

5장 DOM 객체 다루기: 끝말잇기

5.1 이 장에서 만드는 프로그램

5.2 화면 만들고 참가자 수 입력받기

5.3 첫 번째 참가자인지 판단하기

5.4 맞는 단어인지 판단하기

5.5 틀린 단어일 때 표시하기

5.6 순서도 최적화하기

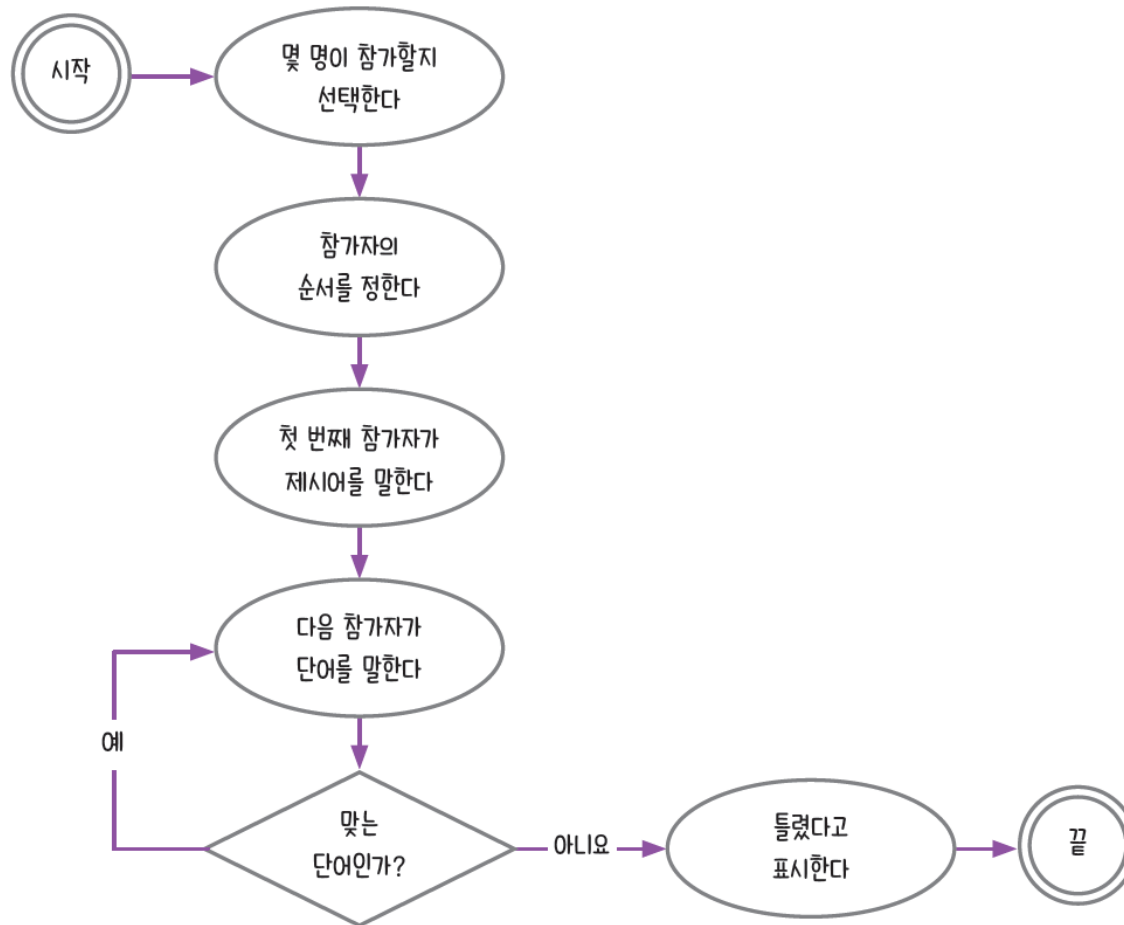
5.1 이장에서 만드는 프로그램

- 절차

1. 게임에 몇 명이 참가할지 선택한다.
2. 참가자의 순서를 정한다(편의상 숫자로 한다).
3. 첫 번째 사람이 어떤 단어를 말한다.
4. 다음 사람이 어떤 단어를 말한다.
5. 절차 4에서 말한 단어가 맞는지 판단한다.
6. 맞다면 다음 사람이 어떤 단어를 말한다.
7. 맞지 않다면 틀렸다고 표시한다.
8. 게임을 계속 진행한다.

