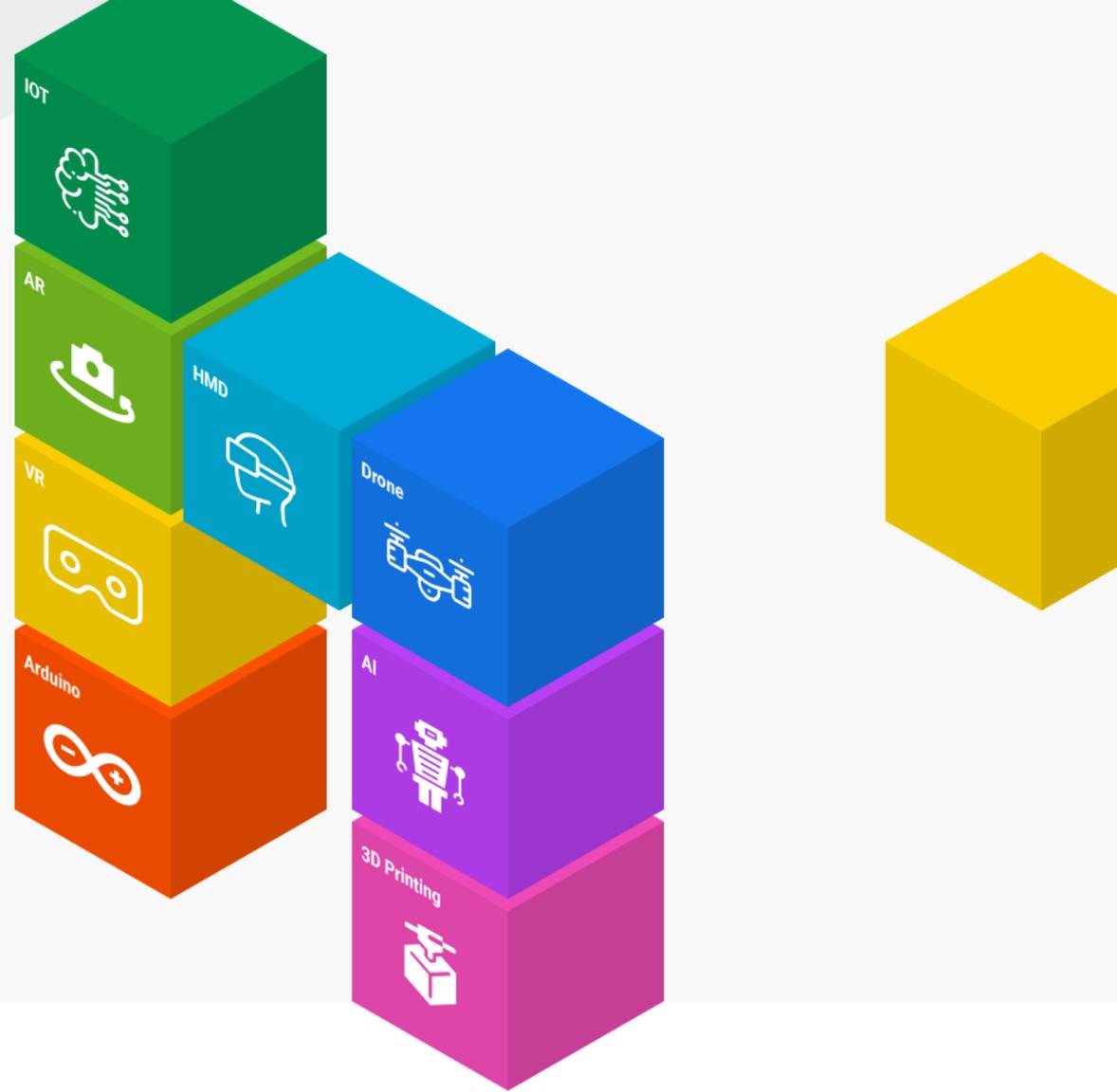


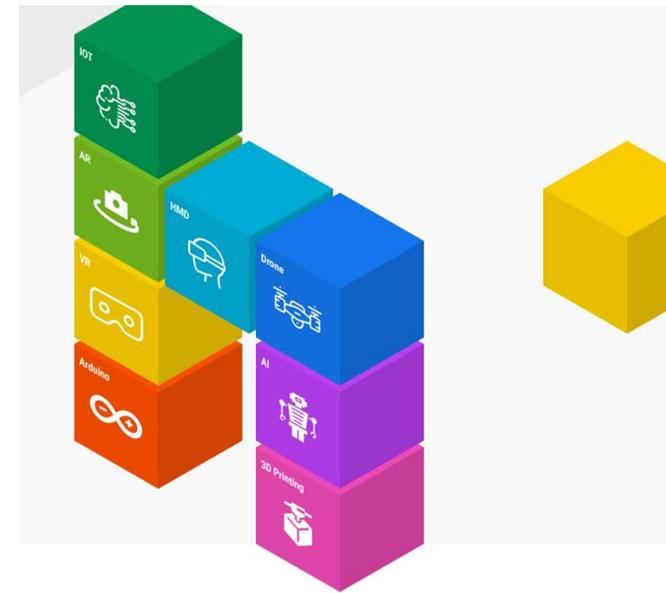
[양자컴퓨터 코딩으로 이해하는 양자역학]  
반복문으로 양자중첩 확률 실험하기



[www.helloapps.co.kr](http://www.helloapps.co.kr)

