튼을 클릭하면 '▲그림 4. 복사'의 상태가 됩니다.

10) 전체화면

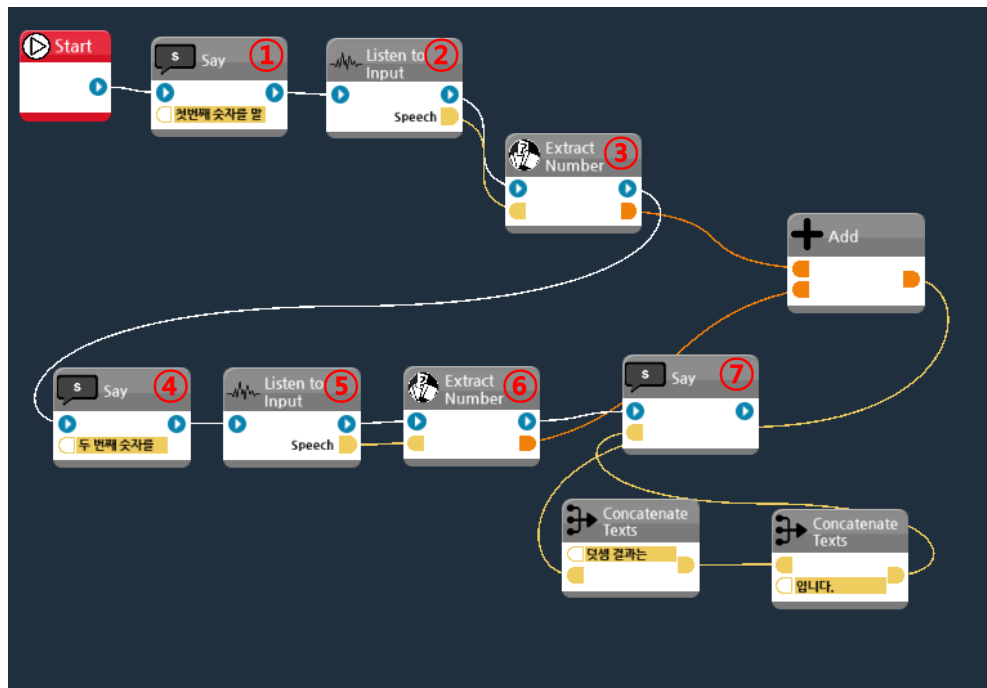


- 해당 버튼 클릭을 통해 로봇 창(③)을 전체 화면으로 전환하여 볼 수 있습니다.
- 상단에는 3개의 버튼이 있습니다.
 - ✓ 화면최소화: '전체화면' 버튼을 클릭하기 이전 상태로 돌아갑니다.
 - ✓ 실행: 현재 프로젝트를 실행합니다.
 - ✓ 멈춤: 프로젝트의 실행을 정지합니다. 정지한 후 해당 버튼은 재실행으로 바뀝니다. 또 한 번의 클릭으로 프로젝트를 다시 실행할 수 있습니다.
 - ✓ 프로젝트 실행 중 'Listen to Input' 블록 순서가 되면 하단의 입력창을 통해 입력한 후 하단 우측의 'Submit' 버튼 클릭합니다. 이를 통해 블록의 입력 값을 넣어줍니다.

11) 실행

- 시나리오 창(②)에 배치된 블록들이 실행됩니다.
- MOCCA Studio에서 블록들은 Flow 기반으로 실행되기 때문에 '▲그림 7'과 같이 Start 노

드부터 Flow 선으로 연결된 순서대로 실행됩니다.

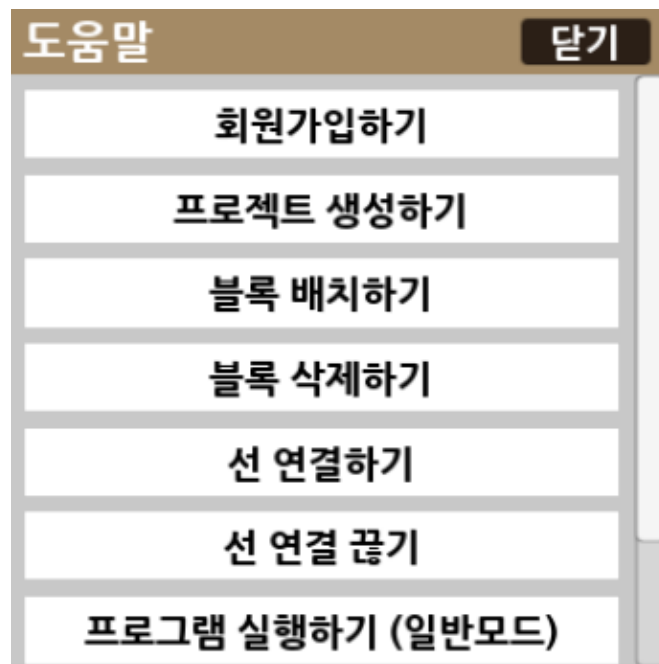


▲그림 7

- Flow가 없는 블록들은 연결된 블록(Flow가 있는 블록)이 실행될 때 같이 실행됩니다.

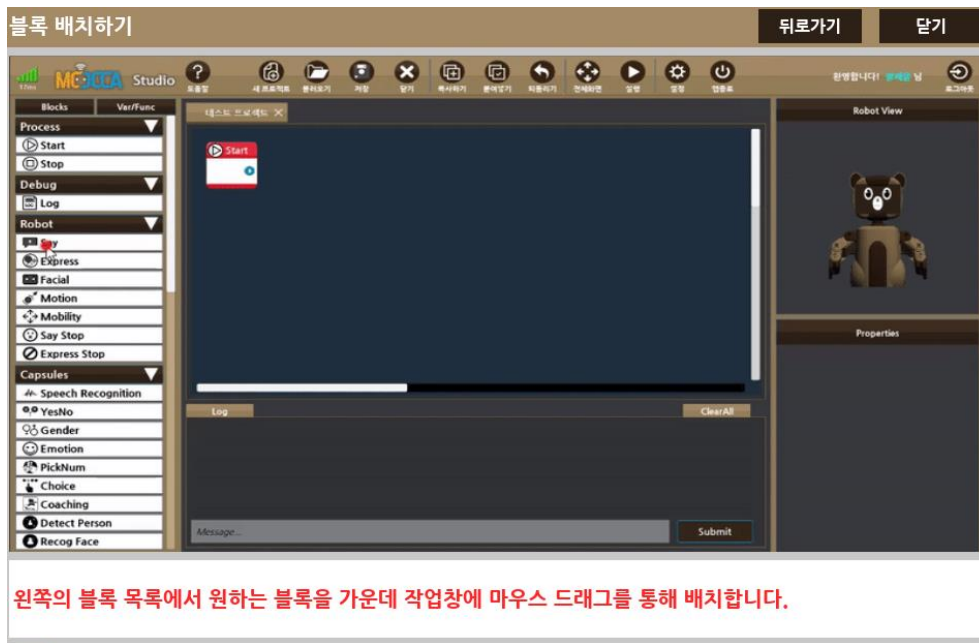
12) 도움말

- MOCCA Studio 사용에 관한 도움말들을 영상으로 확인할 수 있습니다.



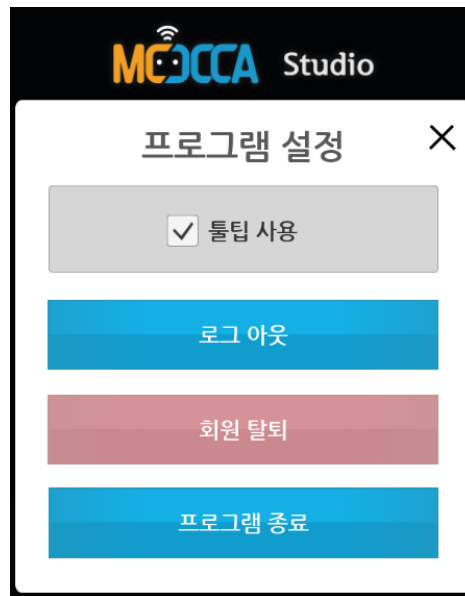
- 우측 상단의 닫기 버튼을 클릭하면 도움말을 닫을 수 있습니다.

- 목록에서 원하는 항목을 누르면 영상과 설명을 열람할 수 있습니다.