5.1 이장에서 만드는 프로그램

- 끝말잇기 순서도



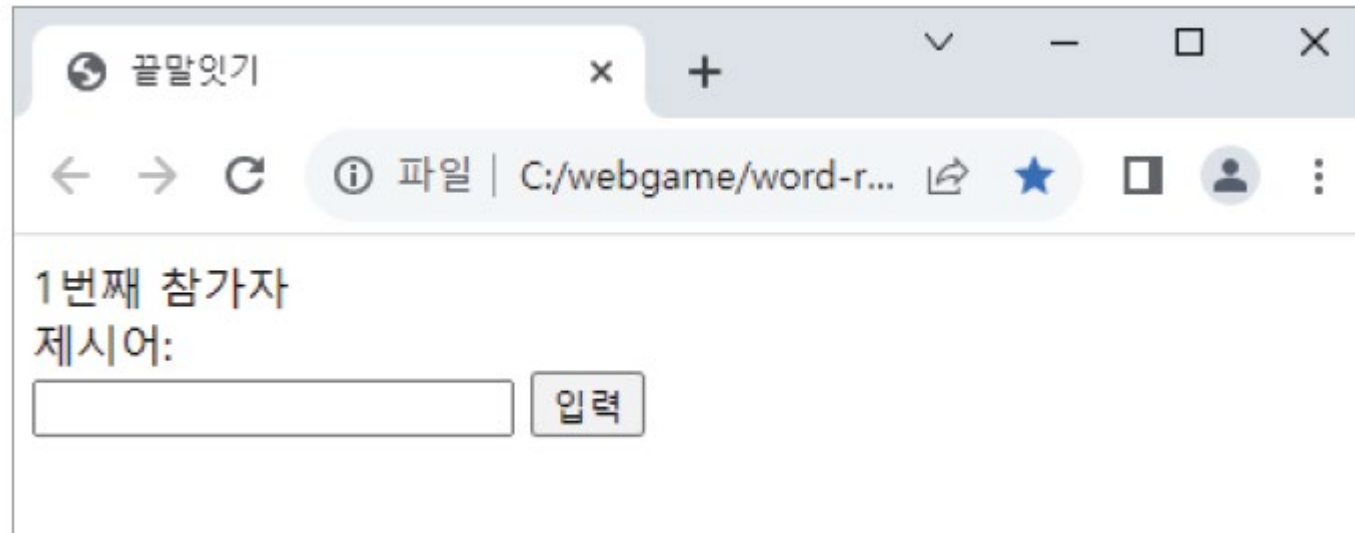
5.2 화면 만들고 참가자 수 입력받기

5.2.1 화면 만들기

- 웹 프로그램: HTML, CSS, 자바스크립트 사용
- HTML은 화면 요소를, CSS는 요소의 디자인을, 자바스크립트는 프로그램의 동작을 담당
- 자바스크립트만으로 화면 요소를 배치하고 디자인할 수 있지만, 자바스크립트로 프로그래밍하는 모든 것을 순서도에 넣어야 함
- 화면 요소는 HTML로 배치하고 디자인은 CSS로 해야 자바스크립트는 부담을 덜고 프로그램의 동작만 담당할 수 있음

5.2 화면 만들고 참가자 수 입력받기

- 화면 만들기 실행결과



A screenshot of a web browser window. The title bar shows a single tab labeled '끝말잇기'. The address bar displays the URL 'C:/webgame/word-r...' with standard navigation icons (back, forward, refresh) and a search icon. The main content area of the browser shows a form titled '1번째 참가자' (1st Participant). Below the title is the label '제시어:' (Prompt:). Underneath the label is a text input field and a button labeled '입력' (Input).

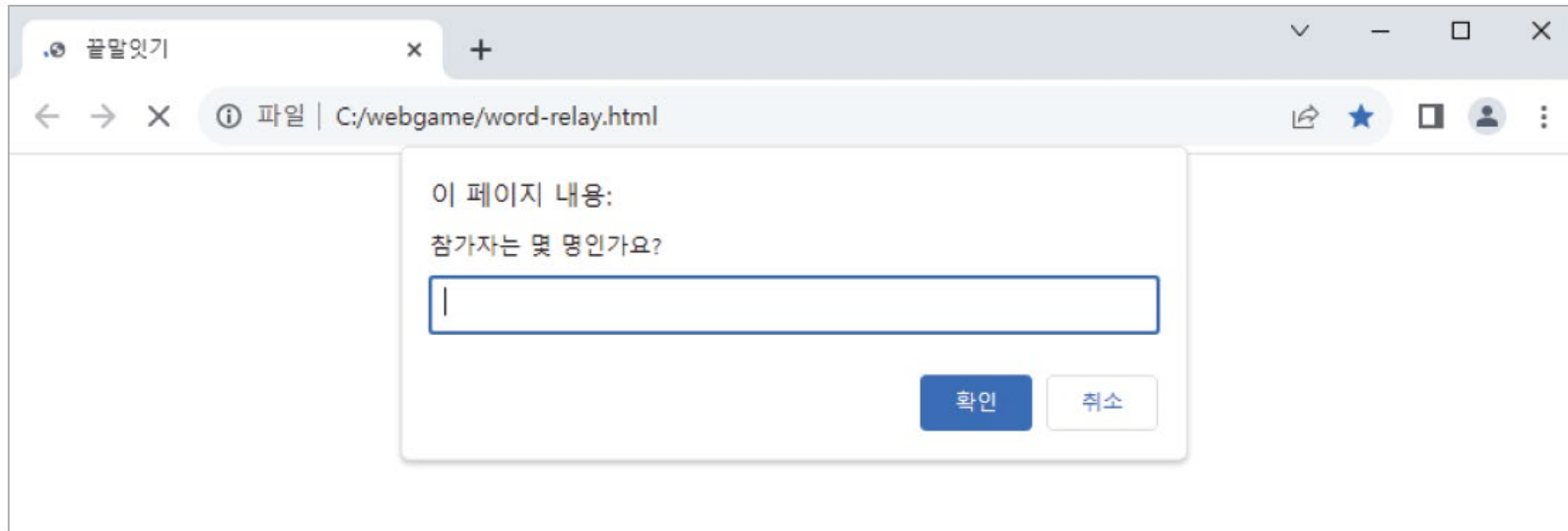
5.2 화면 만들고 참가자 수 입력받기

5.2.2 참가자 수 입력받기

- 절차 1: 몇 명이 참가할지 선택한다
- 몇 명이 참가할지는 입력창으로 직접 프로그램에 알려 줘야 함
- 입력창을 여는 방법
 - input 태그를 사용하는 방법
 - prompt() 메서드를 사용하는 방법
- 자바스크립트도 HTML처럼 위에서 아래로 실행