김 영 준 / 070-4417-1559 / splduino@gmail.com

# 반복문 만들기



# 반복문 만들기



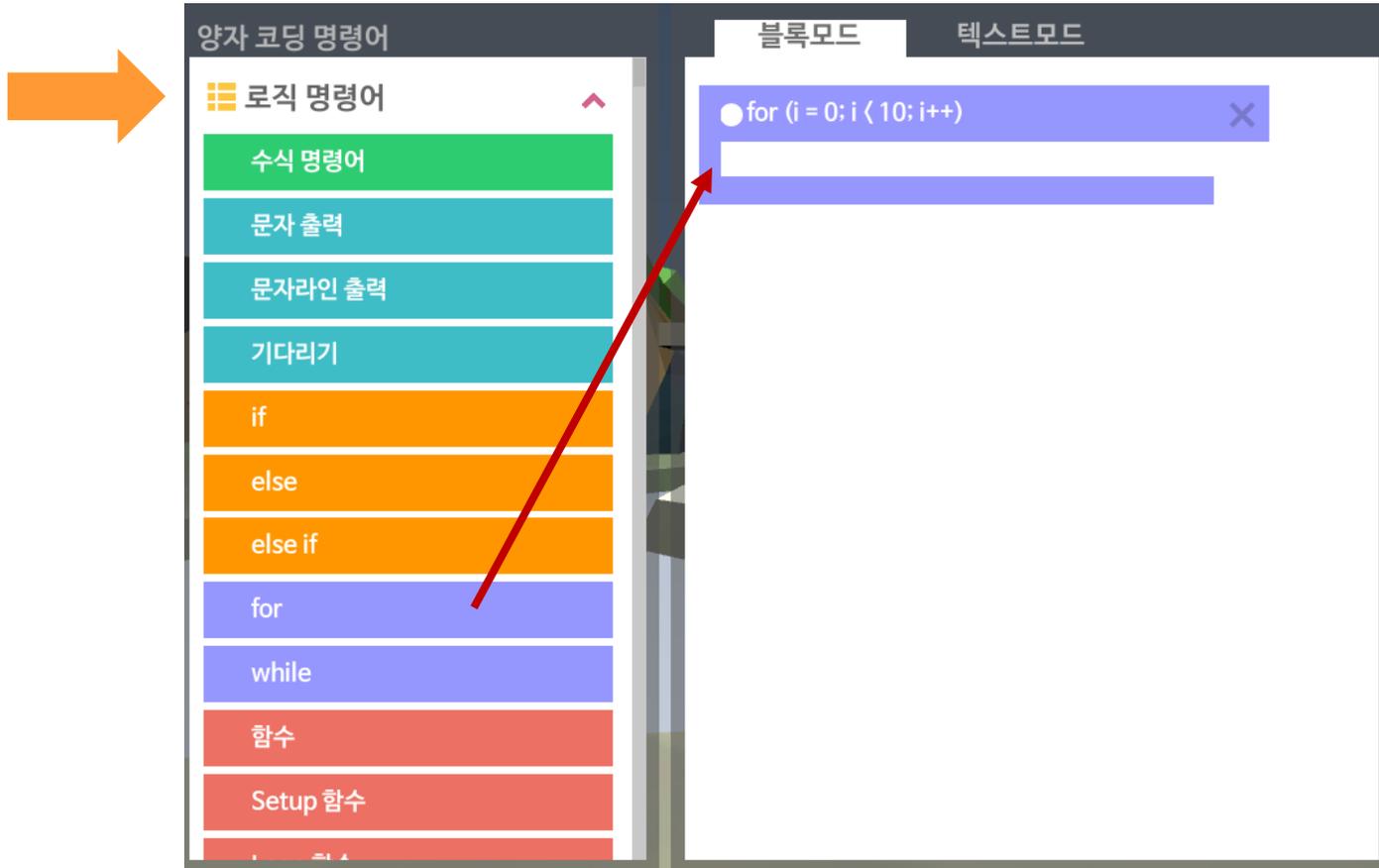
양자 컴퓨팅은 내부적으로 확률 또는 가능성으로 존재하기 때문에 이러한 확률을 확인해 보는 방법은 많은 횟수를 실행시키는 것입니다.

예를 들어 이전 활동에서 작성했던 0 또는 1을 임의로 발생시키는 프로그램은 많은 횟수를 실행시키면 대략 50% 정도의 확률을 보이게 됩니다.

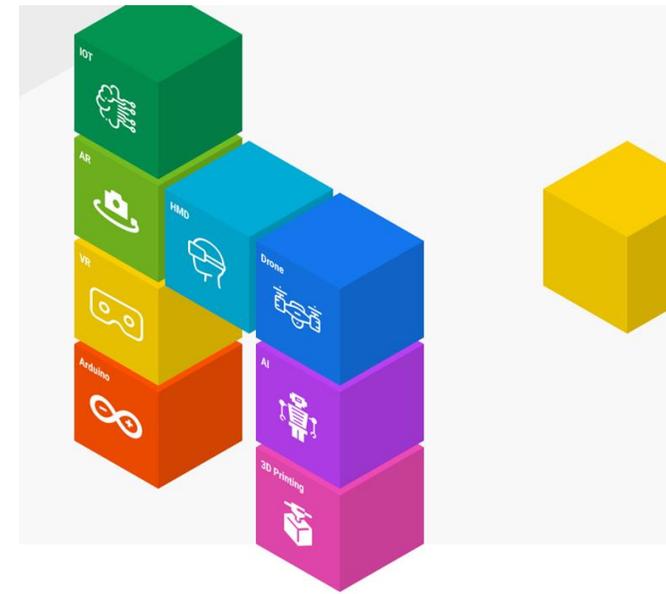
# 반복문 만들기



로직 명령어 그룹에 있는 for 반복문 명령어를 추가합니다.