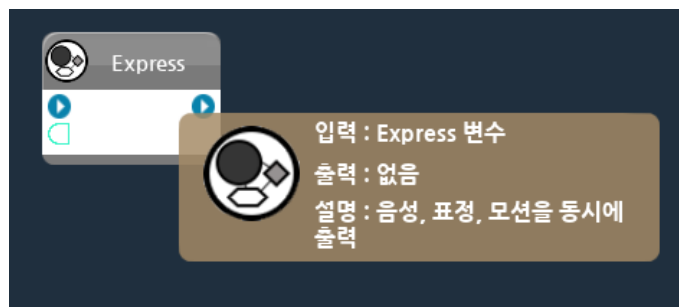


- 뒤로가기 버튼을 클릭하면 도움말 목록으로 돌아갑니다.
- 닫기 버튼을 클릭하여 도움말을 닫을 수 있습니다.
- 도움말 목록
 - ✓ 회원가입하기
 - ✓ 프로젝트 생성하기
 - ✓ 블록 배치하기
 - ✓ 블록 삭제하기
 - ✓ 선 연결하기
 - ✓ 선 연결 끊기
 - ✓ 프로그램 실행하기 (일반모드)
 - ✓ 프로그램 실행하기(전체화면 모드)
 - ✓ 프로그램 종료하기

13) 설정



- ✓ 툴팁 사용: 체크 박스를 해제하면 툴팁 사용 기능을 끄고, 다시 체크하면 툴팁 사용 기능을 사용할 수 있습니다.



- ◆ 툴팁 사용 시 각 블록에 마우스 커서를 가져다 대면 약 2초 후에 위 그림과 같은 간단한 블록 설명이 나타납니다.
- ✓ 로그 아웃: 현재 로그인 된 계정을 로그아웃 합니다.
- ✓ 회원 탈퇴: 현재 로그인 된 계정을 MOCCA Studio에서 회원 탈퇴합니다.
- ✓ 프로그램 종료: MOCCA Studio를 종료합니다.

14) 앱종료

- MOCCA Studio를 종료합니다.

15) 로그아웃

- 현재 로그인 된 계정을 로그아웃 합니다.

2. 블록 & 변수 창(Blocks & Variables Window)

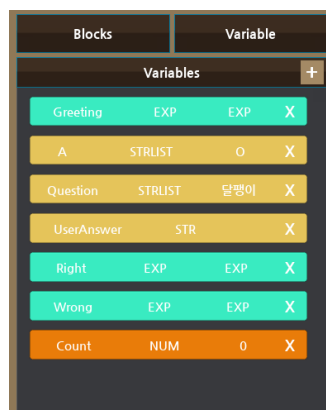
시나리오 작성에 필요한 블록들과 변수를 생성하는 창입니다.

1) Blocks

- Blocks 를 클릭하면 블록의 목록들이 나타납니다. 블록들은 카테고리 별로 정렬되어 있으며 드래그 하여 시나리오 창에 드롭 하면 블록을 생성할 수 있습니다.
- 블록 카테고리
 - ① Process: 프로그램(시나리오)의 실행과 관련된 블록
 - ② Debug: 디버그 블록
 - ③ Robot: 로봇의 표현, 주행과 관련된 블록
 - ④ Text & Speech: 자연어와 관련된 인공지능 블록과 User API 블록
 - ⑤ Image: 이미지에서 정보를 인식하는 블록
 - ⑥ Flow Control: 조건문 블록
 - ⑦ Variables: List 관련 블록
 - ⑧ Math: 수학 연산자 블록
 - ⑨ Logic: 논리 연산자 블록
 - ⑩ Comparatives: 비교 연산자 블록

2) Variables

- Variables 를 클릭하면 변수를 생성하고 관리할 수 있습니다.

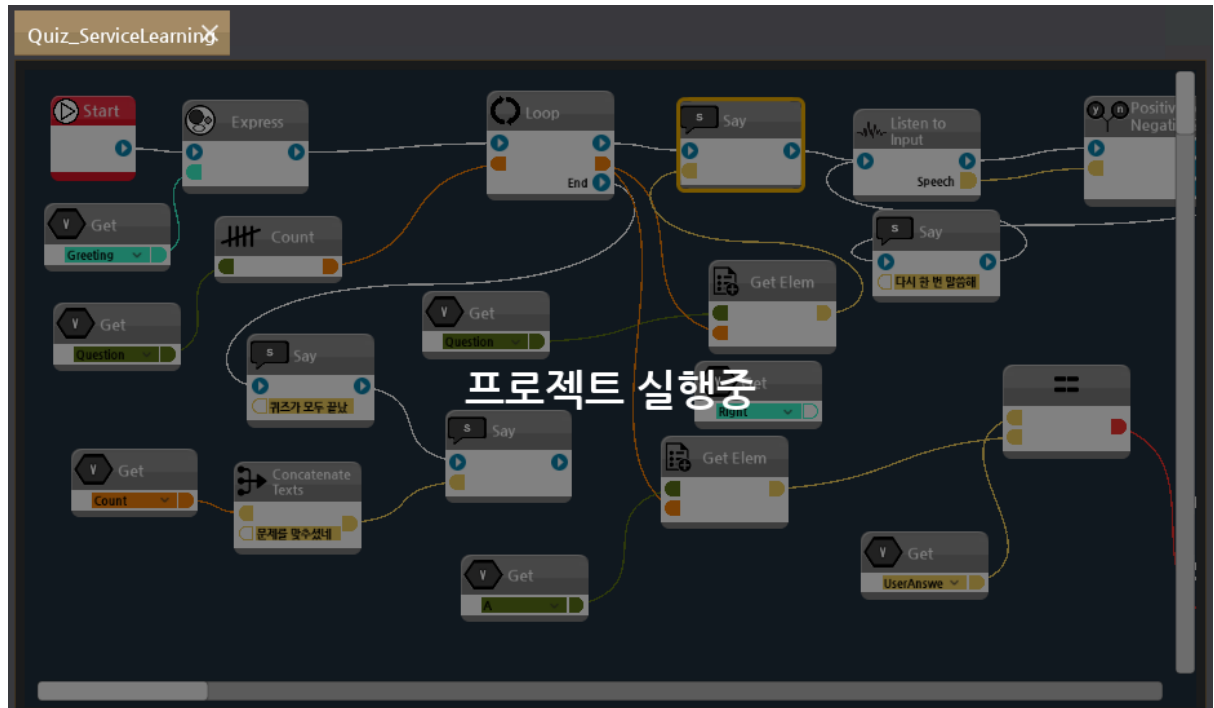


Blocks		Variable	
Variables +			
Greeting	EXP	EXP	X
A	STRLIST	0	X
Question	STRLIST	달팽이	X
UserAnswer	STR		X
Right	EXP	EXP	X
Wrong	EXP	EXP	X
Count	NUM	0	X

- 위 그림처럼 현재 생성된 변수 목록들을 열람할 수 있습니다.
- 변수를 생성하는 부분은 [목차 5]에서 자세히 설명하도록 하겠습니다.

3. 시나리오 창(Scenario Window)

현재 작성 중인 프로젝트를 편집할 수 있는 창입니다. 블록 & 변수 창에서 드래그 앤 드롭으로 블록을 배치하여 시나리오를 작성할 수 있습니다. 프로젝트 실행 시 '프로젝트 실행중'이라는 메시지와 함께 현재 작동 중인 블록이 노란 박스로 강조되어 확인할 수 있습니다.



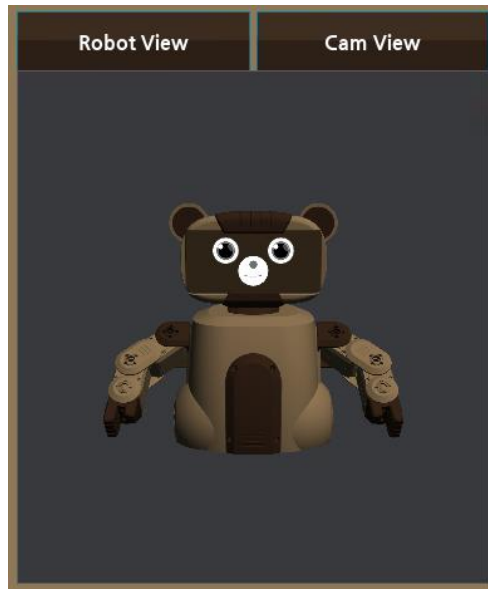
상단 탭에 현재 편집, 실행 중인 프로젝트의 이름이 표시됩니다.

마우스 휠로 시나리오 창을 확대, 축소할 수 있으며, 마우스 우클릭 한 채로 드래그 하면 시나리오 창 안에서 이동할 수 있습니다.

4. 뷰 창(View Window)

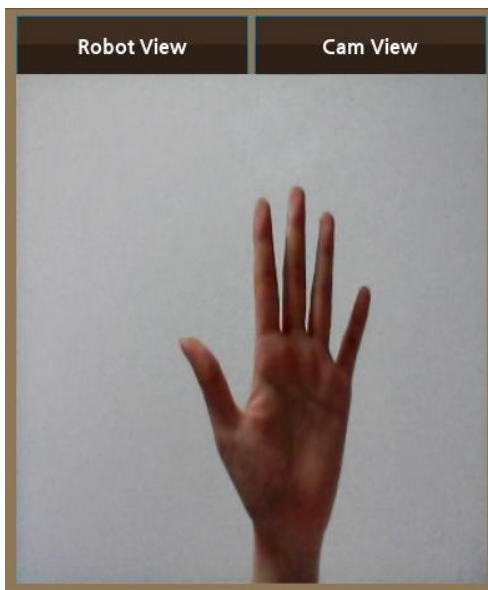
로봇과 카메라로 촬영하는 영상을 확인할 수 있는 창입니다.