5.2 화면 만들고 참가자 수 입력받기

- 참가자 수 입력받기 실행결과

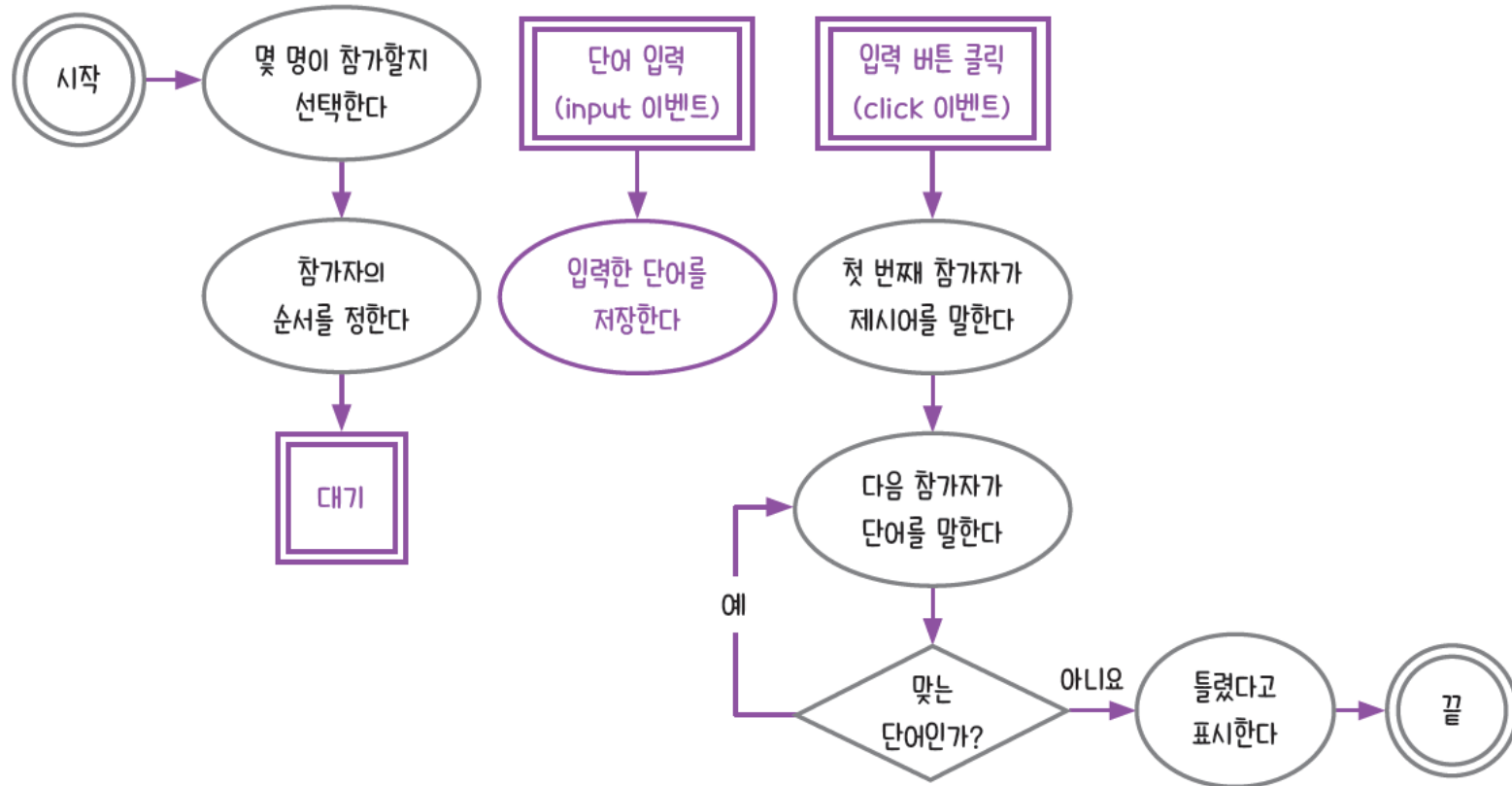


5.3 첫 번째 참가자인지 판단하기

- 절차 3: 첫 번째 참가자가 제시어를 말한다
 - 참가자가 제시어를 입력하는 시간 필요
 - 참가자가 제시어를 입력하는 input 이벤트 발생
- 이벤트가 발생하거나 프로그램이 대기하는 경우에 순서도 끊기
 - 이벤트 발생을 기다리거나 타이머의 콜백 함수가 실행되길 기다리는 상황
 - 이벤트 리스너나 타이머는 대표적인 자바스크립트 비동기 코드
 - 비동기 코드가 실행될 때를 기점으로 순서도를 끊기

