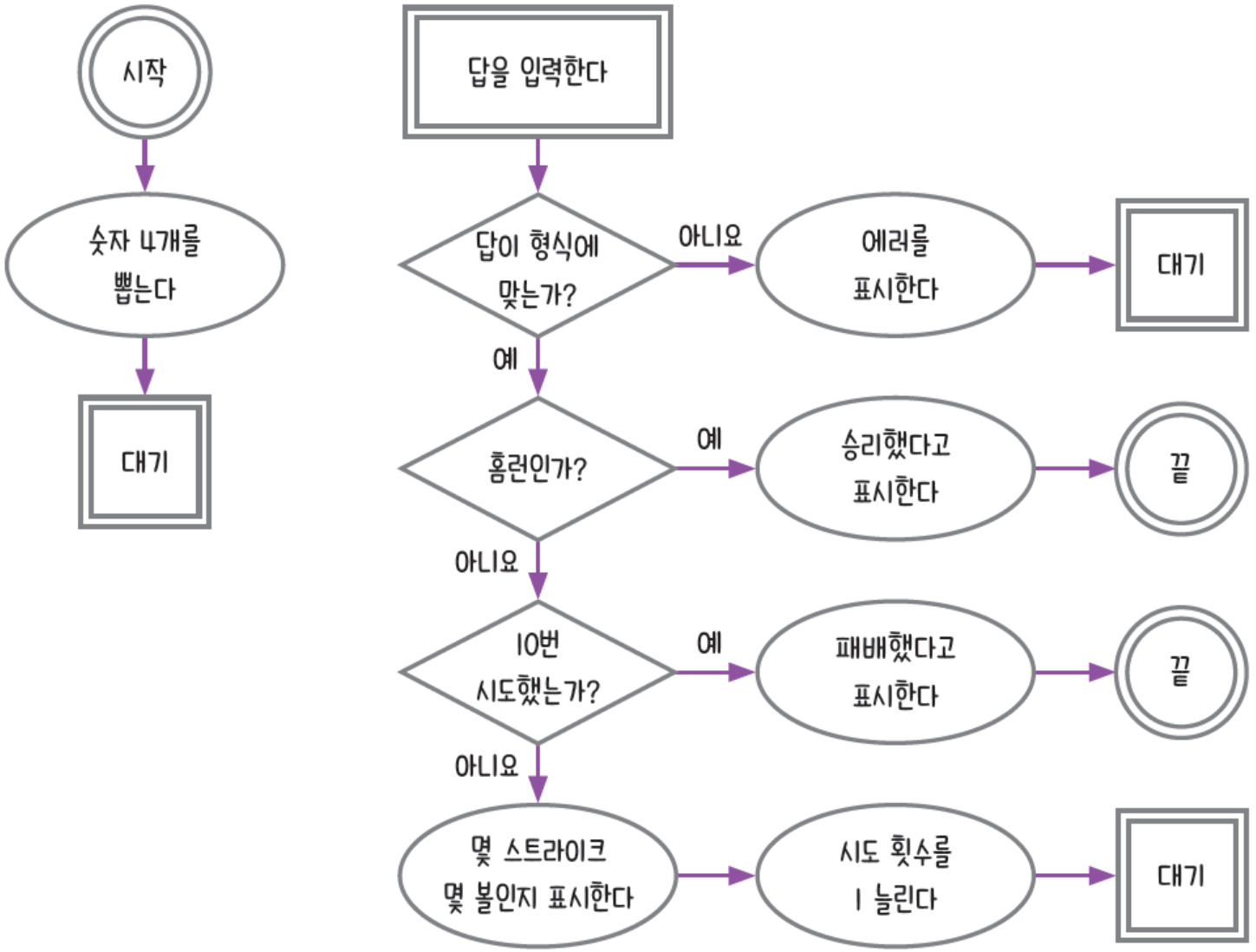
7장 반복문 사용하기: 숫자야구

7.1 이 장에서 만드는 프로그램

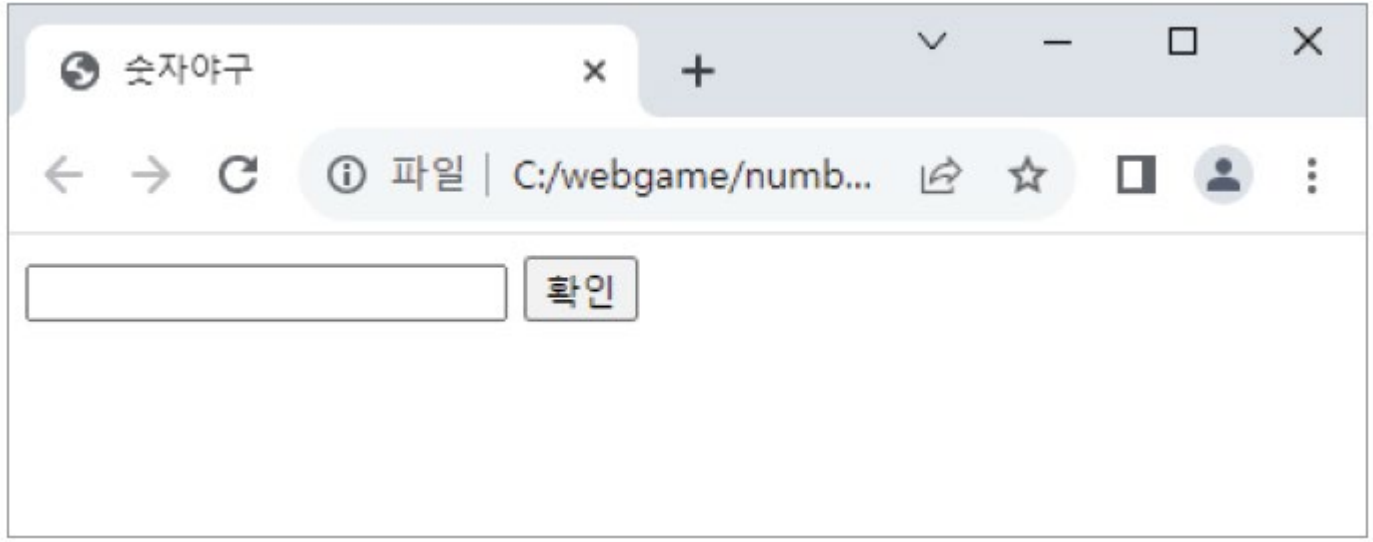
- 숫자야구 순서도



7.2 화면 만들고 숫자 4개 뽑기

7.2.1 화면 만들기

- 숫자 야구 실행 화면




```
<html>
<head>
<meta charset="utf-8">
<title>숫자야구</title>
</head>
<body>
<form id="form">
  <input type="text" id="input">
  <button>확인</button>
</form>
<div id="logs"></div>
<script>
const $input = document.querySelector('#input');
const $form = document.querySelector('#form');
const $logs = document.querySelector('#logs');
</script>
</body>
</html>
```

7.2 화면 만들고 숫자 4개 뽑기

7.2.2 무작위로 숫자 뽑기

- Math 객체 사용
- 무작위 숫자를 중복되지 않게 뽑기
  - 로또의 원리를 이용
  - 1부터 9까지의 숫자를 미리 모아 두고 하나씩 네 번 뽑기
- 반복문에서 조건식에 반복 범위를 작성하는 방법
  - 0부터 9 미만으로 적기
  - 0부터 8 이하로 적기
  - 1부터 10 미만으로 적기
  - 1부터 9 이하로 적기

7.3 입력값 검사하기

- 버튼 태그에 click 이벤트가 아닌 #form 태그에 submit 이벤트 달기
- 버튼을 클릭하지 않고도 [Enter]를 눌러 값을 제출할 수 있음