1) Robot View



- Robot View를 클릭하면 로봇의 모습과 동작을 확인할 수 있습니다.

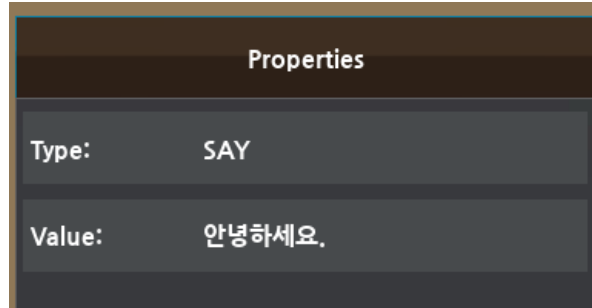
2) Cam View



- Cam View를 클릭하면 카메라로 촬영 중인 영상을 확인할 수 있습니다. 단, 카메라를 사용하는 블록이 실행되어야 활성화되며, 카메라를 사용하지 않을 때는 하얀 화면이 나타납니다.

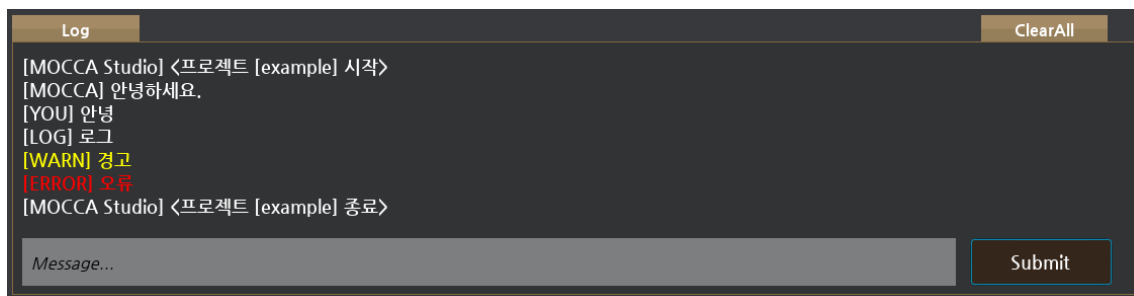
5. 프로퍼티 창(Property Window)

프로퍼티 창은 각 블록의 세부 정보를 보여주는 창입니다. 시나리오 창에 올라온 블록을 클릭했을 때 그 블록의 Type과 Value를 보여줍니다. 이때 Value는 시나리오 창 뿐만 아니라 프로퍼티 창에서 수정하는 것 또한 가능합니다.



6. 로그 창 (Log Window)

로그 창은 프로그램 실행 시 진행상황 및 로그 정보를 보여주는 창입니다. 로그 출력 창, 입력 창, Submit 버튼, ClearAll 버튼으로 구성되어 있습니다.



1) 로그 출력 창

- 상단에 위치한 로그 출력 창은 각종 알림 메시지와 모카의 말, 사용자가 입력한 결과를 보여주는 창입니다. 각 줄의 앞부분에 메시지 종류를 구분하는 표시가 들어가고, 그 표시로는 다음 표의 내용과 같습니다.

[MOCCA Studio]	시스템에서 알림 메시지
[MOCCA]	모카의 말
[YOU]	사용자의 입력
[LOG]	로그 메시지
[WARN]	경고 메시지
[ERROR]	오류 메시지

[알림]	모카스튜디오 사용시간 알림
------	----------------

2) 입력 창, Submit 버튼

- 하단에 위치한 입력 창은 Listen to Input처럼 사용자의 입력을 요구할 때 활성화 되고, 입력 후 Submit 버튼을 클릭하면 로그 출력 창에 [YOU] 표시와 함께 출력됩니다.

3) ClearAll 버튼

- 우측 상단의 ClearAll 버튼을 클릭하면 로그 출력 창의 내용이 모두 지워집니다.

5. 변수

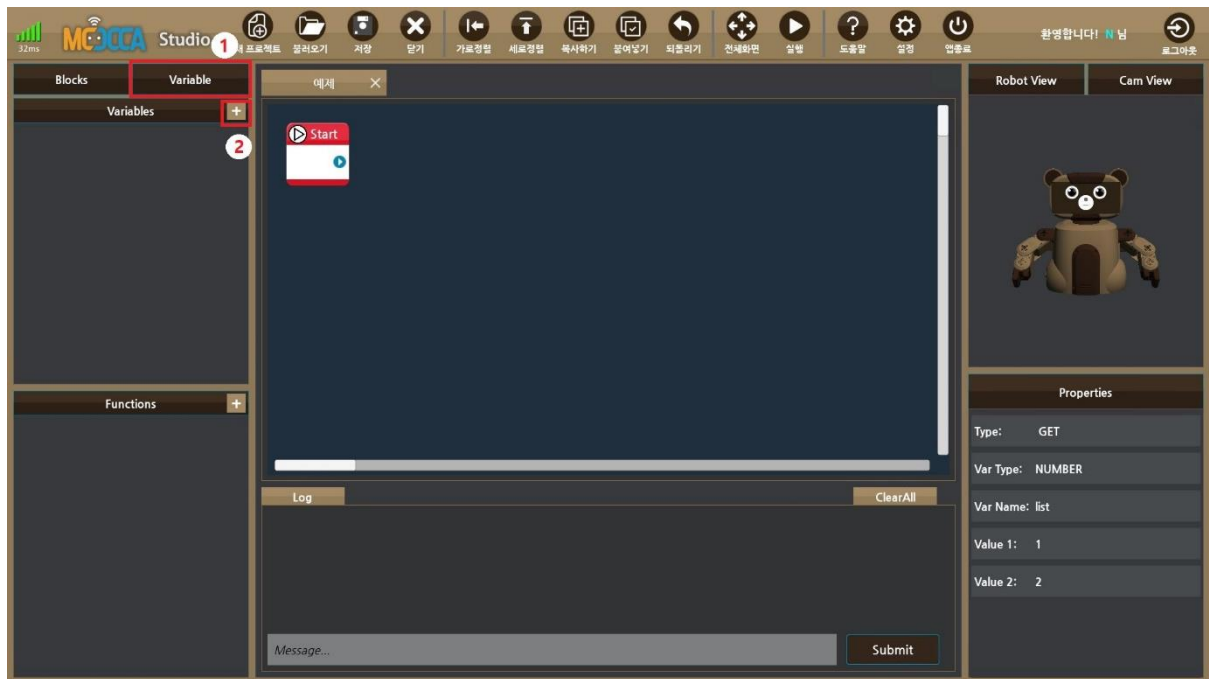
1) 변수 목록



Mocca Studio의 변수 목록은 총 9가지입니다. 각 변수마다 고유의 색을 가지는 색상코딩을 활용하여 한눈에 알아보기 쉽도록 제작되었습니다.

변수형	특징	값(Value)
Bool	참과 거짓	True, False, 1, 0 등
Number	정수 혹은 실수	10, 2.5, 100, -20 등
String	문자열	Hello, 안녕 등
Expression	모카의 말+표정+동작	"안녕"+'행복'표정+'하이'동작
Facial	모카의 표정	'행복', '화남', '쳐다보기' 등
Motion	모카의 동작	'박수', '끄덕이기', '고개돌림' 등
Mobility	모카의 주행 방향	'전진', '후진', '좌회전' 등
List	공통된 변수형을 가지는 복수개의 값	
None	정해지지 않은 경우	