5.3 첫 번째 참가자인지 판단하기

- 대기와 이벤트를 반영해 수정한 끝말잇기 순서도

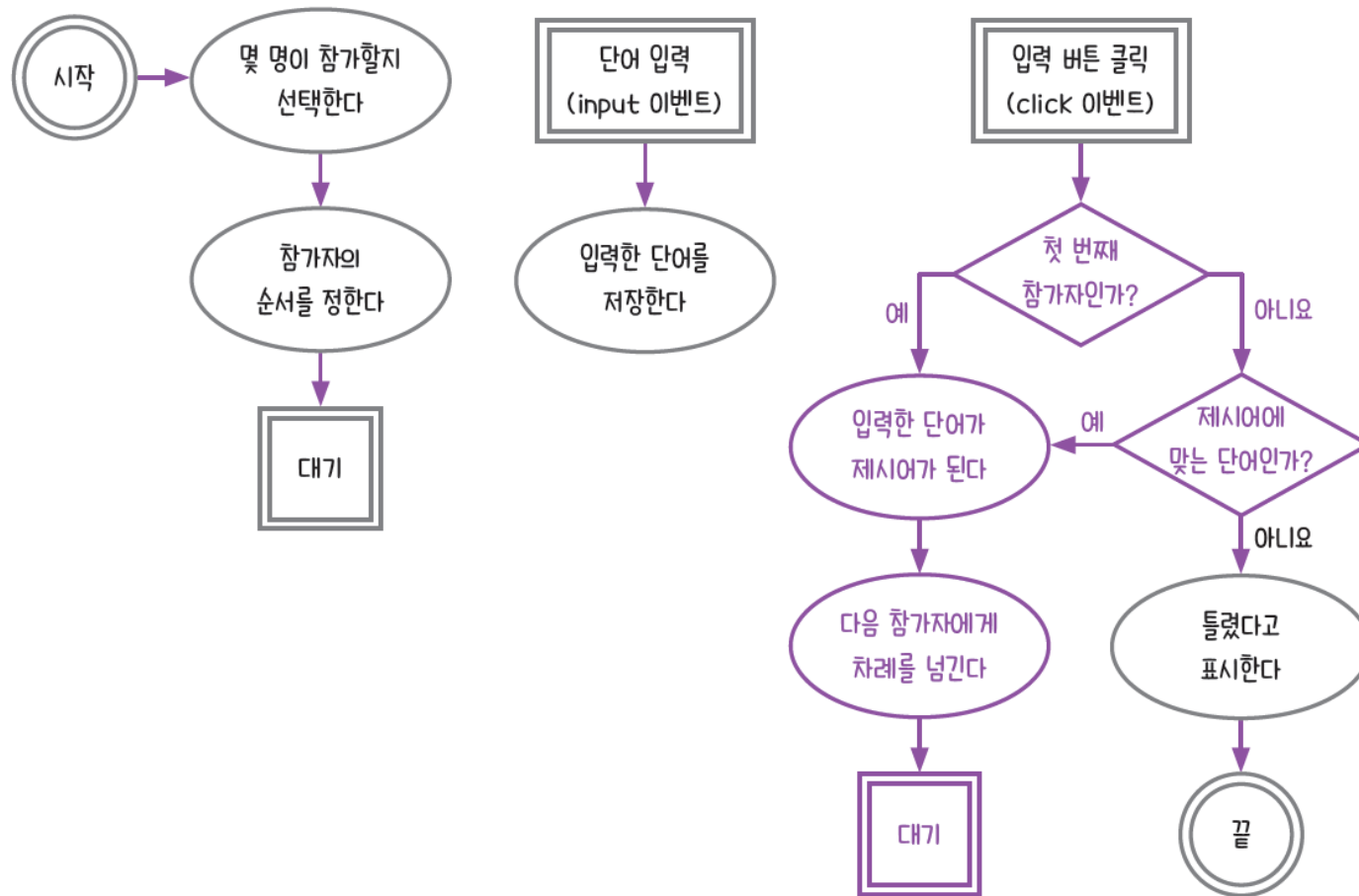


5.3 첫 번째 참가자인지 판단하기

- 단어를 입력한 사람이 첫 번째 참가자인지 판단하는 절차
 - '판단'은 순서도에서 마름모로 표기
 - 판단 결과, 버튼을 클릭한 사람이 첫 번째 참가자라면 입력한 단어는 제시어가 됨
 - 다음 참가자에게 차례를 넘기고 다음 입력이 들어오길 대기
 - 첫 번째 참가자가 아니라면 입력한 단어가 제시어에 맞는 단어인지 판단
 - 맞는 단어라면 다음 참가자에게 차례를 넘긴 후 다시 대기
 - 맞지 않는 단어라면 단어가 틀렸다고 표시하고 프로그램 종료

