

랜덤 뽑기 프롬프트

구글 사이트와 **html**코드를 활용하여 경품 랜덤 뽑기 웹앱을 만들고 싶어. **html** 코드 하나로만 만들어줘.

첫 화면에서는 상품을 추가, 삭제, 수정할 수 있도록 해줘.

상품을 확정하고 확인을 누르면 카드 형태로 상품들이 나열되고,

선택하면 카드가 뒤 돌며 상품이 공개되는 형태로 제작해줘.

상품 개수도 조절할 수 있도록 하여 확률을 교사가 정할 수 있도록 해줘.

상품을 뽑으면 폭죽이 터지는 등 반응형 효과를 추가하여 학생들의 흥미를 유발해줘.

블루킷 문제 생성 프롬프트

학생들과 이름을 묻고 답하는 영어
표현을 중심으로 퀴즈를 만들고 싶어.
퀴즈는 엑셀 표 형식으로 만들어
활용하고 싶어. 1열에는 문제를
2~5열에는 보기를, 6열에는 20이라는
숫자를, 7열에는 정답을 번호만
적어줘. 10문제 만들어줘.

랜덤발표 프롬프트

구글 사이트와 html코드를 활용하여 경품 랜덤 뽑기 웹앱을 만들고 싶어. 첫 화면에서는 상품을 추가, 삭제, 수정할 수 있도록 해줘. 상품을 확정하고 확인을 누르면 카드 형태로 상품들이 나열되고, 선택하면 카드가 뒤 돌며 상품이 공개되는 형태로 제작해줘. 상품 개수도 조절할 수 있도록 하여 확률을 교사가 정할 수 있도록 해줘. 상품을 뽑으면 폭죽이 터지는 등 반응형 효과를 추가하여 학생들의 흥미를 유발해줘. 예시 상품 중 하나는 선생님의 미소와 함께 웃는 이모티콘을 넣어줘.

성적 조회 웹앱 만들기

구글 시트와 구글 앱스 스크립트를
활용하여 성적을 조회하는 웹앱을
만들고 싶어. 구글 시트 '시트1' 1행은
헤더행이야. 헤더 행은 순서대로
타임스탬프 학년 반 번호 이름
비밀번호 점수로 구성되어 있어. 이름과
비밀번호를 넣었을 때 점수가 보이도록
코드 만들어줘. 코드는 [Code.gs](https://code.gs),
index.html로 만들어줘.

나는 코딩을 전혀 모르는 초보자야.
초보자에게 구현 방법을 친절하게
설명해줘. 추후 수정할 경우 수정한
코드만 안내해주지 말고, 전체 코드를
함께 안내해줘.

타이머 처음 만들기

구글 사이트에 html코드 삽입하여
타이머를 만들고 싶어.

타이머에는 1분 추가, 10초 추가,
초기화 버튼이 있으면 좋겠어.

시작을 누르면 시작 버튼이
일시정지로 바뀌면 좋겠어.

타이머는 기본적으로 0초에
맞춰져 있도록 해줘.

디자인은 심플하고 이쁘게
만들어줘. 타이머가 줄어드는
게이지가 원형으로(시계 방향으로
게이지가 줄어듬) 표시되면
좋겠어.

생활기록부 초안 프롬프트

학생 생활기록부를 작성하는 교사라 생각하고 답변해줘. 학생에 대한 간단한 키워드를 제공하면 내가 서술하는 양식을 참고하여 학생에 대한 서술을 작성해줘.

학생 키워드: 코딩 적극, 다소 산만함, 개선 노력

- 작성 예시

학생 키워드: 바른 태도, 친절, 좋은 교우 관계, 정리 정돈 우수

생활기록부 작성: 기본 생활 태도가 바르게 형성되어 있으며, 주어진 일을 침착하고 끈기 있게 해결하고 꾸준히 노력하는 학습 태도를 가졌음. 말과 행동이 차분하고 성품이 온화하며, 남에게 배려를 잘하고 친절하여 친구들과 사이좋게 지냄. 모둠의 친구들과 협력하여 주어진 과제를 해결하는 태도가 바람직하고 다른 친구들의 의견을 존중하며 잘 들어줌. 매사 성실한 태도로 맡은 일의 끝맺음을 잘하고, 단체 생활의 규칙을 바르게 준수하려는 자세를 가졌으며 질서를 잘 지킴. 교실 정리 정돈에 솔선하고 어려운 상황에 있는 친구를 자발적으로 도울 줄 아는 봉사심이 있음.

타이머 수정하기

아래 타이머 코드에 알람 소리도 추가하고 싶어. 알람 웹링크를 줄테니 타이머가 종료되면 울리도록 해줘.