2) 변수 생성 방법



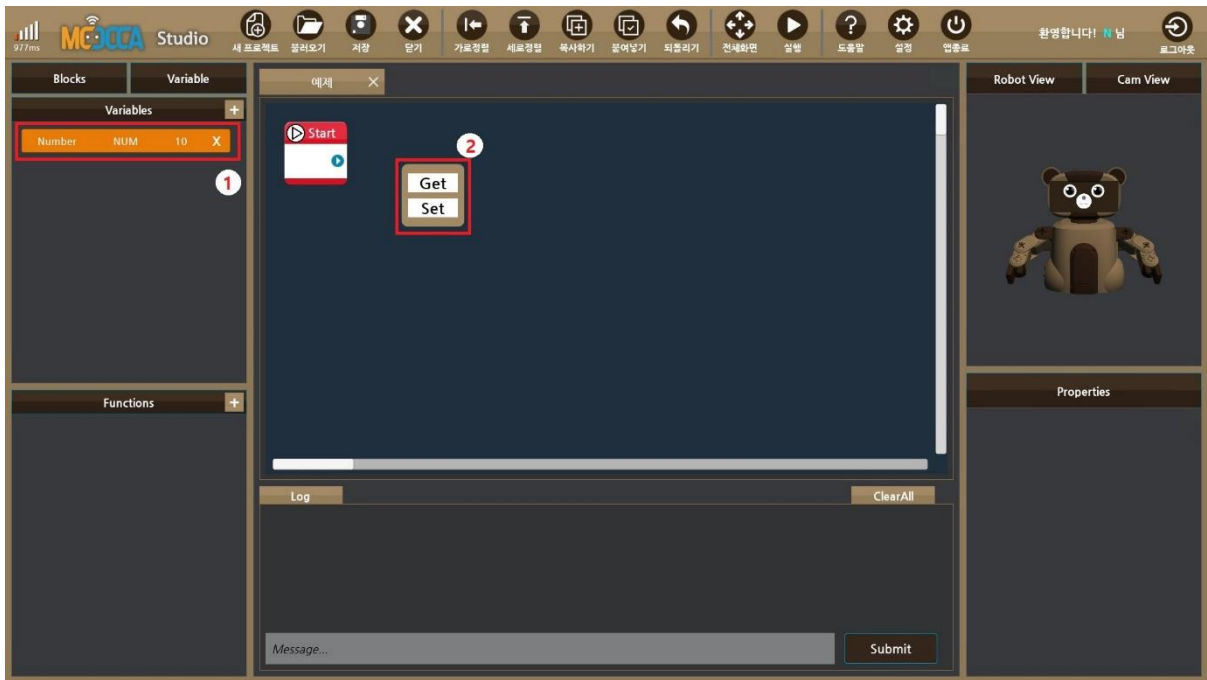
▲ 사진1

Blocks & Variables 창의 Variable(사진1의 1번)을 클릭 후 '+' 버튼(사진1의 2번)을 클릭하면 다음과 같은 창(사진2)을 볼 수 있습니다.



▲ 사진2

'Type' 옆의 Dropdown을 클릭하면 Bool, Number, String, Expression, Facial, Motion, Mobility 등 변수형을 설정할 수 있고, Dropdown 옆의 네모버튼(—)에서 Single 및 List형태를 설정할 수 있습니다. 변수의 속성을 설정한 후 오른쪽 상단의 'OK'버튼을 클릭하면 변수가 생성됩니다.

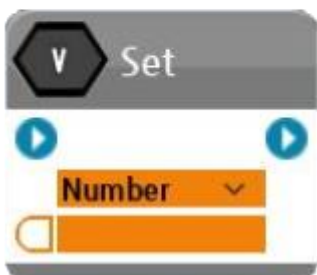


▲ 사진3

생성된 변수는 Variables창(사진3의 1)에서 확인할 수 있습니다. 변수를 클릭한 후 시나리오 창에 드래그하면 변수의 Get, Set여부(사진3의 2)를 물어봅니다.



· Get : 변수의 값을 불러오기





· Set : 변수의 값을 변경하기

위와 같은 방식으로 다양하게 변수를 만들어 사용해 보세요.

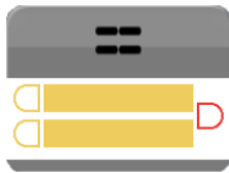
6. 블록의 세부 구조

블록은 flow 와 입력, 출력으로 구성됩니다. flow 는 실행 흐름으로써 왼쪽에서 오른쪽 방향으로 진행합니다. 블록의 왼쪽에서 블록의 시작 흐름과 변수 등을 입력받고, 오른쪽에서 다음 블록으로 이동하는 흐름과 변수를 출력합니다.

- 아이콘

-  : flow
-  : 입력 변수
-  : 출력 변수

같은 데이터 타입의 변수가 연결되면  이처럼 아이콘 내부의 색상이 채워집니다.



왼쪽의 블록처럼 flow 가 없는 블록은 출력 변수가 연결된 블록이 실행될 때 함께 실행됩니다.

- 블록과 블록 간의 연결선을 삭제하고 싶다면, Alt 를 누른 채 선이나 해당 변수의 아이콘을 클릭합니다.
- 변수는 꼭 다른 블록에서 연결하여 입력하지 않고, 블록에서 직접 입력하거나 블록 클릭 후 Property 창의 Value 에서 입력할 수 있습니다.
- 변수에 저장할 데이터를 입력할 때, Unity Engine 의 버그로 인하여 커서의 위치와 입력한 글자의 위치가 달라질 수 있습니다. 따라서 한글로 입력한 데이터를 수정할 때에는 데이터 전체를 모두 지운 후 다시 입력하는 방법을 추천합니다.
- 블록을 삭제하고 싶다면 삭제하려는 블록을 클릭하여 선택한 후 delete 를 눌러 삭제할 수 있습니다.

7. 블록 기능 상세 설명

1. Process



Start

기능: 프로세스를 시작합니다.

입력: 없음

출력: 없음

특징: 블록은 한 프로세스의 제일 앞에 위치합니다. Start에서부터 흐르는 flow 순서대로 프로세스가 진행됩니다.



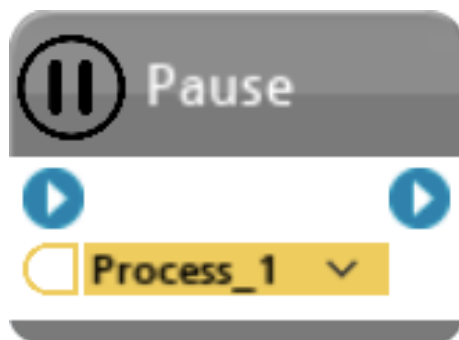
Stop

기능: 프로세스를 종료합니다.

입력: 없음

출력: 없음

특징: 진행 중인 프로세스의 flow에 연결되면 해당 프로세스를 종료합니다.



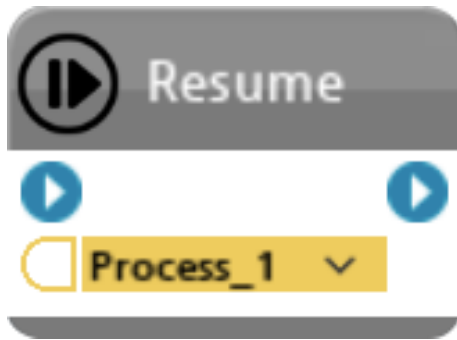
Pause

기능: 프로세스를 정지합니다.

입력: 프로세스

출력: 없음

특징: 드롭다운에서 선택된 프로세스를 정지합니다.



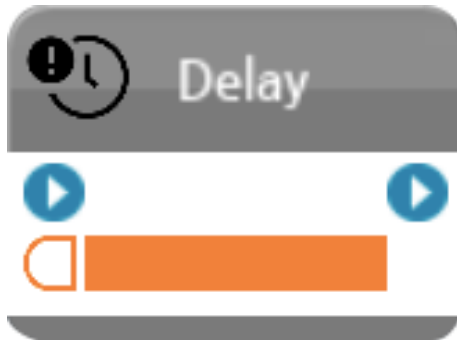
Resume

기능: 프로세스를 다시 실행합니다.

입력: 프로세스

출력: 없음

특징: 드롭다운에서 선택된 프로세스를 다시 실행합니다.



Delay

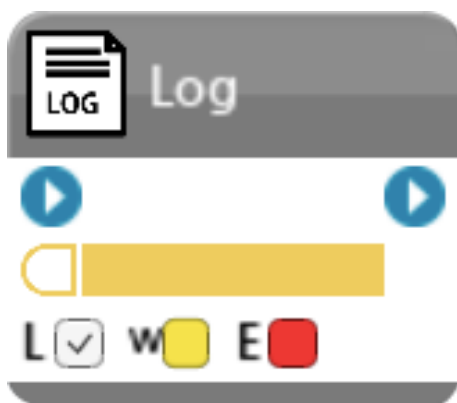
기능: 입력 받은 숫자 값(초)만큼 연결된 프로세스의 진행을 기다립니다.

입력: Number

출력: 없음

특징: 없음

2. Debug



Log

기능: 입력 받은 문자열을 Log창에 출력합니다.

입력: String

출력: 없음

특징: L (Log) / W (Warning) / E (Error) 중 체크한 것에 따라 Log창에

(하얀색 글씨) [Log] 입력 받은 문자열

(노란색 글씨) [WARN] 입력 받은 문자열

(빨간색 글씨) [ERROR] 입력 받은 문자열을 출력합니다.

3. Robot



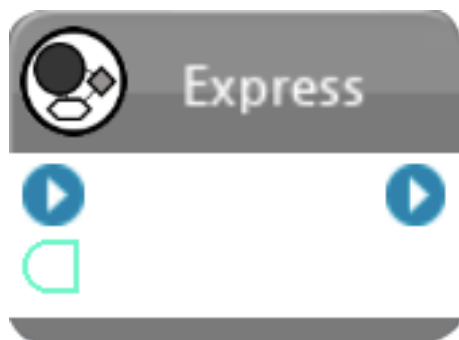
Say

기능: 입력된 값을 음성 출력합니다.

입력: String

출력: 없음

특징: 영어와 한국어 뿐만 아니라 숫자도 입력가능 합니다. 입력된 값에 따라 로봇의 입 모양이 움직이며 로봇의 표정과 동작 또한 입력된 값에 해당되는 감정에 따라 랜덤으로 지정됩니다.



Express

기능: 음성, 표정, 모션을 사용자가 직접 선택할 수 있고, 모두 동시에 출력됩니다.