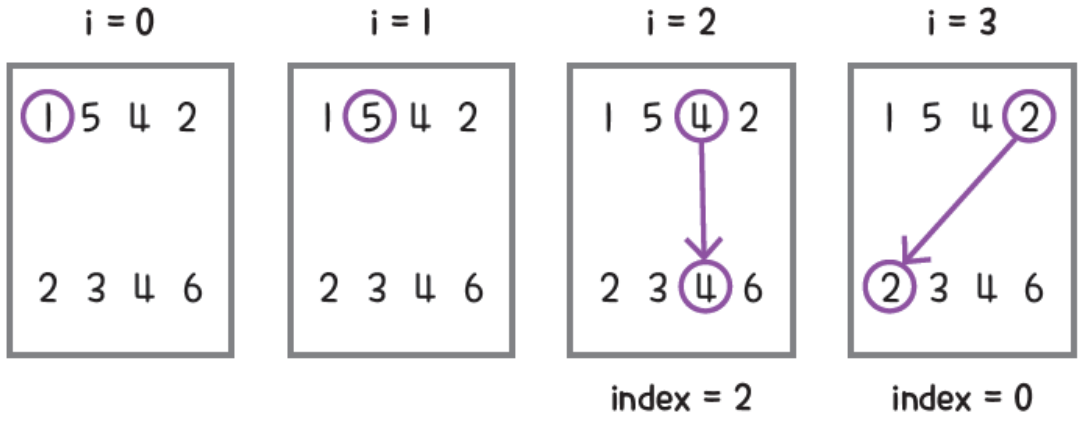
7.4 입력값과 정답 비교하기

7.4.1 홈런 여부와 시도 횟수 검사하기

- 홈런인지 검사
- 시도 횟수가 10번을 넘겼는지 검사

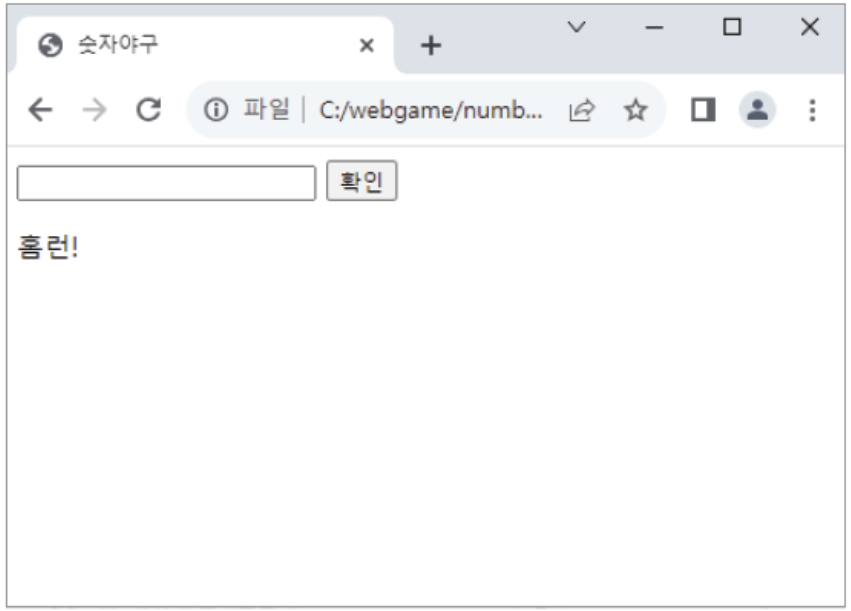
7.4.2 몇 스트라이크 몇 볼인지 표시하기

- 정답 숫자를 하나씩 입력값과 비교해서 같은 숫자가 있는지 찾아보고 자릿수도 일치하는지 확인

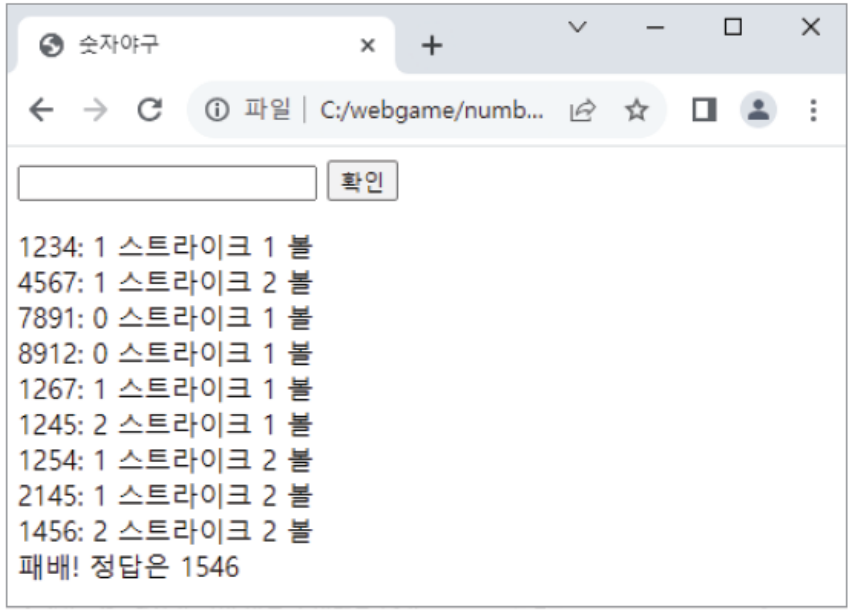