▶ 알람 소리

"https://folio.itt.link/attachfiles/upload/board/SITE_0000000000000001/323/BBSMSTR_000003234647/FILE_000000005724074_1"

▶ 타이머 코드

```
<!DOCTYPE html>
<html lang="ko">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>하느반 타이머</title>
  <style>
    /* Google Fonts: Noto Sans KR (깔끔하고 현대적인 폰트) */
    @import
url('https://fonts.googleapis.com/css2?family=Noto+Sans+KR:wght@400;500;700&display=s
wap');

    /* 전체 페이지 스타일 */
    body {
      display: flex;
      justify-content: center;
      align-items: center;
      min-height: 100vh;
      margin: 0;
      font-family: 'Noto Sans KR', sans-serif;
      /* 부드러운 하늘색 계열 그라데이션 배경 */
      background: linear-gradient(135deg, #e6e9f0 0%, #eef1f5 100%);
    }

    /* 타이머 컨테이너 */
    .timer-container {
      background-color: white;
      padding: 40px;
      border-radius: 25px; /* 둥근 모서리 */
      box-shadow: 0 15px 35px rgba(0, 80, 150, 0.1); /* 부드러운 그림자 */
      text-align: center;
      width: 320px;
    }

    h1 {
      margin-top: 0;
      margin-bottom: 20px;
      color: #333;
      font-weight: 700;
    }
```

```

/* 원형 게이지 디스플레이 영역 */
.timer-display {
  position: relative;
  width: 220px;
  height: 220px;
  margin: 20px auto;
}

.timer-display svg {
  width: 100%;
  height: 100%;
  transform: rotate(-90deg); /* 12시 방향에서 시작하도록 회전 */
}

.progress-ring__circle {
  /* 게이지 애니메이션: 1초 동안 부드럽게 변화 */
  transition: stroke-dashoffset 1s ease-in-out;
  transform: scaleY(-1); /* 반시계 방향으로 채워지도록 Y축 반전 */
  transform-origin: 50% 50%;
}

/* 시간 텍스트 */
#time-display {
  position: absolute;
  top: 50%;
  left: 50%;
  transform: translate(-50%, -50%);
  color: #2c3e50;
  font-size: 3.5rem;
  font-weight: 500;
  letter-spacing: 2px;
}

/* 버튼 컨트롤 영역 */
.controls {
  display: flex;
  justify-content: center;
  gap: 10px; /* 버튼 사이 간격 */
  margin-bottom: 20px;
}

.controls button {
  background-color: #f0f4f8;
  border: none;
  border-radius: 10px;
  padding: 10px 15px;
  font-family: 'Noto Sans KR', sans-serif;
}

```

```

    font-size: 1rem;
    font-weight: 500;
    color: #4a6a8a;
    cursor: pointer;
    transition: all 0.2s ease;
}

.controls button:hover {
    background-color: #e1e8f0;
    transform: translateY(-2px);
}

/* 시작, 초기화 버튼 영역 */
.main-controls {
    display: flex;
    justify-content: center;
    gap: 10px;
}

.main-controls button {
    border: none;
    border-radius: 10px;
    padding: 12px 25px;
    font-family: 'Noto Sans KR', sans-serif;
    font-size: 1.1rem;
    font-weight: 700;
    cursor: pointer;
    transition: all 0.2s ease;
    flex-grow: 1; /* 버튼이 공간을 균등하게 차지 */
}

/* 시작 버튼 */
#start-btn {
    background-color: #3498db; /* 차분한 파란색 */
    color: white;
}
#start-btn:hover {
    background-color: #2980b9;
}

/* 초기화 버튼 */
#reset-btn {
    background-color: #bdc3c7;
    color: #333;
}
#reset-btn:hover {
    background-color: #a5acb0;
}

```

```

</style>
</head>
<body>

<div class="timer-container">
  <h1>하느반 타이머</h1>

  <div class="timer-display">
    <svg class="progress-ring" width="220" height="220">
      <!-- 배경이 되는 원 (연한 회색) -->
      <circle class="progress-ring__background"
        stroke="#e6e6e6"
        stroke-width="12"
        fill="transparent"
        r="100"
        cx="110"
        cy="110"/>
      <!-- 진행 상태를 나타내는 원 (차분한 파란색) -->
      <circle class="progress-ring__circle"
        stroke="#3498db"
        stroke-linecap="round"
        stroke-width="12"
        fill="transparent"
        r="100"
        cx="110"
        cy="110"/>
    </svg>
    <span id="time-display">00:00</span>
  </div>

  <div class="controls">
    <button id="add-10s">+ 10초</button>
    <button id="add-1m">+ 1분</button>
  </div>

  <div class="main-controls">
    <button id="start-btn">시작</button>
    <button id="reset-btn">초기화</button>
  </div>
</div>

<script>
  // DOM 요소 가져오기
  const timeDisplay = document.getElementById('time-display');
  const progressCircle = document.querySelector('.progress-ring__circle');
  const startBtn = document.getElementById('start-btn');
  const resetBtn = document.getElementById('reset-btn');