# 양자 중첩 명령어 추가하기



# 양자 중첩 명령어 추가하기



반복문 안에 큐비트를 생성하는 명령어를 추가합니다.

The screenshot shows a quantum programming interface with two main panels: a left sidebar for commands and a right workspace for code.

- Left Panel (양자 코딩 명령어):** A list of commands including '로직 명령어', '양자 로직 명령어', '양자 회로 입출력', '큐비트 레지스터', '출력 레지스터', '측정하기', '초기화하기', '측정 문자열 읽기', and '단일 큐비트 게이트'. An orange arrow points to the '양자 회로 입출력' category.
- Right Panel (블록모드):** Shows a code block in '블록모드' (Block Mode) containing a loop: `for (i = 0; i < 10; i++)`. Inside the loop, two red blocks have been added: '큐비트 레지스터 q, 1' and '출력 레지스터 c, 1'. Red arrows point from the '큐비트 레지스터' and '출력 레지스터' items in the left panel to these blocks in the workspace.

# 양자 중첩 명령어 추가하기



50% 양자중첩을 만들기 위해 H 게이트 명령어를 추가합니다.

The screenshot shows a quantum programming interface with two main panels. On the left, a sidebar titled '양자 코딩 명령어' (Quantum Coding Commands) lists various gates. An orange arrow points from the 'H 게이트' (H Gate) entry in this list to the right panel. The right panel, titled '블록모드' (Block Mode), shows a circuit diagram. It contains a blue box with the text 'for (i = 0; i < 10; i++)' and a red box with '큐비트 레지스터 q, 1'. Below these are two red boxes for '출력 레지스터 c, 1'. The 'H 게이트 q[i]' is highlighted in orange, and a red arrow points to it from the sidebar. The interface also has a '텍스트모드' (Text Mode) tab.

# 양자 중첩 명령어 추가하기



측정하기 명령어를 추가합니다.

The screenshot shows a quantum programming interface with two main sections: a command list on the left and a code editor on the right.

**양자 코딩 명령어 (Quantum Coding Commands):**

- 로직 명령어 (Logic Commands) - collapsed
- 양자 로직 명령어 (Quantum Logic Commands) - collapsed
- 양자 회로 입출력 (Quantum Circuit I/O) - expanded, highlighted by an orange arrow
- 큐비트 레지스터 (Qubit Register)
- 출력 레지스터 (Output Register)
- 측정하기 (Measurement)
- 초기화하기 (Initialize)
- 측정 문자열 읽기 (Read Measurement String)
- 단일 큐비트 게이트 (Single Qubit Gate) - collapsed
- H 게이트 (H Gate)

**블록모드 (Block Mode):**

- for (i = 0; i < 10; i++)
- 큐비트 레지스터 q, 1
- 출력 레지스터 c, 1
- H 게이트 q[0]
- 측정하기 q, c (highlighted by a red arrow)

# 양자 중첩 명령어 추가하기



측정 문자열 읽기 명령어를 추가합니다.

The screenshot shows a quantum programming interface with two main panels: a command palette on the left and a code editor on the right.

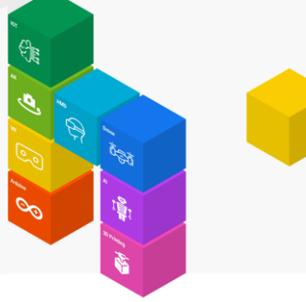
**양자 코딩 명령어 (Quantum Coding Commands):**

- 로직 명령어 (Logic Commands) - collapsed
- 양자 로직 명령어 (Quantum Logic Commands) - collapsed
- 양자 회로 입출력 (Quantum Circuit I/O) - expanded, highlighted by an orange arrow
- 큐비트 레지스터 (Qubit Register)
- 출력 레지스터 (Output Register)
- 측정하기 (Measure)
- 초기화하기 (Initialize)
- 측정 문자열 읽기 (Read Measurement String) - highlighted by a red arrow
- 단일 큐비트 게이트 (Single Qubit Gate) - collapsed
- H 게이트 (H Gate)

**블록모드 (Block Mode):**

- for (i = 0; i < 10; i++)
- 큐비트 레지스터 q, 1
- 출력 레지스터 c, 1
- H 게이트 q[0]
- 측정하기 q, c
- r = 측정 문자열 읽기(c)

# 양자 중첩 명령어 추가하기



로직 명령어에 있는 문자라인 출력 명령어를 추가합니다.

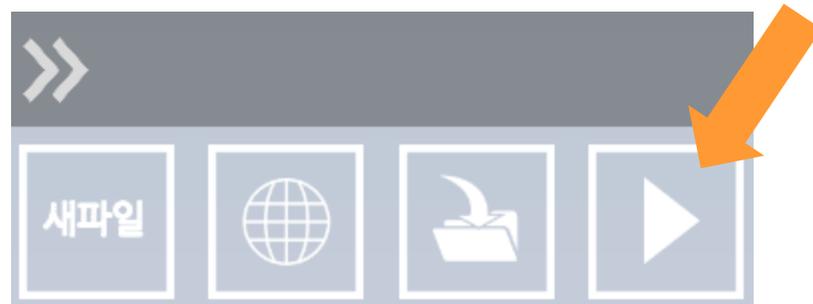
The screenshot shows a quantum programming environment with two main panels. On the left, a sidebar titled '양자 코딩 명령어' (Quantum Coding Commands) lists various command categories. An orange arrow points to the '로직 명령어' (Logic Commands) category, which is expanded to show '수식 명령어' (Mathematical Commands), '문자 출력' (Text Output), '문자라인 출력' (Text Line Output), '기다리기' (Wait), 'if', 'else', 'else if', 'for', and 'while'. A red arrow points from the '문자라인 출력' command in the sidebar to the '문자라인 출력(r)' command in the main editor. The main editor is in '블록모드' (Block Mode) and shows a code block with the following commands, each with a close button (X):

- for (i = 0; i < 10; i++)
- 큐비트 레지스터 q, 1
- 출력 레지스터 c, 1
- H 게이트 q[0]
- 측정하기 q, c
- r = 측정 문자열 읽기(c)
- 문자라인 출력(r)

# 양자 중첩 명령어 추가하기



실행 버튼을 클릭합니다.