7.4 입력값과 정답 비교하기

- 숫자야구 실행결과



승리

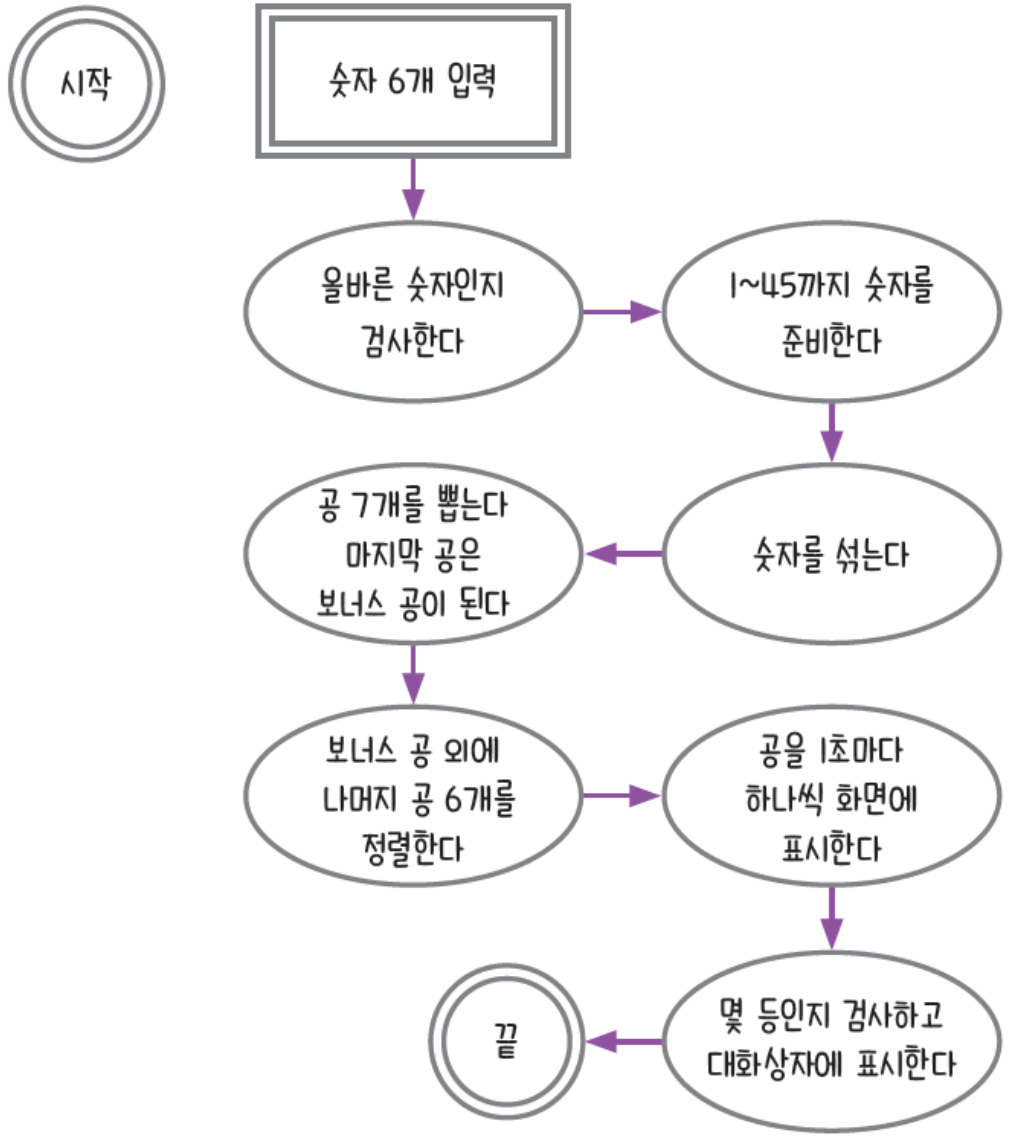


패배

8장 타이머 사용하기: 로또 추첨기

8.1 이 장에서 만드는 프로그램

- 로또 추첨기 순서도



8.2 화면 만들고 숫자 입력받기

- 화면 작성하기
  - 숫자를 입력받을 폼 만들기
  - 폼에 submit 이벤트 리스너 달기

숫자 6개를 쉼표로 구분해 입 | 추첨

당첨 숫자:

보너스 숫자:

```
<!DOCTYPE html>
<head>
<meta charset="utf-8">
<title>로또 추첨기</title>
<style>
.ball {
  display: inline-block;
  border: 1px solid black;
  border-radius: 20px;
  width: 40px;
  height: 40px;
  line-height: 40px;
  font-size: 20px;
  text-align: center;
  margin-right: 20px;
}
</style>
</head>
<body>
<form id="form"> ---------- ①
  <input name="input" placeholder="숫자 6개를 쉼표로 구분해 입력하세요." />
  <button>추첨</button>
</form>
<div id="result">당첨 숫자: </div>
<div id="bonus">보너스 숫자: </div>
<script>
const $form = document.querySelector('#form');
const $result = document.querySelector('#result');
const $bonus = document.querySelector('#bonus');
$form.addEventListener('submit', (event) => { ---------- ②
  event.preventDefault();
});
</script>
</body>
```

8.2 화면 만들고 숫자 입력받기

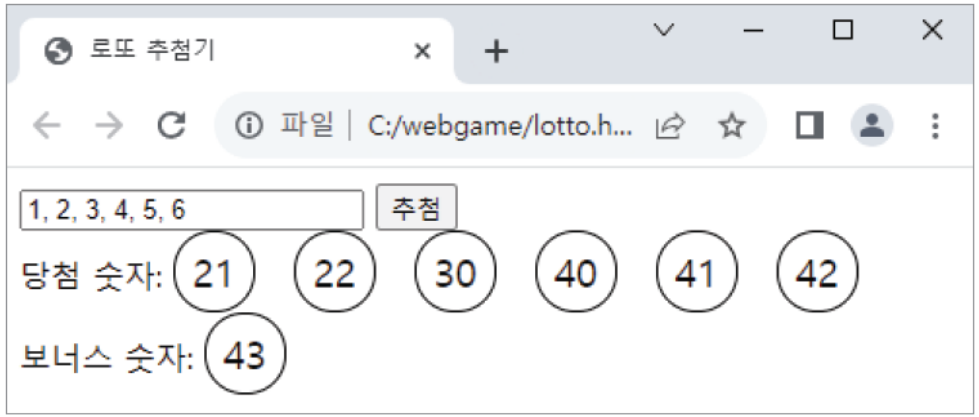
- 사용자로부터 숫자 6개를 입력받기
  - input 태그 하나만 사용
  - input 태그 안에 숫자 6개가 제대로 들어 있는지 확인
  - 숫자를 입력했는지, 숫자가 6개인지, 숫자가 중복되지 않는지, 1부터 45까지 숫자 중에서 입력했는지 검사

8.3 무작위로 공 뽑고 정렬하기

- 무작위로 당첨 숫자 뽑기
  - 7장에서 만든 로직 사용
  - 전체 숫자를 45개로 하고 이 중에서 총 7개를 뽑기
  - 마지막 숫자는 보너스 공
- 숫자 무작위로 섞기
  - 피셔-예이츠 셔플 알고리즘(Fisher-Yates Shuffle algorithm)
  - 무작위 인덱스를 뽑은 후, 해당하는 요소를 새로운 배열로 옮기기
  - 이를 반복하다 보면 새 배열에 무작위로 섞인 숫자들이 들어감
- 숫자 정렬하기
  - 선택 정렬(selection sort)
  - 전체 숫자를 쭉 훑어보면서 작은 순으로 하나씩 가져오며 숫자 정렬

8.4 공 순서대로 표시하기

- 공을 1초에 하나씩 뽑기
  - 타이머 함수(setTimeout()) 사용
- 화면에 공 표시
  - 당첨 공 6개 표시
  - 보너스 공 1개 표시
- 7개의 공을 뽑았을 때



8.4 공 순서대로 표시하기

8.4.1 async/await로 공을 순서대로 표시하기

setTimeout() 대신 프로미스와 async/await 문법 사용

- async/await 문법은 프로미스에만 적용 가능
- setTimeout()을 setTimeoutPromise 고차 함수로 변경합
- 함수 내부에서 await 문법을 사용하기 위해 submit 이벤트 리스너를 async() 함수로 변경