```

```

const add10sBtn = document.getElementById('add-10s');
const add1mBtn = document.getElementById('add-1m');

// SVG 원의 반지름과 둘레 계산
const radius = progressCircle.r.baseVal.value;
const circumference = 2 * Math.PI * radius;

// SVG 스타일 초기 설정
progressCircle.style.strokeDasharray = `${circumference} ${circumference}`;
progressCircle.style.strokeDashoffset = circumference;

// 타이머 상태 변수
let totalSeconds = 0;
let initialTotalSeconds = 0;
let timerInterval = null;
let isRunning = false;

// 게이지를 업데이트하는 함수
function setProgress(percent) {
  const offset = circumference - (percent / 100) * circumference;
  progressCircle.style.strokeDashoffset = offset;
}

// 남은 시간을 MM:SS 형식으로 변환하여 표시하는 함수
function updateDisplay() {
  const minutes = Math.floor(totalSeconds / 60);
  const seconds = totalSeconds % 60;
  timeDisplay.textContent = `${String(minutes).padStart(2, '0')}:${String(seconds).padStart(2, '0')}`;
}

// 시간 추가 함수
function addTime(secondsToAdd) {
  if (isRunning) return; // 타이머 작동 중에는 시간 추가 방지
  totalSeconds += secondsToAdd;
  // 최대 시간 제한 (예: 99분 59초)
  if (totalSeconds > 5999) {
    totalSeconds = 5999;
  }
  initialTotalSeconds = totalSeconds;
  updateDisplay();
  // 시간을 설정할 때, 게이지가 반시계 방향으로 채워지는 효과
  setProgress((totalSeconds / initialTotalSeconds) * 100 || 0);
}

// 타이머 시작 함수
function startTimer() {
  if (isRunning || totalSeconds <= 0) return;

```

```
isRunning = true;
initialTotalSeconds = totalSeconds; // 시작 시점의 시간을 총 시간으로 설정
setProgress(100); // 시작과 동시에 게이지를 100%로 채움
```

```
timerInterval = setInterval(() => {
    totalSeconds--;
    updateDisplay();

    // 남은 시간에 따라 게이지 업데이트 (시계 방향으로 후퇴)
    const percent = (totalSeconds / initialTotalSeconds) * 100;
    setProgress(percent);

    if (totalSeconds <= 0) {
        clearInterval(timerInterval);
        isRunning = false;
        // 타이머 종료 알림 (선택)
        // alert("타이머 종료!");
        setTimeout(resetTimer, 500); // 0.5초 후 초기화
    }
}, 1000);
}
```

```
// 타이머 초기화 함수
function resetTimer() {
    clearInterval(timerInterval);
    isRunning = false;
    totalSeconds = 0;
    initialTotalSeconds = 0;
    updateDisplay();
    setProgress(0); // 게이지를 0%로 초기화
}
```

```
// 버튼 클릭 이벤트 리스너 추가
add10sBtn.addEventListener('click', () => addTime(10));
add1mBtn.addEventListener('click', () => addTime(60));
startBtn.addEventListener('click', startTimer);
resetBtn.addEventListener('click', resetTimer);
```

```
// 페이지 로드 시 초기 상태 설정
updateDisplay();
setProgress(0);
</script>
```

```
</body>
</html>
```

랜덤 발표 및 자동 기록 웹앱

구글 시트와 구글 앱스 스크립트를 활용하여 수업 발표자를 랜덤 선정하는 웹앱을 만들고 싶어. 코드는 [Code.gs](#), index.html로 만들어줘.

구글 시트 '시트1' 1~3행은 헤더행이야.

1행은 순서대로 **순, 발표 주제, 일시, 학생 번호**로 구성되어 있어. 순, 발표 주제, 일시 셀은 1행~2행이 병합되어 있어.

2행은 순서대로 순, 발표 주제, 일시, **학생 이름들**로 구성되어 있어.

3행은 발표점수 종합란이야. **1~3열은 발표점수 종합**이라고 입력되어 있고, 이후 열들은 각 학생들의 이름 아래 **발표점수가 자동으로 합산되어 기록**되도록 해줘.

웹앱에서는 발표 점수, 발표 주제를 입력하고 준비를 누르면 각 학생들의 이름이 랜덤 카드 형식으로 섞이도록 해줘. 카드를 선택하면 반응형으로 카드가 뒤집히며 학생을 뽑도록 해줘. 학생의 발표 성공, 실패 여부에 따라 각각 +, - 점수가 시트1에 자동으로 기록되도록 해줘.

API 활용 프롬프팅

구글 시트와 앱스 스크립트, 구글 스튜디오 **AI API**를 활용하여 학생이 감정 일기를 적으면 피드백을 해주는 웹앱을 만들고 싶어. **API** 코드는 다음과 같아. (복사 붙여넣기)

구글 시트는 코드가 작동되면 알아서 양식에 맞춰 작성되도록 해줘.