5.3 첫 번째 참가자인지 판단하기

- 첫 번째 참가자인지 판단하는 절차 반영해 수정한 끝말잇기 순서도

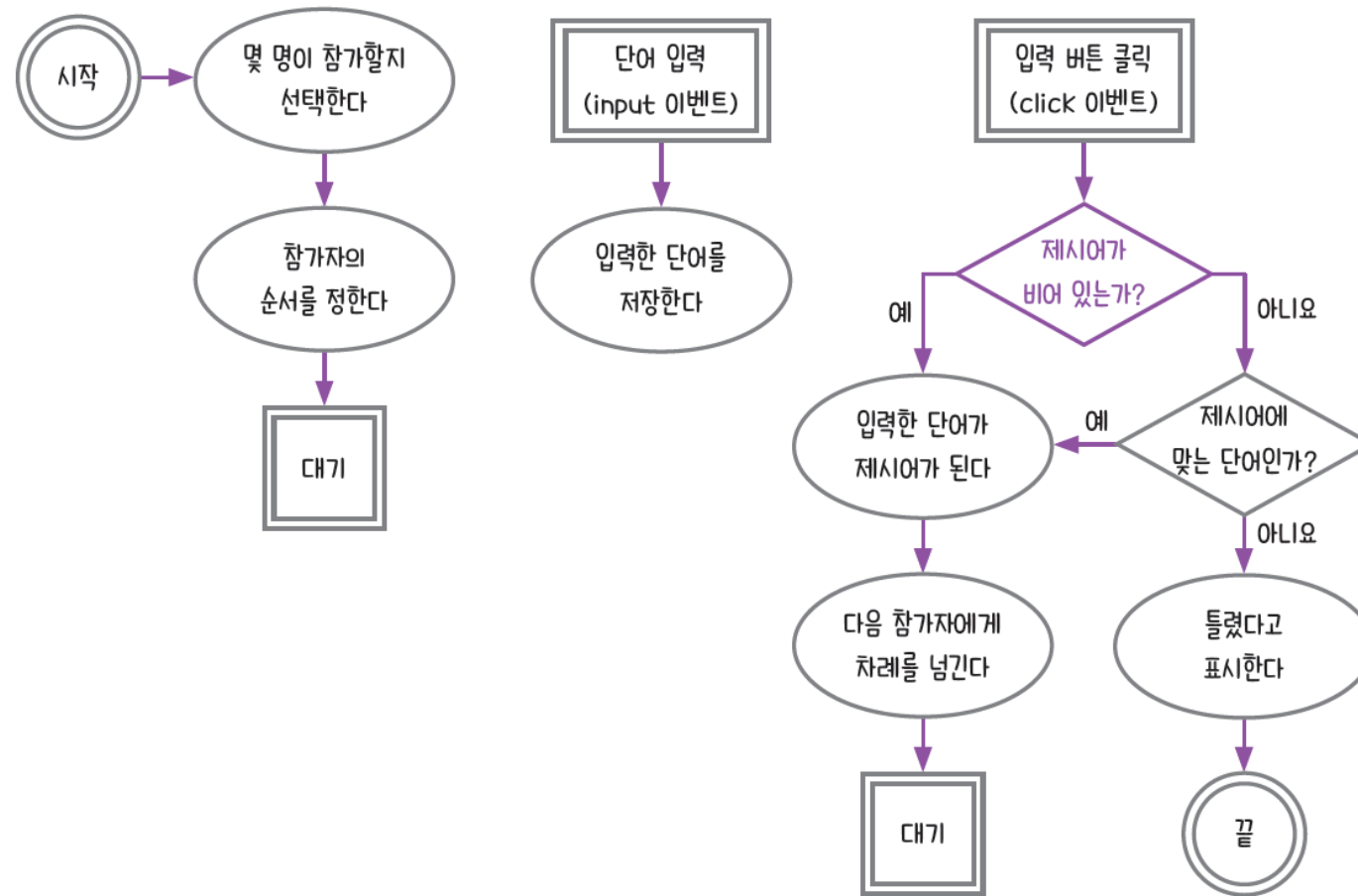


5.3 첫 번째 참가자인지 판단하기

- 입력 버튼을 클릭한 사람이 첫 번째 참가자인지 판단하는 부분
 - 제시어 유무로 판단
 - 제시어를 저장하는 변수 필요, 변수는 빈 값으로 둠
 - 제시어를 저장하고, 다음 사람이 입력한 단어와 비교해서 맞는 단어인지 판단
 - 제시어가 비어 있다면 아직 아무도 단어를 입력하지 않았다는 것이고, 이는 단어를 입력하는 사람이 첫 번째 참가자라는 의미
 - '첫 번째 참가자인가?'라는 절차는 '제시어가 비어 있는가?'로 바꾸기

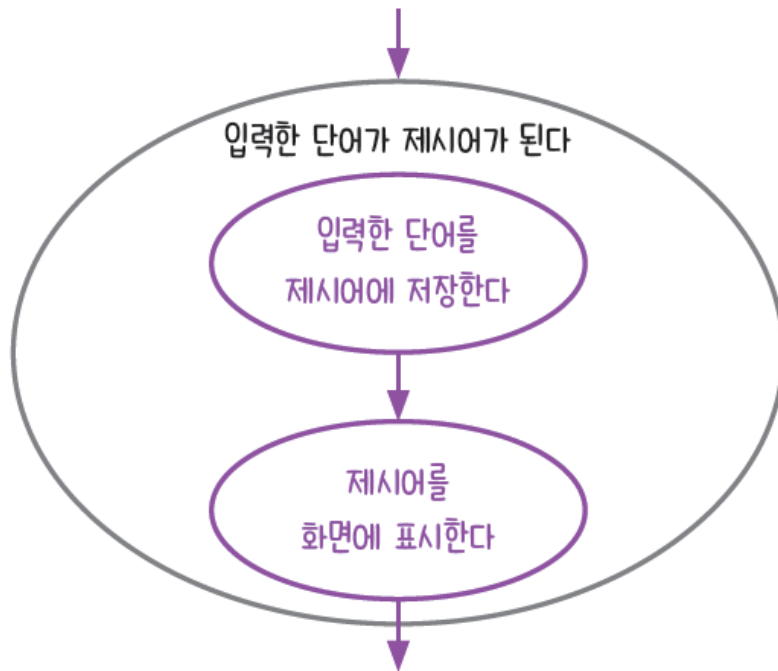
5.3 첫 번째 참가자인지 판단하기

- 제시어가 비어 있는지 판단하는 절차를 반영해 수정한 끝말잇기 순서도



5.3 첫 번째 참가자인지 판단하기

- '입력한 단어가 제시어가 된다'는 절차
 - 다음 차례가 끝말잇기를 하려면 앞에서 입력한 제시어를 알아야 함
 - 제시어를 화면에 표시
- '입력한 단어가 제시어가 된다'는 절차를 반영해 수정한 끝말잇기 순서도