입력: Express 변수

출력: 없음

특징: Block & Variable 창에서 Express 변수를 미리 생성해주어야 합니다.



Facial

기능: 입력된 값을 로봇의 표정으로 제어합니다.

입력: Facial 변수

출력: 없음

특징: Facial 변수를 따로 생성하여 입력으로 연결 해주거나 Facial 블록의 드롭다운에서 선택하여 로봇의 표정을 제어할 수 있습니다.



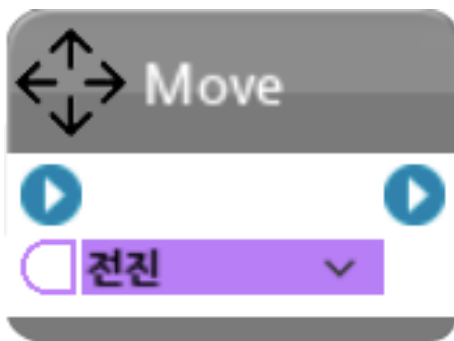
Gesture

기능: 입력된 값을 로봇의 동작으로 제어합니다.

입력: Motion 변수

출력: 없음

특징: Motion 변수를 따로 생성하여 입력으로 연결해주거나 Gesture 블록의 드롭다운에서 선택하여 로봇의 동작을 제어할 수 있습니다.



Move

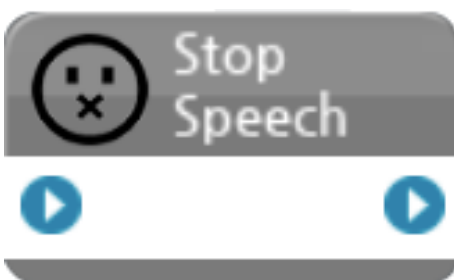
기능: 입력된 방향 정보에 따라 실물 로봇을 구동합니다.

입력: Mobility 변수

출력: 없음

특징: Mobility 변수를 따로 생성하여 입력으로 연결해주거나 Move 블록의 드롭다운에서 선택하여 로봇을 구동합니다.

해당 블록은 시뮬레이션 로봇에서 실행 결과를 확인하실 수 없습니다.



Stop Speech

기능: 음성 출력을 중단시킵니다.

입력: 없음

출력: 없음

특징: 없음



Stop Expression

기능: Express를 중단시킵니다.

입력: 없음

출력: 없음

특징: 없음

4. Text & Speech



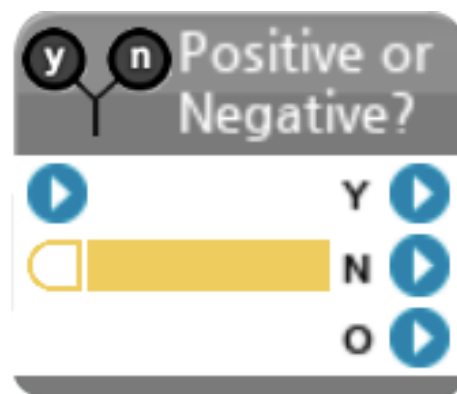
Listen to Input

기능: 사용자로부터 입력 받은 문자열을 출력합니다.

입력: 없음

출력: String

특징: Log창에서 타이핑한 입력 값 또는 모바일 앱을 통한 음성을 문자열로 출력합니다.



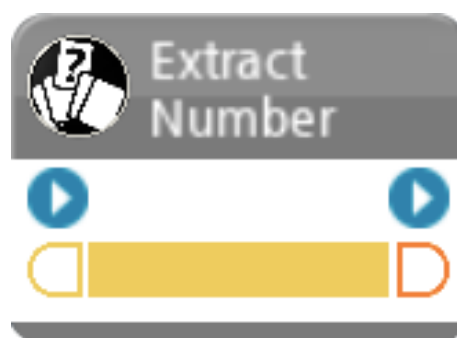
Positive or Negative?

기능: 입력 받은 문자열을 긍정, 부정, 모호함으로 구분합니다.

입력: String

출력: 없음

특징: Y(Yes)의 대표적인 말-응, 맞아, 그래 등
N(No)의 대표적인 말-아니, 틀려 등
O(Obscure)의 대표적인 말-글쎄, 몰라 등



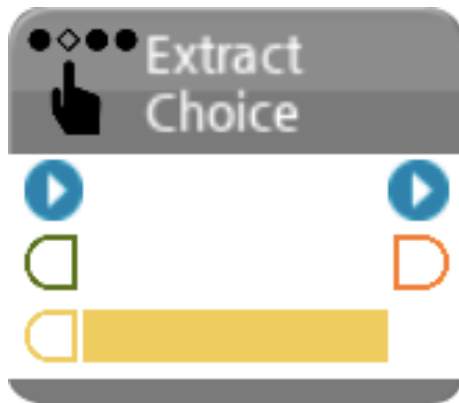
Extract Number

기능: 입력 받은 문자열로부터 숫자를 판단하여 출력합니다.

입력: String

출력: Number

특징: 일, 첫째, 처음 등의 문자를 1로 판단하여 숫자로 출력합니다.



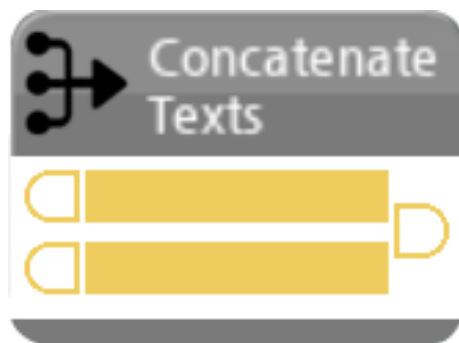
Extract Choice

기능: 입력된 문자열 리스트에서 어떤 문자열을 선택하였는지 판단합니다.

입력: String List, String

출력: Number

특징: 입력 받은 문자열에서 리스트에 있는 문자열 또는 '-번째' 정보를 판단하여 리스트에서 해당 정보의 인덱스 번호를 출력합니다.



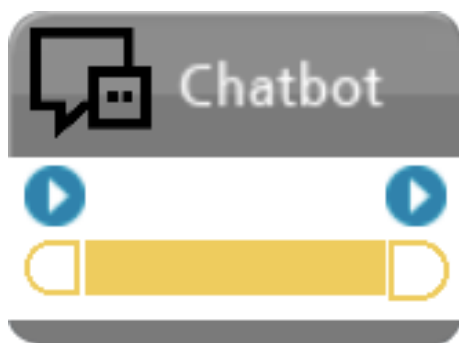
Concatenate Texts

기능: 입력 받은 두 문자열을 하나의 문자열로 변환합니다.

입력: String, String

출력: String

특징: 없음



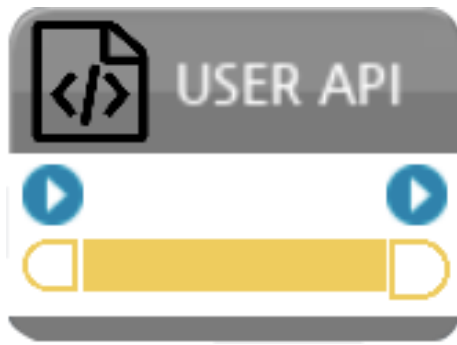
Chatbot

기능: 입력 받은 문자열을 심심이에게 전달함으로써 심심이의 답변을 출력합니다.

입력: String

출력: String

특징: 프로그램 외부에 있는 심심이와 대화를 할 수 있는 블록입니다.



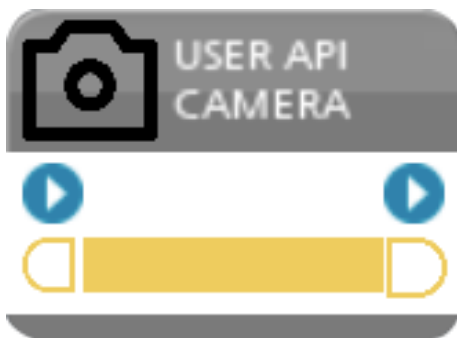
USER API

기능: 입력 받은 서버 주소에서 반환되는 값을 출력합니다.

입력: String(URL)

출력: String

특징: 사용자가 직접 서버를 제작한 후 연결하여 사용할 수 있습니다.



USER API CAMERA

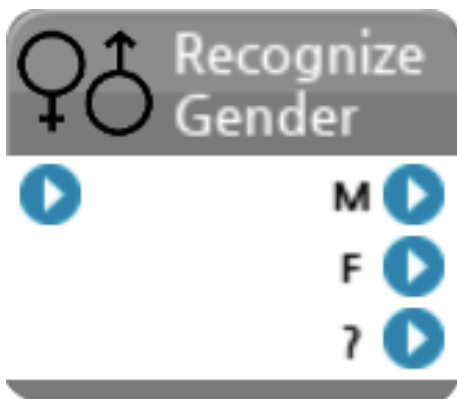
기능: 입력 받은 서버에 웹 캠 이미지를 전송하여 결과를 전달받습니다.

