

Math 객체란?

Math 객체는 수학적 상수와 함수를 제공하는 내장 객체입니다. 특징은 다음과 같습니다:

- **정적(Static) 객체:** Math는 생성자 함수가 아니므로 `new Math()`와 같이 인스턴스를 생성할 수 없으며, 메서드와 상수를 직접 `Math.methodName()` 또는 `Math.constantName` 형태로 사용합니다.
- **목적:** 복잡한 산술 계산, 무작위 수 생성, 삼각함수, 로그, 지수 계산 등 다양한 수학 연산을 쉽게 구현할 수 있도록 도와줍니다.

Math 상수

Math 객체는 몇 가지 중요한 수학 상수를 제공합니다. 예를 들어:

- `Math.PI` : 원주율 π (약 3.14159)
- `Math.E` : 자연상수 e (약 2.71828)
- `Math.LN10` : 10의 자연로그
- `Math.LN2` : 2의 자연로그
- `Math.LOG2E` : 밑이 2인 로그로 표현한 e
- `Math.LOG10E` : 밑이 10인 로그로 표현한 e
- `Math.SQRT2` : 2의 제곱근
- `Math.SQRT1_2` : $1/2$ 의 제곱근 (즉, $1/\sqrt{2}$)

이러한 상수들은 수학적 계산에서 자주 사용되는 값들을 미리 정의해 놓은 것입니다.

Math 주요 메서드

메서드	설명	구문	예제
Math.abs(x)	주어진 숫자 x 의 절대값을 반환	Math.abs(x)	Math.abs(-5) → 5
Math.acos(x)	x 의 아크코사인(역 코사인, 라디안 단위)을 반환 (x 는 -1 ~ 1)	Math.acos(x)	Math.acos(0.5) → 약 1.0472
Math.asin(x)	x 의 아크사인(역 사인, 라디안 단위)을 반환 (x 는 -1 ~ 1)	Math.asin(x)	Math.asin(0.5) → 약 0.5236
Math.atan(x)	x 의 아크탄젠트(역 탄젠트, 라디안 단위)를 반환	Math.atan(x)	Math.atan(1) → 약 0.7854
Math.atan2(y, x)	좌표 (x, y)에 대한 아크탄젠트를 구하여, 점의 사분면을 고려한 결과를 반환	Math.atan2(y, x)	Math.atan2(1, 1) → 약 0.7854
Math.ceil(x)	x 보다 크거나 같은 최소 정수를 반환 (올림)	Math.ceil(x)	Math.ceil(1.2) → 2
Math.floor(x)	x 보다 작거나 같은 최대 정수를 반환 (내림)	Math.floor(x)	Math.floor(1.8) → 1
Math.round(x)	x 를 반올림한 정수를 반환	Math.round(x)	Math.round(1.5) → 2
Math.cos(x)	x (라디안 단위)의 코사인을 반환	Math.cos(x)	Math.cos(0) → 1
Math.sin(x)	x (라디안 단위)의 사인 값을 반환	Math.sin(x)	Math.sin(Math.PI/2) → 1
Math.tan(x)	x (라디안 단위)의 탄젠트를 반환	Math.tan(x)	Math.tan(0) → 0
Math.exp(x)	자연상수 e 의 x 제곱을 반환	Math.exp(x)	Math.exp(1) → 약 2.7183
Math.log(x)	x 의 자연로그(밑이 e)를 반환	Math.log(x)	Math.log(Math.E) → 1
Math.pow(x, y)	x 의 y 제곱 값을 반환	Math.pow(x, y)	Math.pow(2, 3) → 8
Math.sqrt(x)	x 의 제곱근을 반환	Math.sqrt(x)	Math.sqrt(9) → 3
Math.random()	0 (포함)과 1 (미포함) 사이의 무작위 숫자를 반환	Math.random()	Math.random() → 예: 0.237

Math.sign(x)	x의 부호를 반환: 양수는 1, 음수는 -1, 0은 0, NaN은 NaN	Math.sign(x)	Math.sign(-5) → -1
Math.trunc(x)	x의 소수점 이하 자리를 버리고 정수 부분만 반환	Math.trunc(x)	Math.trunc(4.9) → 4
Math.hypot(...values)	주어진 모든 값들의 제곱의 합의 제곱근을 반환 (유클리드 거리 계산)	Math.hypot(val1, val2, ...)	Math.hypot(3, 4) → 5

2. 연습문제

문제 1. 숫자 맞추기 게임

설명: 컴퓨터가 1 부터 100 사이의 무작위 숫자를 생성하고, 사용자가 입력한 숫자와 비교하는 게임을 구현하세요.

요구사항:

- **무작위 숫자 생성:** `Math.random()`을 활용해서 1~100 의 정수를 생성 (예: `Math.floor(Math.random() * 100) + 1`)
- **입력 및 비교:** 사용자 입력 필드와 버튼을 만들어, 버튼 클릭 시 사용자가 입력한 숫자와 컴퓨터의 숫자를 비교
- **조건문 활용:**
 - 입력한 숫자가 컴퓨터의 숫자보다 크면 "너무 큼니다" 출력
 - 작으면 "너무 작습니다" 출력
 - 같으면 "축하합니다! 맞추셨습니다." 출력
- **DOM 조작:** 페이지 내의 특정 요소(`innerHTML` 이나 `textContent`)를 이용해 결과 메시지를 표시

숫자 맞추기 게임

1부터 100 사이의 숫자를 맞춰보세요.

너무 큼니다.