# 양자 중첩 명령어 추가하기

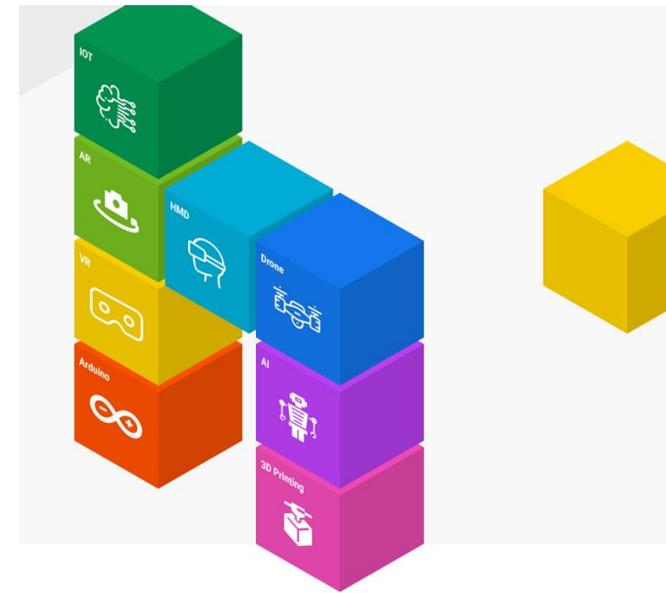


실행시키면 0과 1이 무작위로 나오는 것을 볼 수 있습니다.



0  
0  
1  
0  
0  
1  
0  
0  
0  
1

# 통계 그래프로 보기



# 통계 그래프로 보기



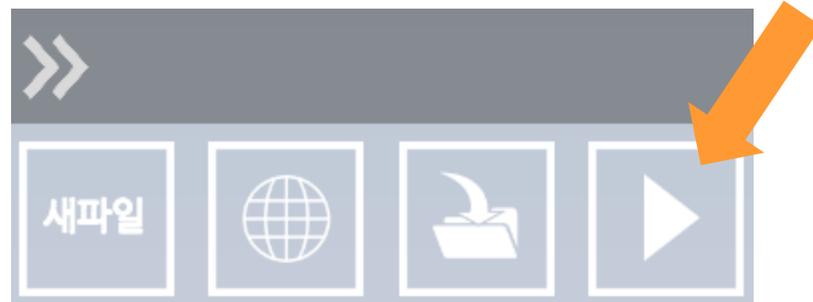
양자 회로 분석 그룹에서 통계 그래프 표시하기 명령어를 추가합니다.

The screenshot shows a software interface for quantum circuit analysis. On the left, a sidebar titled '양자 코딩 명령어' (Quantum Coding Commands) lists various analysis options. An orange arrow points to the '양자 회로 분석' (Quantum Circuit Analysis) group, which is expanded to show several options. A red arrow points from the '통계 그래프 표시하기' (Statistics Graph Display) option in this list to a corresponding command block in the main workspace. The workspace is divided into '블록모드' (Block Mode) and '텍스트모드' (Text Mode). The '블록모드' section contains a sequence of colored blocks representing circuit operations: a blue loop block 'for (i = 0; i < 10; i++)', followed by red blocks for '큐비트 레지스터 q, 1' and '출력 레지스터 c, 1', an orange block for 'H 게이트 q[0]', a purple block for '측정하기 q, c', a blue block for 'r = 측정 문자열 읽기(c)', and a teal block for '문자라인 출력(r)'. The '통계 그래프 표시하기' block is shown in pink and is selected with a radio button. Each block has a close button (X) in its top right corner.

# 통계 그래프로 보기



실행 버튼을 클릭합니다.



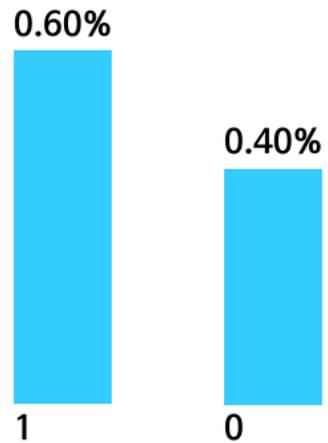
# 통계 그래프로 보기



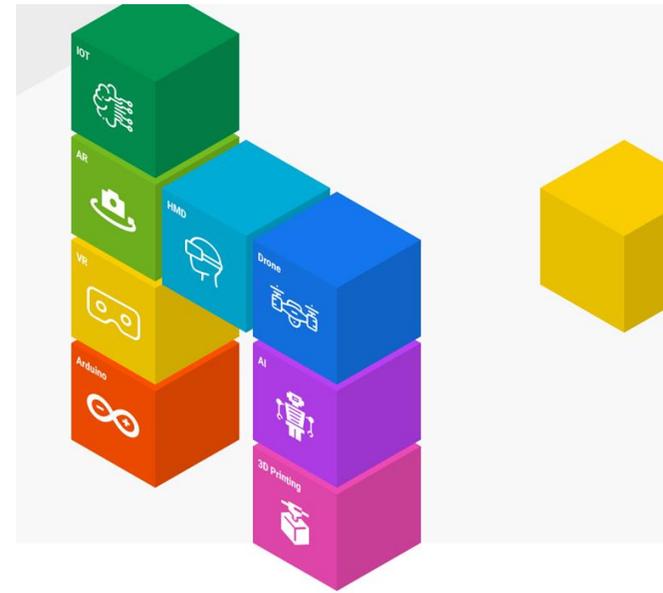
1과 0이 나오는 확률이 각각 60%와 40%로 측정되었습니다.