입력: String

출력: String

특징: 모카 스튜디오 이외에서 웹 캠을 사용하면 프로그램 내부에서 웹 캠 사용이 불가능합니다.

5. Image



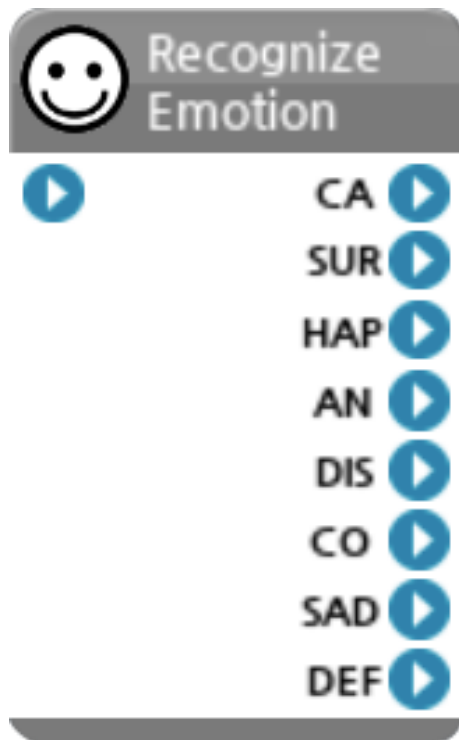
Recognize Gender

기능: 사용자의 성별을 판단합니다.

입력: 없음

출력: 없음

특징: 카메라를 통해 얼굴인식을 하여 남성이라 판단하였을 경우 'M'(Male)으로, 여성일 경우 'F'(Female), 판단하지 못했을 경우 '?'으로 Flow가 분기됩니다.



Recognize Emotion

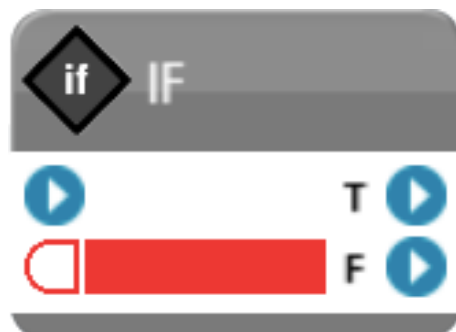
기능: 사용자의 현재 표정을 판단합니다.

입력: 없음

출력: 없음

특징: 카메라를 통해 얼굴인식을 하여 사용자의 표정을 판단합니다. 사용자의 표정을 CA(Calm), SUR(Surprised), HAP(Happy), AN(Angry), DIS(Disgusted), CON(Confused), SAD(Sad), DEF(Default) 의 총 8가지의 Flow로 분기됩니다.

6. Flow Control



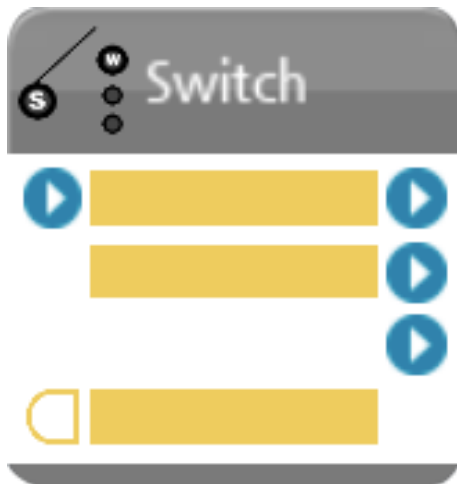
IF

기능: 입력된 값에 따라 참, 거짓으로 분기합니다.

입력: Bool

출력: 없음

특징: 입력 창에 True, 1 또는 False, 0을 직접 입력할 수 있습니다.



Switch

기능: 입력된 값과 일치하는 조건으로 분기합니다.

입력: String

출력: 없음

특징: 조건의 개수는 Properties 창에서 + 버튼을 통해 조절할 수 있습니다. 한글과 영어뿐만 아니라 숫자, Number 변수로 입력 가능합니다.



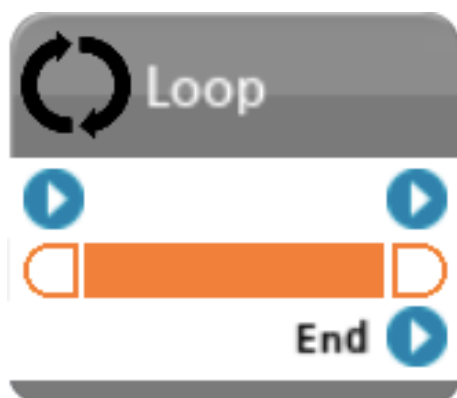
While

기능: 입력된 부울 값이 참(True, 1)이면 반복, 거짓(False, 0)이면 End로 분기합니다.

입력: Bool

출력: 없음

특징: Flow에 연결된 블록들에서 다음 Flow가 연결되지 않은 경우 암묵적으로 While 블록으로 돌아온다.



Loop

기능: 입력된 값만큼 반복 후 End로 분기합니다.

입력: Number(반복 횟수)

출력: Number(몇 번째 반복인지)

특징: 출력되는 Index 값은 0부터 시작합니다. Flow에 연결된 블록들에서 다음 Flow가 연결되지 않은 경우 암묵적으로 Loop 블록으로 돌아온다.



Match Texts

기능: 입력된 값을 포함하는 조건으로 분기합니다.

입력: String

출력: 없음

특징: 조건의 개수는 Properties 창에서 '+' 버튼을 통해 조절할 수 있습니다. 일치하는 조건이 없으면 마지막 Flow로 분기됩니다. 한글과 영어뿐만 아니라 숫자, Number 변수로 입력 가능합니다.

7. Variables



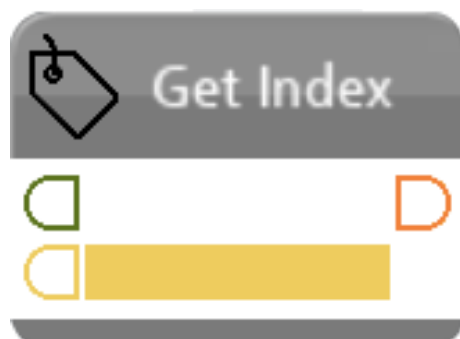
Count

기능: List의 길이(크기)를 출력합니다.

입력: List

출력: Number

특징: 없음



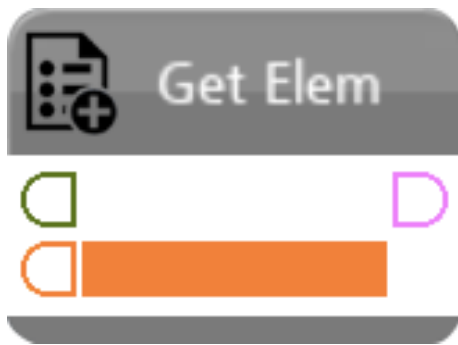
Get Index

기능: List에서 입력된 String 값에 해당하는 index를 출력합니다.

입력: List, String

출력: Number

특징: 없음



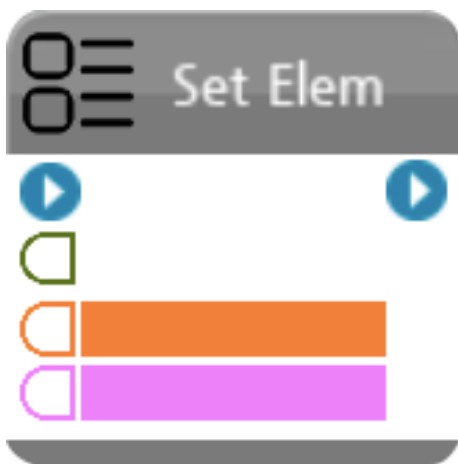
Get Element

기능: List의 index에 해당하는 값을 출력합니다.

입력: List, Number

출력: List의 Data Type

특징: 리스트를 입력으로 연결하면 세번째 입력 값의 데이터형은 리스트와 동일한 데이터형으로 변환됩니다.



Set Element

기능: List에 입력된 index 요소 값을 수정합니다.

입력: List, Number, Data

출력: 없음

특징: 리스트를 입력으로 연결하면 세번째 입력 값의 데이터형은 리스트와 동일한 데이터형으로 변환됩니다.



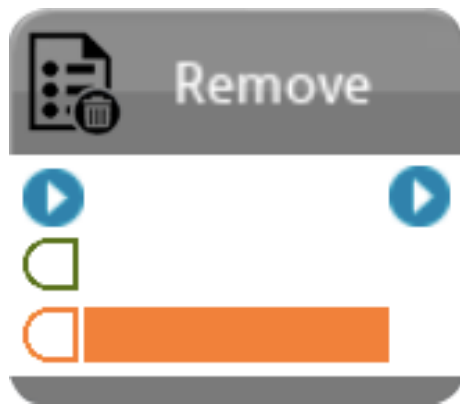
Insert

기능: List에 입력된 index 위치에 값을 삽입합니다.