문제 2. 간단한 사칙연산 퀴즈

설명: 두 개의 무작위 숫자와 무작위로 선택된 사칙연산 기호(+, -, *, /)를 조합하여 산술 문제를 생성하고, 사용자의 답안을 검증하는 퀴즈를 만들어보세요.

요구사항:

- **문제 생성:**
 - 두 개의 무작위 숫자 생성 (예: 1~20 사이)
 - 사칙연산 기호 중 하나를 랜덤으로 선택 (조건문 또는 배열 활용 가능)
- **정답 계산:** 선택된 기호에 따라 정답을 계산 (나눗셈의 경우 소숫점 처리에 주의)
- **사용자 입력:** 입력 필드와 버튼을 만들어 사용자가 답을 제출하면,
- **조건문과 연산자:** 사용자의 답과 계산된 정답을 비교하여 "정답입니다" 혹은 "오답입니다" 출력
- **DOM 조작:** 문제와 결과를 HTML 요소에 동적으로 표시

사칙연산 퀴즈

문제: $5 + 10 = ?$

정답입니다!

문제 3. 무작위 배경색 변경

설명: 버튼을 클릭할 때마다 페이지의 배경색이 무작위로 변경되는 프로그램을 작성하세요. 단, 색상은 미리 정해진 3 가지 색상(예: 빨강, 초록, 파랑) 중 하나여야 합니다.

요구사항:

- **무작위 색상 선택:**
 - `Math.random()`으로 0~2 사이의 정수를 생성
 - 조건문이나 `switch` 문을 사용해 숫자에 따라 "red", "green", "blue" 중 하나를 선택
- **DOM 조작:** `document.body.style.backgroundColor` 를 이용해 페이지의 배경색 변경
- **추가:** 버튼이나 클릭 가능한 요소를 만들어 사용자가 직접 색상을 변경하도록 구현

무작위 배경색 변경

배경색 변경

문제 4. 주사위 굴리기 시뮬레이션

설명: 버튼을 클릭하면 1 부터 6 까지의 숫자가 무작위로 생성되며, 주사위를 굴리는 것처럼 동작하는 프로그램을 작성하세요.

요구사항:

- 무작위 주사위 숫자 생성: `Math.floor(Math.random() * 6) + 1`
- 조건문 활용:
 - 주사위 결과가 6 이면 "최고의 행운!" 메시지 출력
 - 그 외의 경우 "다음 기회에..." 메시지 출력
- DOM 조작: 화면에 주사위의 결과 수정 및 메시지 표시

주사위 굴리기

주사위 굴리기

주사위 결과: 6 - 최고의 행운!

문제 5. 랜덤 인사말 출력

설명: 버튼 클릭 시 "안녕하세요", "반갑습니다", "좋은 하루 되세요" 등의 인사말이 무작위로 한 개 출력되는 프로그램을 만들어보세요.

요구사항:

- **무작위 메시지 선택:**
 - 메시지를 배열에 저장
 - `Math.random()`을 사용해서 배열의 인덱스를 무작위로 선택
- **조건문/Switch 문 활용:** (선택 사항) 숫자에 따라 다른 인사말을 매핑해도 좋습니다.
- **DOM 조작:** 버튼 클릭 시 선택된 인사말을 페이지의 특정 요소에 표시

랜덤 인사말 출력

인사말 출력

반갑습니다

심화 문제: 컴퓨터와의 가위바위보 게임 만들기

문제 설명

HTML, CSS, JavaScript 를 활용하여 사용자가 컴퓨터와 가위바위보 게임을 즐길 수 있는 웹페이지를 만들어보세요. 사용자 인터페이스에 "가위", "바위", "보" 버튼을 배치하고, 사용자가 버튼을 클릭하면 자바스크립트 함수가 실행되어 다음을 처리해야 합니다.

1. 컴퓨터 선택:

- Math.random()을 사용해 1 부터 3 사이의 무작위 정수를 생성하고, 조건문(if/else 또는 switch 문)을 통해 1 은 "가위", 2 는 "바위", 3 은 "보"로 매핑합니다.

2. 게임 결과 결정:

- 사용자의 선택과 컴퓨터의 선택을 비교하여, 승리, 패배, 무승부를 판단합니다.
- 예를 들어, 사용자가 **가위**를 선택했을 때:
 - 컴퓨터가 **보**를 선택하면 사용자 **승리**
 - 컴퓨터가 **바위**를 선택하면 사용자 **패배**
 - 컴퓨터가 **가위**를 선택하면 **비김**

3. 결과 출력:

- DOM 조작을 이용해 컴퓨터가 선택한 값과 게임 결과(승리, 패배, 비김)를 웹 페이지에 출력합니다.

요구사항

• HTML 구성:

- 제목과 간단한 설명
- "가위", "바위", "보" 버튼 (각 버튼 클릭 시 사용자의 선택에 해당하는 값을 함수에 전달)

• JavaScript 구현:

- 컴퓨터의 선택을 랜덤하게 결정 (배열 대신 if/else 로 처리)
- 조건문을 사용하여 사용자의 선택과 컴퓨터의 선택을 비교해 결과를 결정
- DOM 조작(document.getElementById 등)을 통해 결과를 출력

• 주의사항:

- 배열을 사용하지 않고, if 문이나 switch 문만을 활용하여 컴퓨터의 선택을 매핑할 것
- 사용자와 컴퓨터의 선택 및 결과 메시지를 HTML 페이지에 명확하게 표시할 것

가위바위보 게임

아래 버튼을 클릭하여 가위, 바위, 보 중 하나를 선택하세요.

가위	바위	보
----	----	---

컴퓨터의 선택: 보

이겼습니다!