5.4 맞는 단어인지 판단하기

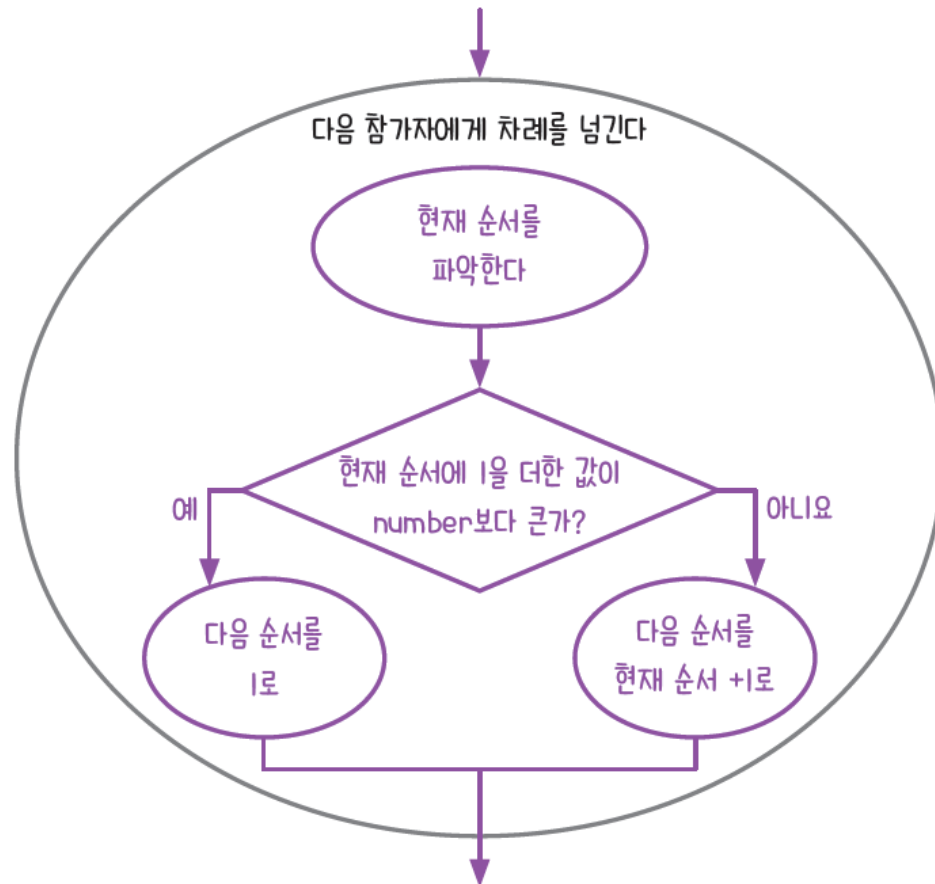
- 맞는 단어인지 판단하는 절차
 - 순서도에서 제시어가 비어 있지 않은 경우
 - 두 번째 사람이 단어를 입력할 때
 - 맞는 단어인지 판단하려면 제시어의 끝 글자와 현재 단어의 첫 글자 비교
 - 맞는 단어라면 다음 차례를 위해 현재 단어를 제시어에 저장, 화면 표시

5.4 맞는 단어인지 판단하기

- 다음 사람에게 차례를 넘기는 절차
 - 참가자가 몇 명인지는 number 변수에 저장, 참가자가 3명이면 number는 3
 - 참가자가 3명일 때는 1번 참가자부터 시작하고, 다음은 2번, 그다음은 3번이 됐다가 다시 1번 참가자로 순서가 돌아감
 - 몇 번째 참가자인지 #order 태그에
 - 현재 순서가 1이라면(#order 내부의 값 1) 여기에 1을 더한 값은 2로 number보다 작으니 다음 순서는 2
 - 현재 순서가 3이라면(#order 내부의 값 3) 여기에 1을 더한 값은 4로 number보다 크니 다음 순서가는 1