입력: List, Number, Data

출력: 없음

특징: 리스트를 입력으로 연결하면 세번째 입력 값의 데이터형은 리스트와 동일한 데이터형으로 변환됩니다. 이미 존재하는 index에 값을 추가하면 'Set Element'와 동일한 역할을 수행합니다.



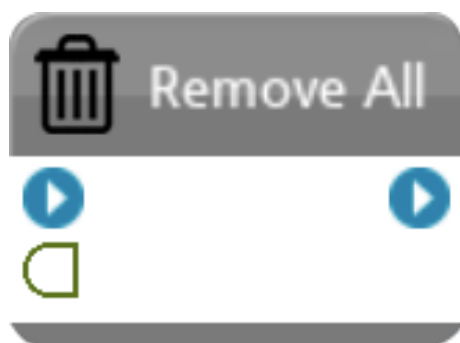
Remove

기능: List에서 해당 인덱스의 값을 제거합니다.

입력: List, Number

출력: 없음

특징: 없음



Remove All

기능: List의 요소를 모두 제거합니다.

입력: List

출력: 없음

특징: 없음

8. Math



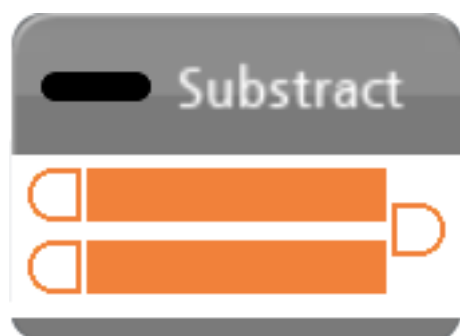
Add

기능: 입력된 두 값을 더합니다.

입력: Number, Number

출력: Number

특징: 없음



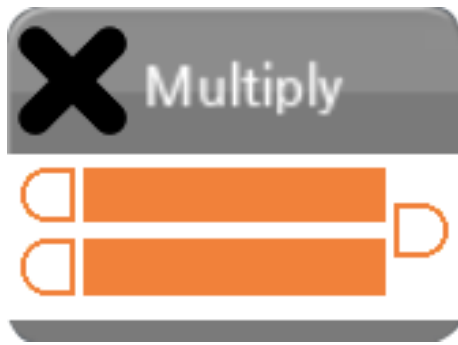
Subtract

기능: 첫번째로 입력된 값에서 두번째로 입력된 값을 뺍니다.

입력: Number, Number

출력: Number

특징: 없음



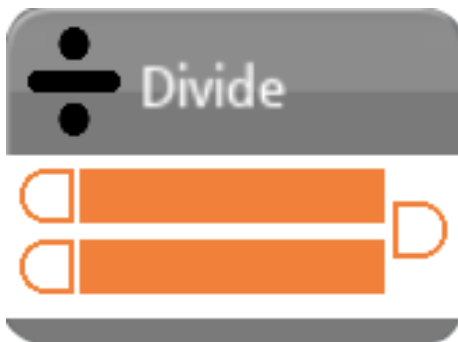
Multiply

기능: 입력된 두 값을 곱합니다.

입력: Number, Number

출력: Number

특징: 없음



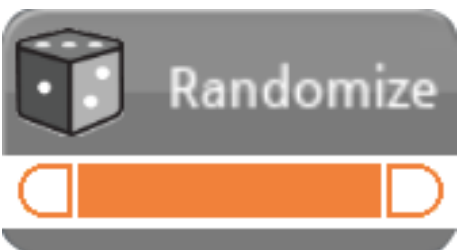
Divide

기능: 첫번째로 입력된 값에서 두번째로 입력된 값을 나눕니다.

입력: Number, Number

출력: Number

특징: 없음



Randomize

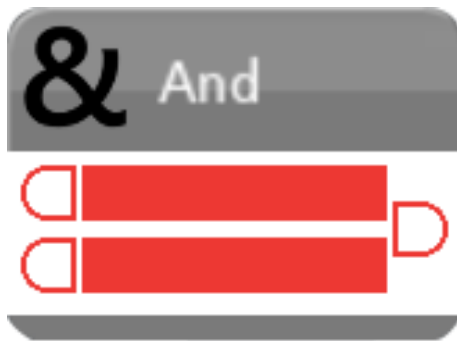
기능: 1부터 입력된 값까지의 범위에서 무작위의 정수 값을 출력합니다.

입력: Number

출력: Number

특징: 없음

9. Logic



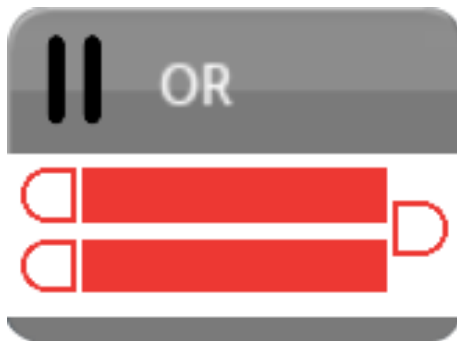
And

기능: 입력된 두 값을 AND 연산(논리곱)하여 출력합니다.

입력: Bool, Bool

출력: Bool

특징: 입력된 두 값이 모두 참일 때만 True(1)를 출력하고, 입력된 두 값 중 1개라도 거짓이면 False(0)를 출력합니다.



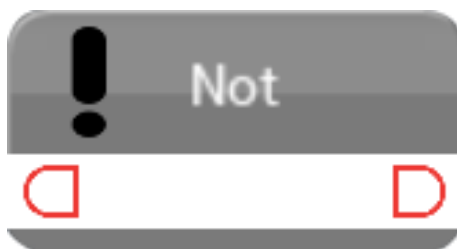
OR

기능: 입력된 두 부울 값을 OR 연산(논리합)하여 출력합니다.

입력: Bool, Bool

출력: Bool

특징: 입력된 두 값 중 1개라도 참이면 True(1)를 출력하고, 그 외의 경우에는 False(0)를 출력합니다.



Not

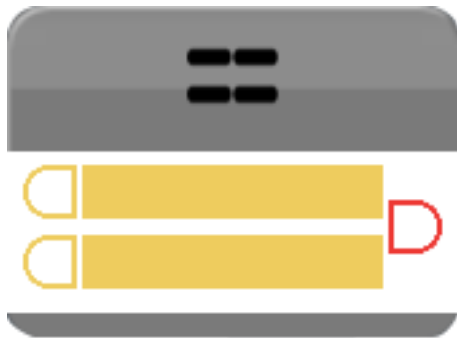
기능: 입력된 부울 값을 반전시켜(논리부정) 출력합니다.

입력: Bool

출력: Bool

특징: True가 입력되면 False를 출력하고, False가 입력되면 True를 출력합니다.

10. Comparatives



Equal

기능: 입력된 두 값이 같은지를 비교하여 참/거짓으로 출력합니다.

입력: String, String

출력: Bool

특징: 같으면 참(True, 1), 다르면 거짓(False, 0)입니다.



Not Equal

기능: 입력된 두 값이 다른 지 비교하여 참/거짓으로 출력합니다.

입력: String, String

출력: Bool

특징: 다르면 참(True, 1), 같으면 거짓(False, 0)입니다.



Less Equal

기능: 입력된 앞의 값이 뒤의 값보다 작거나 같은지 비교하여 참/거짓으로 출력합니다.

입력: Number, Number

출력: Bool

특징: 없음



Greater

기능: 입력된 앞의 값이 뒤의 값보다 큰지 비교하여 참/거짓으로 출력합니다.

입력: Number, Number

출력: Bool

특징: 없음



Greater Equal

기능: 입력된 앞의 값이 뒤의 값보다 크거나 같은지를 비교하여 참/거짓으로 출력합니다.

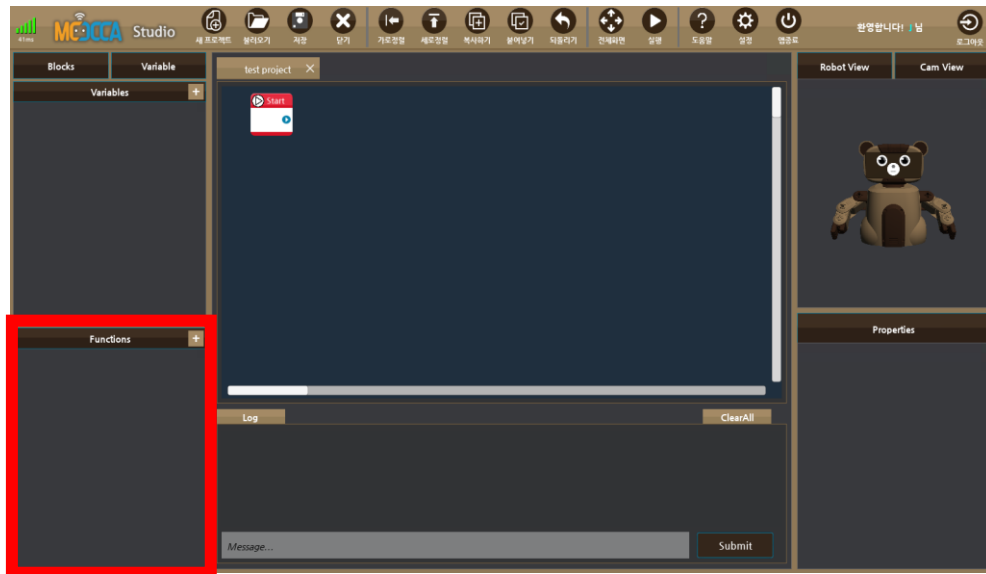
입력: Number, Number

출력: Bool

특징: 없음

8. 함수 만들기

- 함수(Function): 특정 작업을 캡슐화 하는 것으로 재사용되는 로직을 함수로 형성합니다.
- 함수 만드는 과정
 - 1) Blocks & Variable창에서 Variable을 클릭하면 아래의 창이 뜹니다. 이 창에서 좌측 하단에 있는 칸이 함수를 형성하고 관리하는 곳입니다.