통계 그래프 표시하기



# 100회로 실험 횟수 늘리기



# 실험 횟수 늘리기



반복문의 반문 횟수를 10에서 100으로 늘립니다.

The screenshot shows a programming environment with two main panes. The left pane is titled '블록모드' (Block Mode) and contains a list of components for a 'for' loop. The right pane is titled '컴포넌트' (Component) and shows the configuration for the loop body.

**블록모드 (Block Mode):**

- for (i = 0; i < 100; i++)
- 큐비트 레지스터 q, 1
- 출력 레지스터 c, 1
- H 게이트 q[0]
- 측정하기 q, c
- r = 측정 문자열 읽기(c)
- 문자라인 출력(r)
- 통계 그래프 표시하기

**컴포넌트 (Component):**

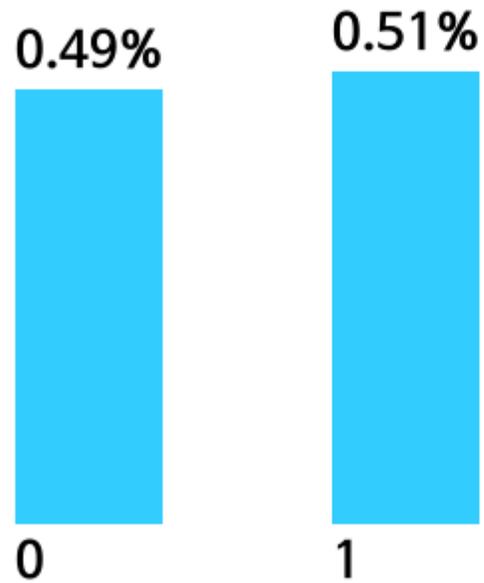
속성 for

- i = 0
- i < 100
- i++

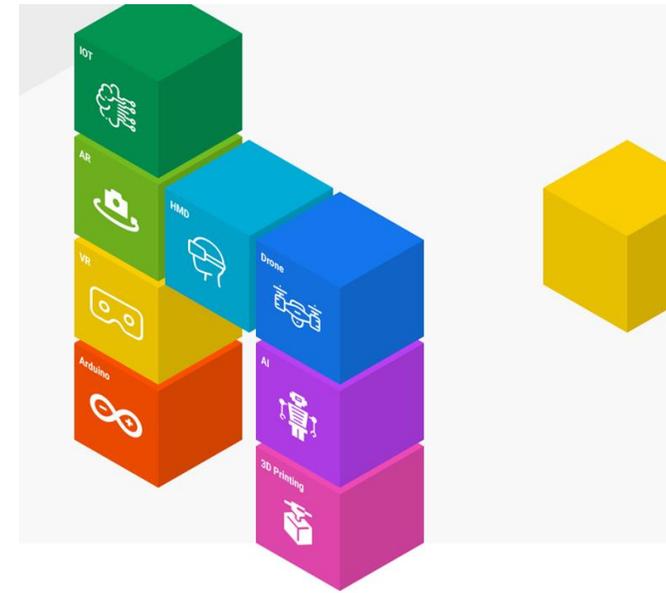
# 실험 횟수 늘리기



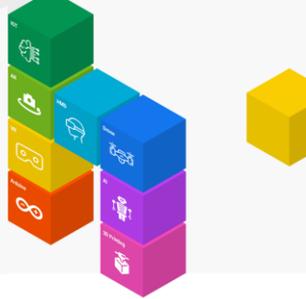
다시 실행해 보면 통계값이 각각 49%와 51%로 나오는 것을 볼 수 있습니다.



# 1,000회로 실험 횟수 늘리기



# 실험 횟수 늘리기



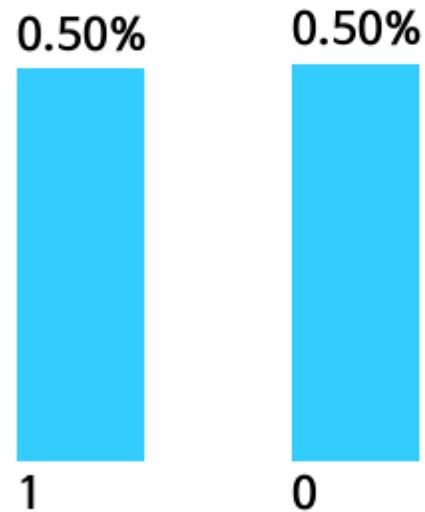
반복문의 반문 횟수를 100에서 1000으로 늘립니다.