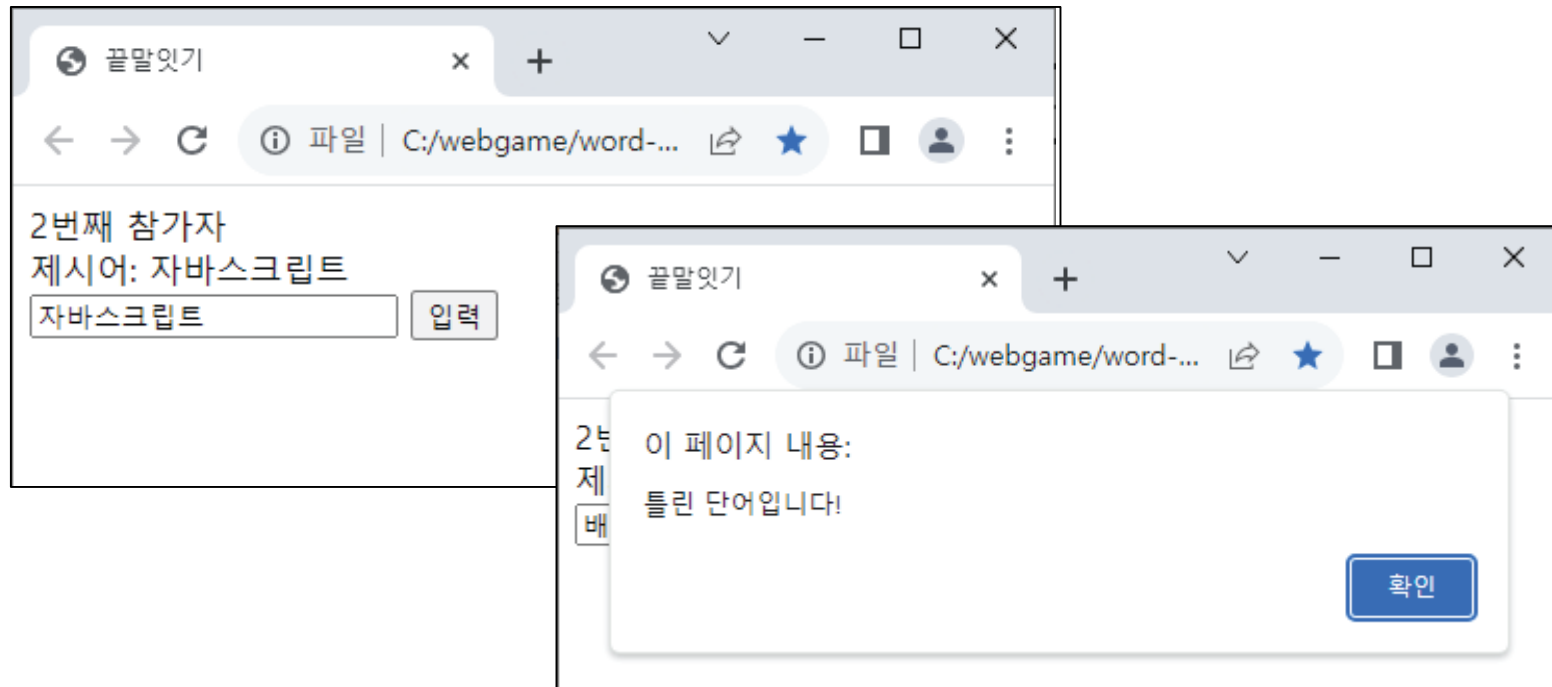
5.4 맞는 단어인지 판단하기

- 다음 사람에게 차례를 넘기는 절차를 반영해 수정한 끝말잇기 순서도



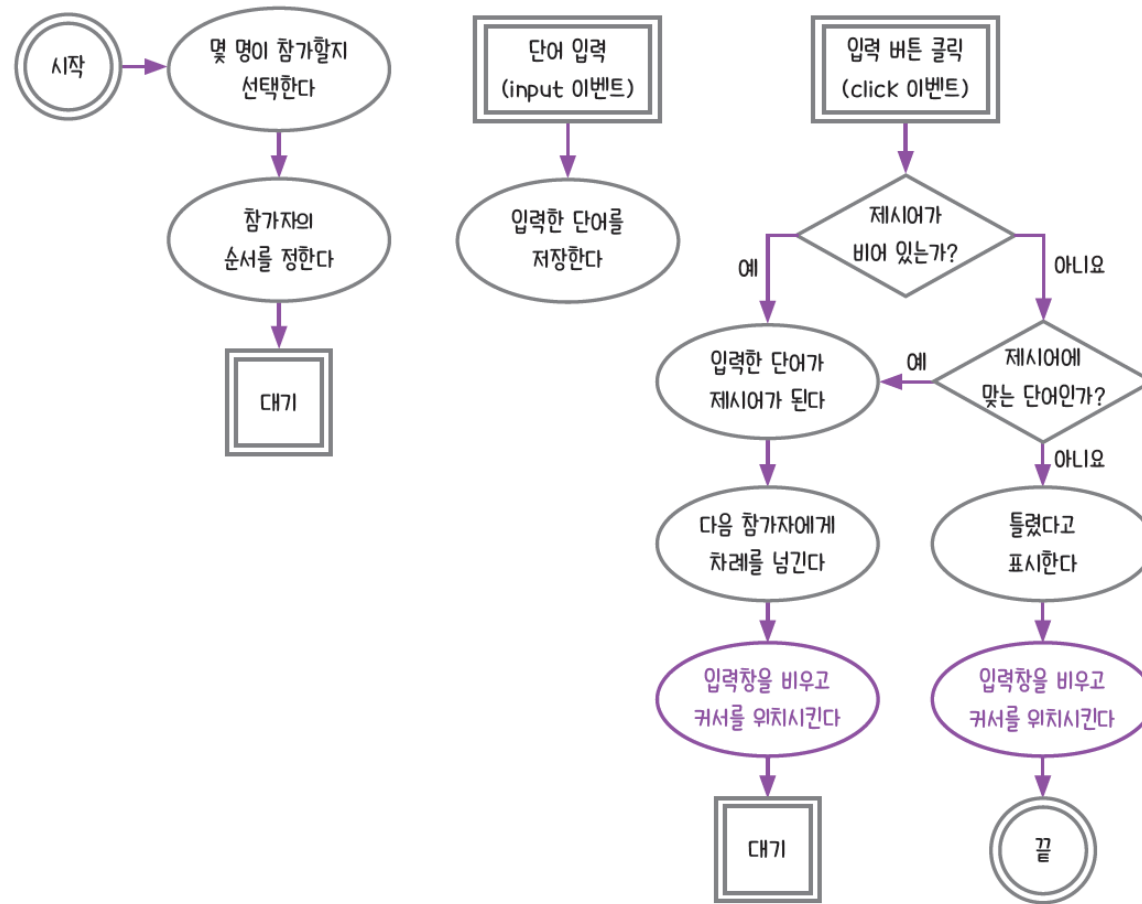
5.5 틀린 단어일 때 표시하기

- 제시어에 맞지 않는 단어를 입력하는 경우를 처리하는 절차
 - 맞지 않는 단어를 입력하면 alert() 메서드를 사용해 화면에 틀린 단어라고 표시