- 2) 함수 칸에서 우측 상단에 있는 '+' 버튼을 클릭하면 함수의 속성을 설정하는 창이 뜹니다.

Function [OK] [Cancel]

① Name : AddCalculator

② Input [+] [X]

③ Output [+] [X]

- Name(①) : 함수의 이름을 설정합니다. 이때 주의사항은 한글, 띄어쓰기, 특수문자를 포함시킬 수 없습니다.

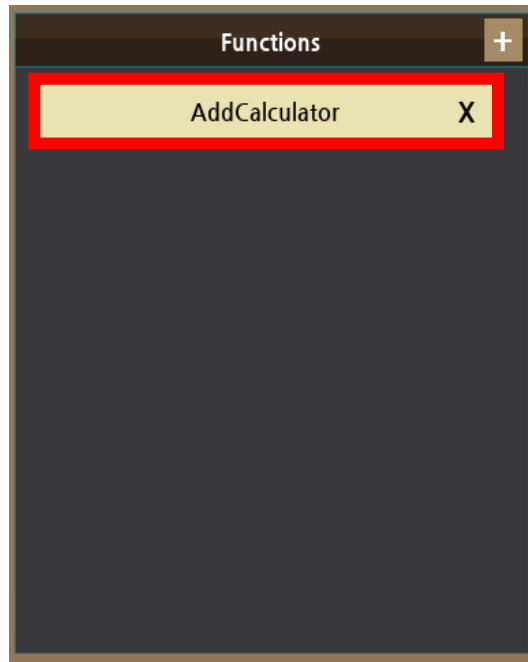
- 3) Input(②)에서 '+' 버튼을 클릭하면 매개변수를 설정할 수 있고, 'X'를 클릭하면 설정한 매개변수들이 모두 삭제됩니다.

- 좌측에 있는 칸을 통해 매개변수의 이름을 설정합니다.
- 드롭다운을 통해 매개변수의 데이터 타입을 설정합니다.
- 우측에 있는 '+' 버튼을 클릭하면 해당 매개변수가 삭제됩니다.

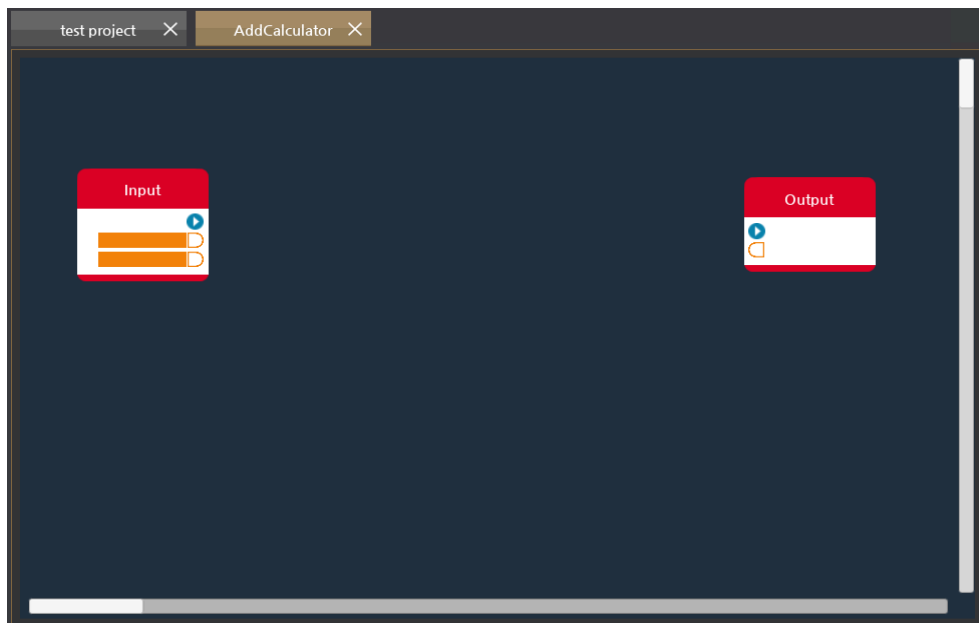
- 4) Output(③)에서 '+' 버튼을 클릭하면 함수의 return 값을 설정할 수 있고, 'X'를 클릭하면 설정한 return 값들이 모두 삭제됩니다.

- 좌측에 있는 칸을 통해 return 값의 이름을 설정합니다.
- 드롭다운을 통해 return 값의 데이터 타입을 설정합니다.
- 우측에 있는 '+' 버튼을 클릭하면 해당 return 값이 삭제됩니다.

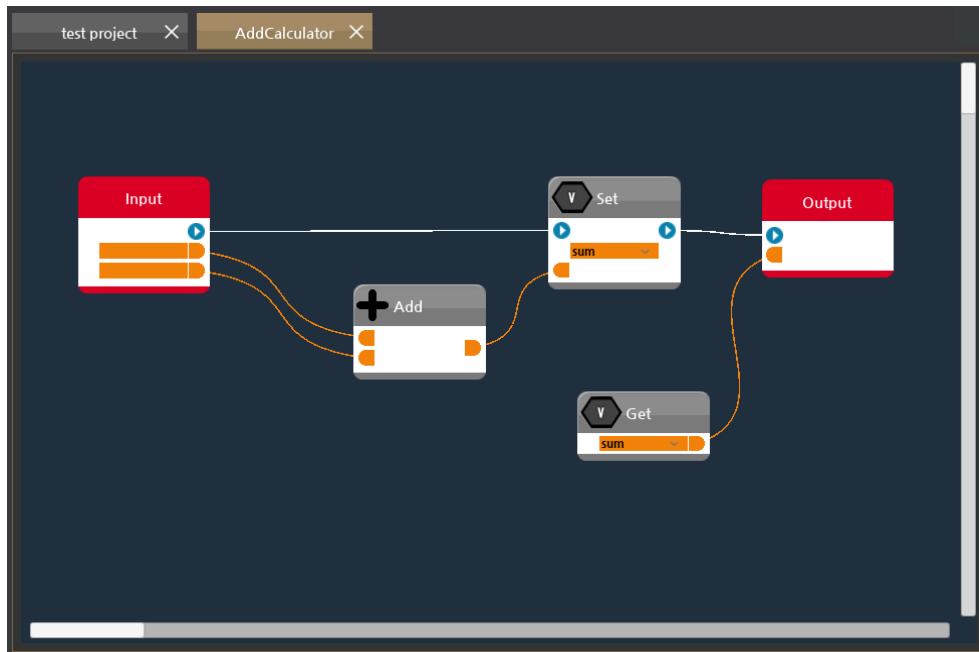
- 5) 함수 구성요소 설정이 모두 끝나면 'OK' 버튼을 클릭합니다. 이를 통해 함수 칸에 사용자가 설정한 함수가 생긴 것을 확인할 수 있습니다. 함수의 구성요소를 다시 수정하고 싶을 경우, 해당 함수 블록을 한번 클릭하면 위 번호에서 본 창이 뜹니다.



- 6) 해당 함수 블록을 더블 클릭함으로써 함수의 로직을 짤 수 있는 탭이 시나리오 창에 띄워집니다. 함수 구성요소 설정 창에서 사용자가 지정한 매개변수와 return 값에 따른 블록이 자동으로 구성된 것을 확인하실 수 있습니다.



- 7) 두 입력된 숫자를 더한 값을 return하는 함수 로직을 구성해보았습니다. 함수 로직을 짜면서 주의해야 할 사항은 flow를 가진 블록을 최소 하나 이상 배치야 합니다. 따라서, 입력으로 들어온 두 개의 number를 Add 블록을 통해 더한 후 따로 생성한 변수에 그 값을 설정하였습니다. 이렇게 설정된 변수의 값을 return해 주었습니다.



- 8) 생성한 함수를 프로젝트 탭에서 실행되도록 구성해보았습니다. 함수 블록은 함수 칸에 생성된 함수 블록을 드래그해서 시나리오 창에 배치하면 됩니다. 사용자로부터 입력 받은 2개의 숫자를 함수에 전달해주고, 함수를 통해 더한 결과를 출력합니다.