The screenshot shows a programming environment with two main panels: '블록모드' (Block Mode) and '컴포넌트' (Component). The '블록모드' panel is active and shows a list of code blocks. The first block is a 'for' loop: `for (i = 0; i < 1000; i++)`. Below it are several other blocks: '큐빗 레지스터 q, 1', '출력 레지스터 c, 1', 'H 게이트 q[0]', '측정하기 q, c', 'r = 측정 문자열 읽기(c)', and '문자라인 출력(r)'. At the bottom of the '블록모드' panel is a pink block: '통계 그래프 표시하기'. The '컴포넌트' panel shows a '속성' (Property) section with a 'for' loop. The loop is visualized with three input fields: 'i = 0', 'i < 1000', and 'i++'. A red arrow points from the 'for' loop in the '블록모드' panel to the 'for' loop in the '컴포넌트' panel.

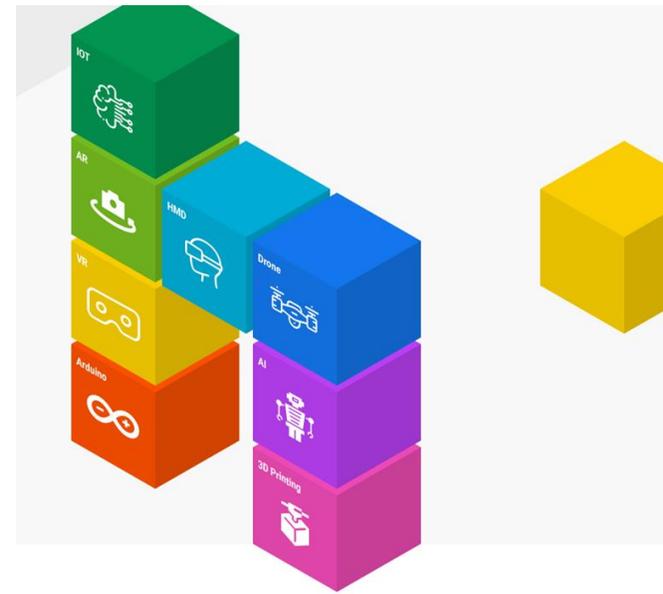
# 실험 횟수 늘리기



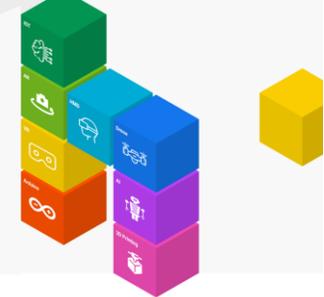
다시 실행해 보면 통계값이 각각 50%와 50%로 나오는 것을 볼 수 있습니다.



# 2개의 큐비트에 대해 발생 확률 측정하기



# 2개의 큐비트에 대해 중첩 만들기



다음과 같이 프로그램을 수정하여 큐비트의 개수를 2개로 늘립니다.

The screenshot shows a quantum programming environment with two main panels: '블록모드' (Block Mode) on the left and '컴포넌트' (Component) on the right.

**블록모드 (Block Mode):**

- for (i = 0; i < 1000; i++)
- 큐비트 레지스터 q, 2
- 출력 레지스터 c, 1
- H 게이트 q[0]
- 측정하기 q, c
- r = 측정 문자열 읽기(c)
- 문자라인 출력(r)
- 통계 그래프 표시하기

**컴포넌트 (Component):**

속성  
큐비트 레지스터

이름	q	X
길이	2	X

A red arrow points from the '● 큐비트 레지스터 q, 2' block in the code mode to the '길이' (Length) field in the component settings, which is set to 2.

# 2개의 큐비트에 대해 중첩 만들기



다음과 같이 프로그램을 수정하여 출력 레지스터의 개수를 2개로 늘립니다.

The screenshot shows a quantum programming interface with two main panels: '블록모드' (Block Mode) and '컴포넌트' (Component).

**블록모드 (Block Mode):**

- for (i = 0; i < 1000; i++)
- 큐비트 레지스터 q, 2
- 출력 레지스터 c, 2
- H 게이트 q[0]
- 측정하기 q, c
- r = 측정 문자열 읽기(c)
- 문자라인 출력(r)
- 통계 그래프 표시하기

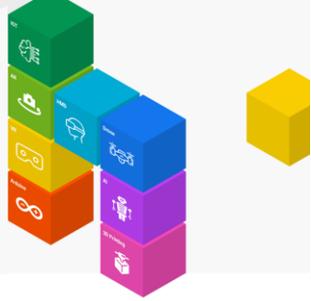
**컴포넌트 (Component):**

속성 출력 레지스터

이름	c
길이	2

A red arrow points from the '출력 레지스터 c, 2' block in the code editor to the '출력 레지스터' component settings, specifically to the '길이' (length) field which is set to 2.

# 2개의 큐비트에 대해 중첩 만들기



전체 큐비트에 대해 H 게이트가 적용되도록 이름을 다음과 같이 수정합니다.

The screenshot shows a quantum circuit editor with two main panels: '블록모드' (Block Mode) on the left and '컴포넌트' (Component) on the right.

**블록모드 (Block Mode):**

- for (i = 0; i < 1000; i++)
- 큐비트 레지스터 q, 2
- 출력 레지스터 c, 2
- H 게이트 q** (highlighted with a red circle and a red arrow pointing to the 3D view)
- 측정하기 q, c
- r = 측정 문자열 읽기(c)
- 문자라인 출력(r)
- 통계 그래프 표시하기

**컴포넌트 (Component):**

- 속성 H 게이트
- 큐비트 q

# 2개의 큐비트에 대해 중첩 만들기



횟수를 10으로 줄여 봅니다.

The screenshot shows a quantum programming interface with two tabs: '블록모드' (Block Mode) and '텍스트모드' (Text Mode). The '블록모드' tab is active, displaying a list of components for a for loop. A red arrow points from the 'for (i = 0; i < 10; i++)' block to the '속성' (Properties) panel on the right.