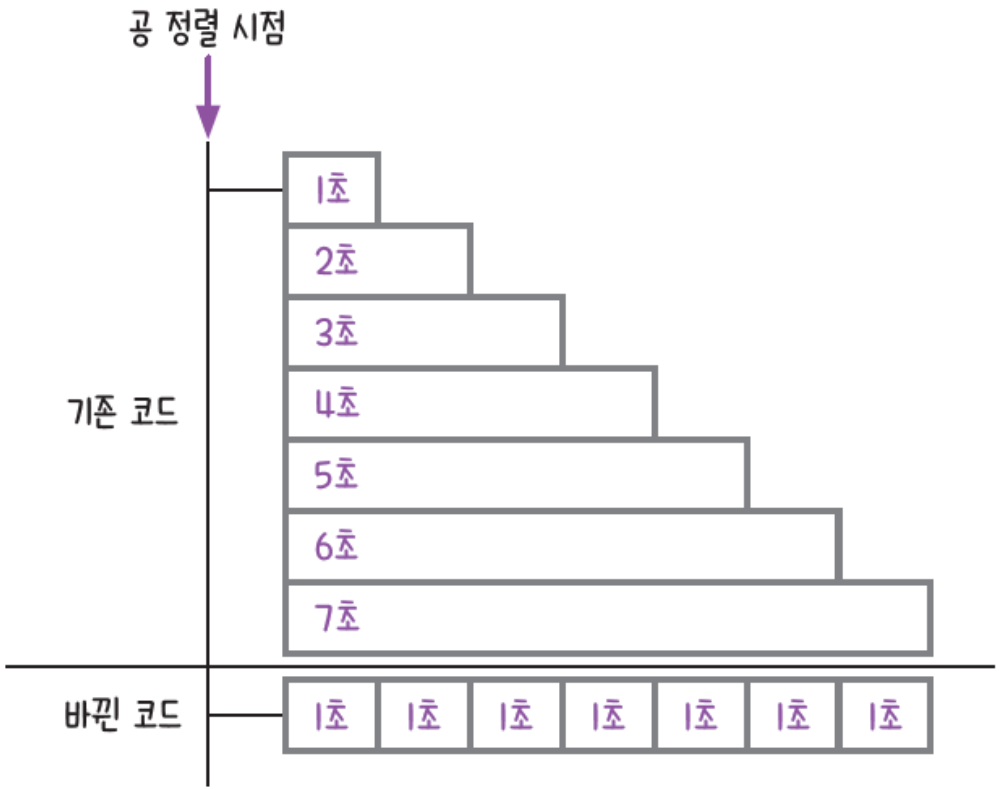
8.4 공 순서대로 표시하기

- 변경된 코드 실행

이전 코드	바뀐 코드
• 정렬 후 1초 뒤에 첫 번째 공 표시	• 정렬 후 1초 뒤에 첫 번째 공 표시
• 정렬 후 2초 뒤에 두 번째 공 표시	• 첫 번째 공 표시 1초 뒤에 두 번째 공 표시
• 정렬 후 3초 뒤에 세 번째 공 표시	• 두 번째 공 표시 1초 뒤에 세 번째 공 표시
• 정렬 후 4초 뒤에 네 번째 공 표시	• 세 번째 공 표시 1초 뒤에 네 번째 공 표시
• 정렬 후 5초 뒤에 다섯 번째 공 표시	• 네 번째 공 표시 1초 뒤에 다섯 번째 공 표시
• 정렬 후 6초 뒤에 여섯 번째 공 표시	• 다섯 번째 공 표시 1초 뒤에 여섯 번째 공 표시
• 정렬 후 7초 뒤에 보너스 공 표시	• 여섯 번째 공 표시 1초 뒤에 보너스 공 표시

8.4 공 순서대로 표시하기

- 타이머의 생성 시점



8.5 몇 등인지 표시하기

- 추첨 결과가 몇 등인지 대화상자로 표시하기
  - 1등과 4~5등은 당첨 숫자를 맞춘 개수에 따라 등수 결정
  - 2등과 3등은 5개의 숫자를 맞췄을 때 보너스 공의 숫자를 맞췄는지에 따라 갈림
  - 보너스 공을 뽑고 나면 등수를 대화상자로 표시

8.5 몇 등인지 표시하기

- alert() 대화상자가 보너스 공보다 더 먼저 보이는 오류
  - alert() 메서드가 화면을 그리는 동작보다 더 먼저 실행되는
  - 화면이 그려지기까지 조금 기다리기(0초만 기다려도 문제 해결)
  - 화면을 그리는 코드(drawBall())와 alert() 사이에 비동기 함수가 있으면 해결됨