**블록모드**    텍스트모드

- for (i = 0; i < 10; i++)
- 큐비트 레지스터 q, 2
- 출력 레지스터 c, 2
- H 게이트 q
- 측정하기 q, c
- r = 측정 문자열 읽기(c)
- 문자라인 출력(r)
- 통계 그래프 표시하기

**속성**

for

- i = 0
- i < 10
- i++

# 2개의 큐비트에 대해 중첩 만들기



통계 그래프 표시하기 명령어를 삭제합니다.

● for (i = 0; i < 10; i++)

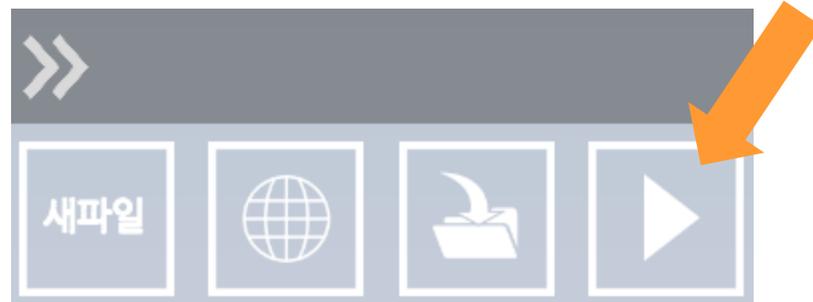
- 큐비트 레지스터 q, 2
- 출력 레지스터 c, 2
- H 게이트 q
- 측정하기 q, c
- r = 측정 문자열 읽기(c)
- 문자라인 출력(r)
- 통계 그래프 표시하기

삭제

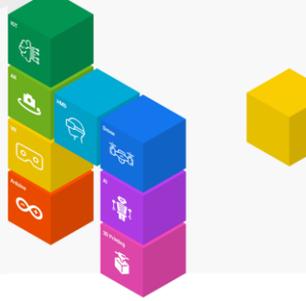
# 2개의 큐비트에 대해 중첩 만들기



실행 버튼을 클릭합니다.



# 2개의 큐비트에 대해 중첩 만들기



2개 큐비트의 값들이 임의로 생성되는 것을 볼 수 있습니다.

```
● for (i = 0; i < 10; i++)  
  큐비트 레지스터 q, 2  
  출력 레지스터 c, 2  
  H 게이트 q  
  측정하기 q, c  
  r = 측정 문자열 읽기(c)  
  문자라인 출력(r)
```



01  
10  
10  
10  
01  
01  
10  
11  
01  
00

# 2개의 큐비트에 대해 중첩 만들기



상태 다이어그램 명령어를 추가합니다.

The screenshot shows a quantum programming interface with two main panels: a menu on the left and a code editor on the right.

**양자 코딩 명령어 (Quantum Coding Commands):**

- 로직 명령어 (Logic Commands) - collapsed
- 양자 로직 명령어 (Quantum Logic Commands) - collapsed
- 양자 회로 입출력 (Quantum Circuit I/O) - collapsed
- 단일 큐비트 게이트 (Single Qubit Gate) - collapsed
- 멀티 큐비트 게이트 (Multi Qubit Gate) - collapsed
- 양자 회로 분석 (Quantum Circuit Analysis) - expanded, highlighted by an orange arrow
- 회로 다이어그램 표시하기 (Show Circuit Diagram)
- 상태 다이어그램 표시하기 (Show State Diagram) - highlighted by a red arrow
- 큐비트 정보 저장하기 (Save Qubit Information)
- 3차원 양자상태 표시하기 (Show 3D Quantum State)

**블록모드 (Block Mode):**

- for (i = 0; i < 10; i++)
- 큐비트 레지스터 q, 2
- 출력 레지스터 c, 2
- H 게이트 q
- 측정하기 q, c
- r = 측정 문자열 읽기(c)
- 문자라인 출력(r)

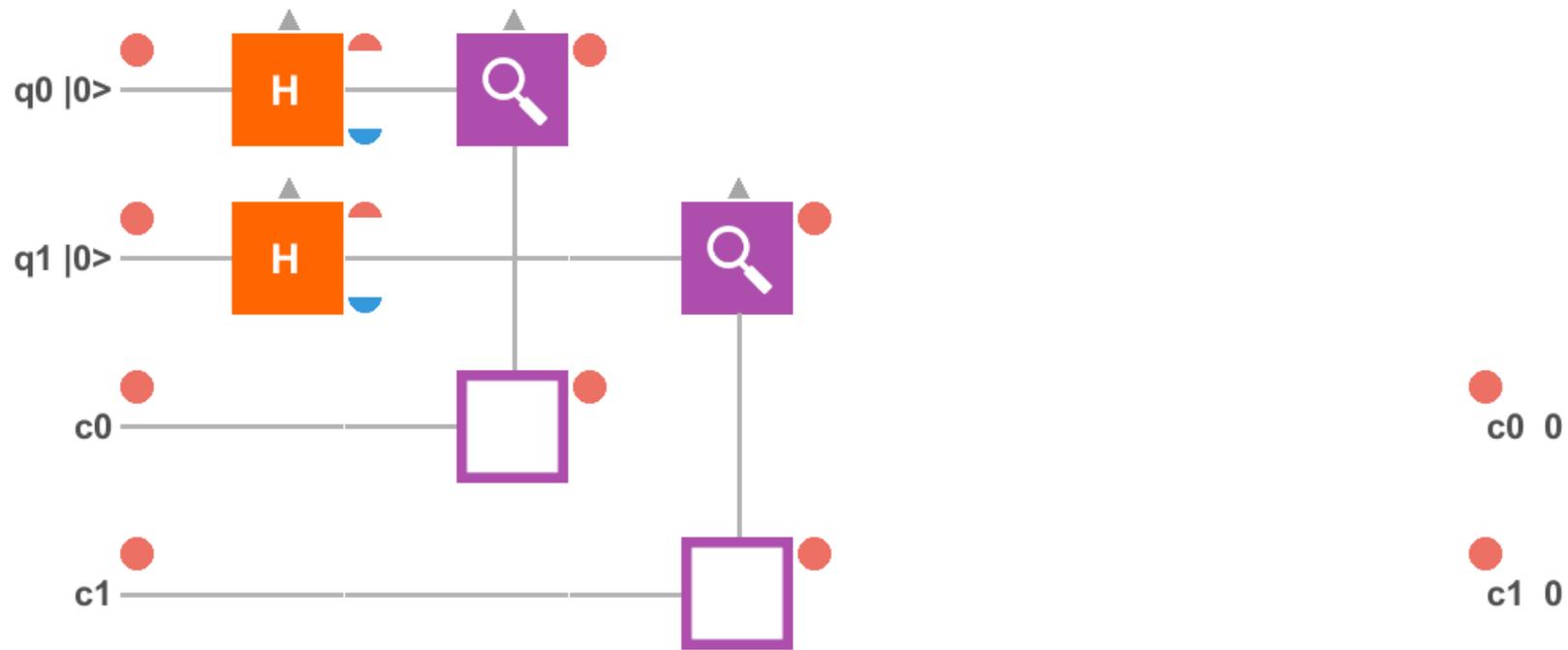
**텍스트모드 (Text Mode):**

- 상태 다이어그램 표시하기 (Selected)

# 2개의 큐비트에 대해 중첩 만들기



다음과 같이 2개의 큐비트에 측정이 연결된 것을 볼 수 있습니다.↳



# 2개의 큐비트에 대해 중첩 만들기



상태 다이어그램 명령어를 제거한 후, 통계 그래프 표시하기 명령어를 추가해 줍니다.

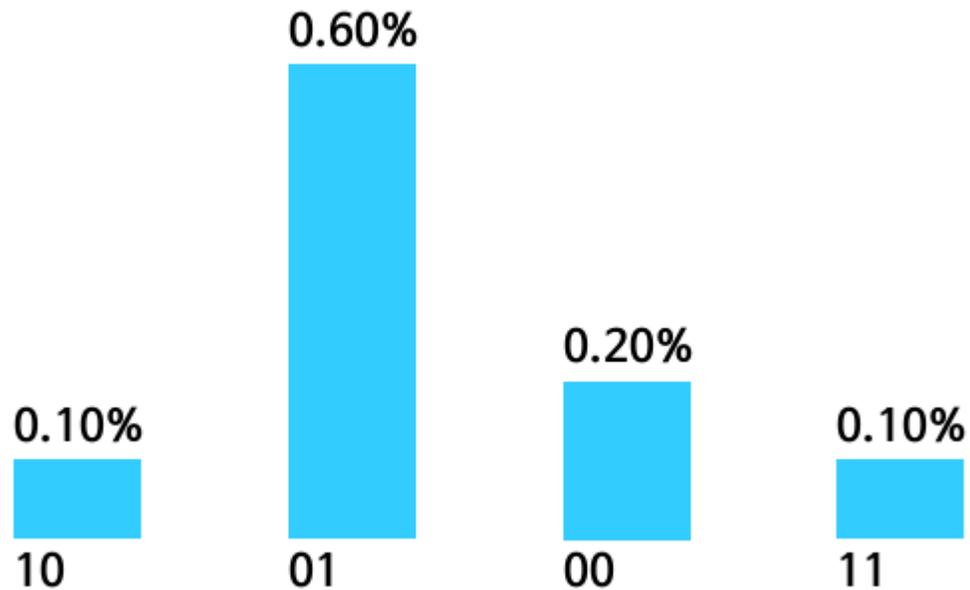
The screenshot shows a quantum programming interface with two main panels: '양자 코딩 명령어' (Quantum Coding Commands) on the left and '블록모드' (Block Mode) on the right. The '양자 코딩 명령어' panel has a dropdown menu open for '양자 회로 분석' (Quantum Circuit Analysis), with an orange arrow pointing to it. The '블록모드' panel shows a list of commands for a quantum circuit, with a red arrow pointing to the '통계 그래프 표시하기' (Show statistics graph) command at the bottom.

양자 코딩 명령어	블록모드	텍스트모드
양자 로직 명령어	for (i = 0; i < 10; i++)	
양자 회로 입출력	큐비트 레지스터 q, 2	
단일 큐비트 게이트	출력 레지스터 c, 2	
멀티 큐비트 게이트	H 게이트 q	
양자 회로 분석	측정하기 q, c	
회로 다이어그램 표시하기	r = 측정 문자열 읽기(c)	
상태 다이어그램 표시하기	문자라인 출력(r)	
큐비트 정보 저장하기		
3차원 양자상태 표시하기		
통계 그래프 표시하기	● 통계 그래프 표시하기	
통계 정보 출력하기		

# 2개의 큐비트에 대해 중첩 만들기



반복 횟수가 10번 밖에 안되기 때문에 다음과 같은 수치가 나왔습니다.



# 2개의 큐비트에 대해 중첩 만들기



반복횟수를 충분히 늘리면 2개의 큐비트에 대해서는 다음과 같이 4가지 상태가 25%에 가까운 확률로 발생하는 것을 볼 수 있습니다.