5.5 틀린 단어일 때 표시하기

- 틀린 단어 표시 절차를 반영해 수정한 끝말잇기 순서도



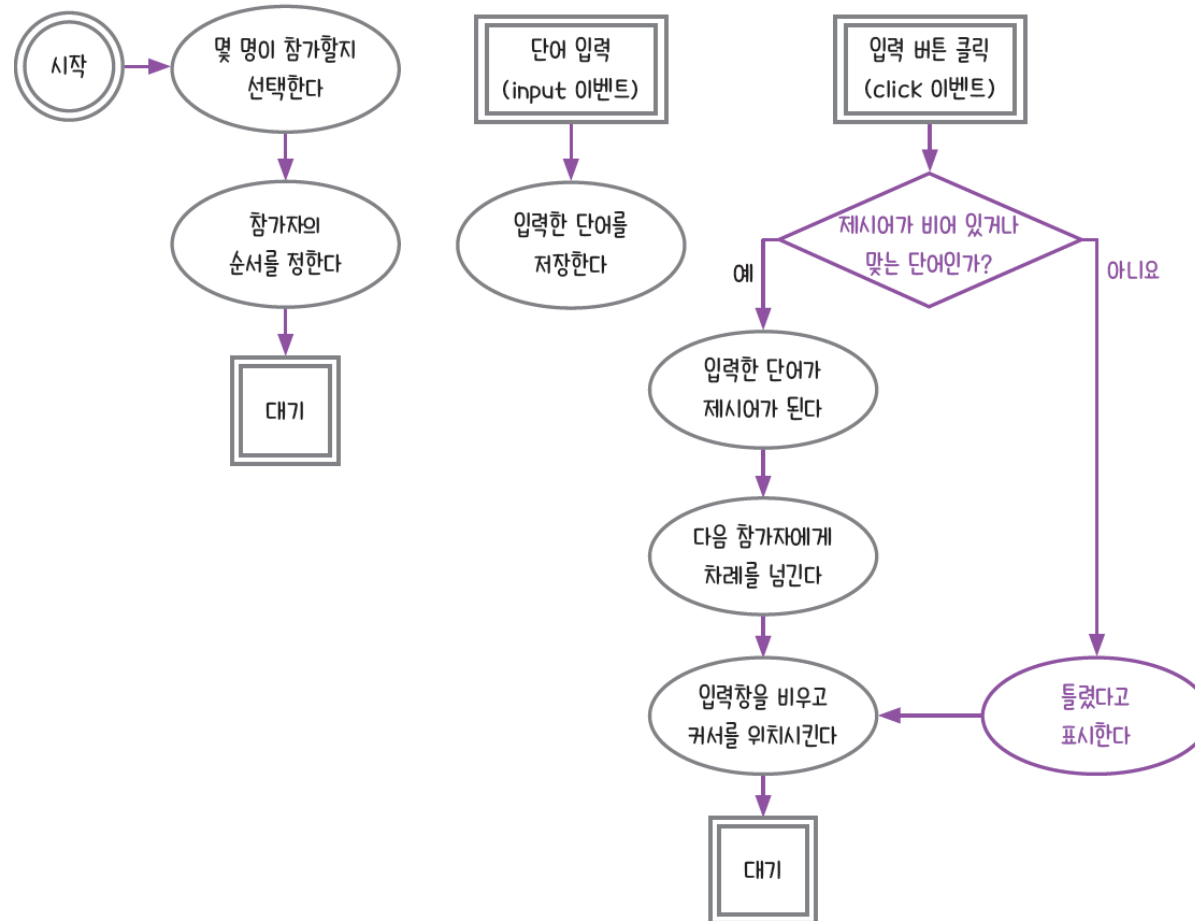
5.6 순서도 최적화하기

- 중복되는 부분을 찾아 최적화하기
 - ① 순서도에서 중복되는 절차 하나로 합치기
 - ② 순서도에서는 중복되지 않으나 코드에서 중복되는 부분 수정
 - ③ 단어가 틀렸더라도 다시 입력할 기회를 주도록 무한 반복, 순서도에서 끝표시만 제거
 - ④ 순서도에 판단하는 절차가 연이어 나올 때 합칠 수 있는 절차인지 판단해서 합치기

판단1	판단2	결과
제시어가 비어 있다	맞는 단어다	입력한 단어가 제시어가 된다
제시어가 비어 있다	틀린 단어다	입력한 단어가 제시어가 된다
제시어가 비어 있지 않다	맞는 단어다	입력한 단어가 제시어가 된다
제시어가 비어 있지 않다	틀린 단어다	틀렸다고 표시한다

5.6 순서도 최적화하기

- 최적화한 끝말잇기 순서도

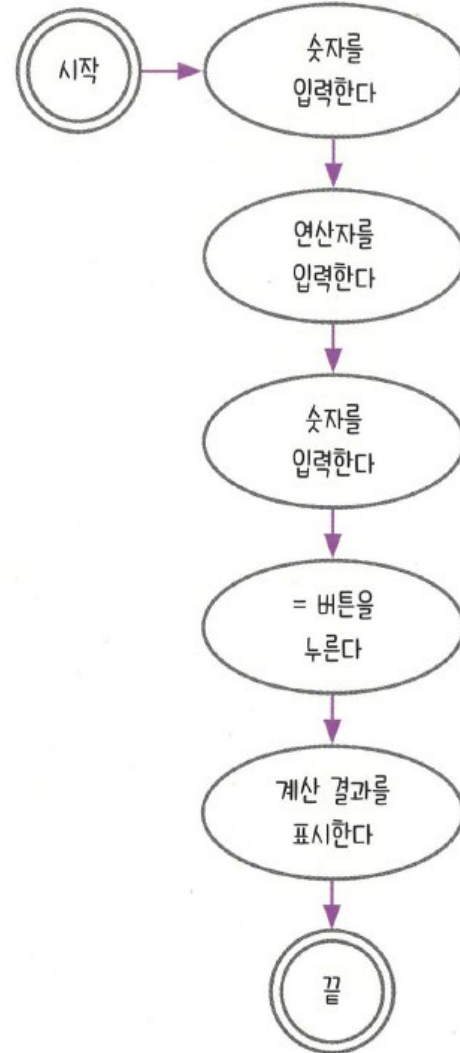


6장 함수 사용하기: 계산기

- 6.1 이 장에서 만드는 프로그램
- 6.2 계산기 화면 만들고 이벤트 달기
- 6.3 고차 함수로 중복 제거하기
- 6.4 중첩 if 문 줄이기
- 6.5 결과 계산하기
- 6.6 계산기 초기화하기

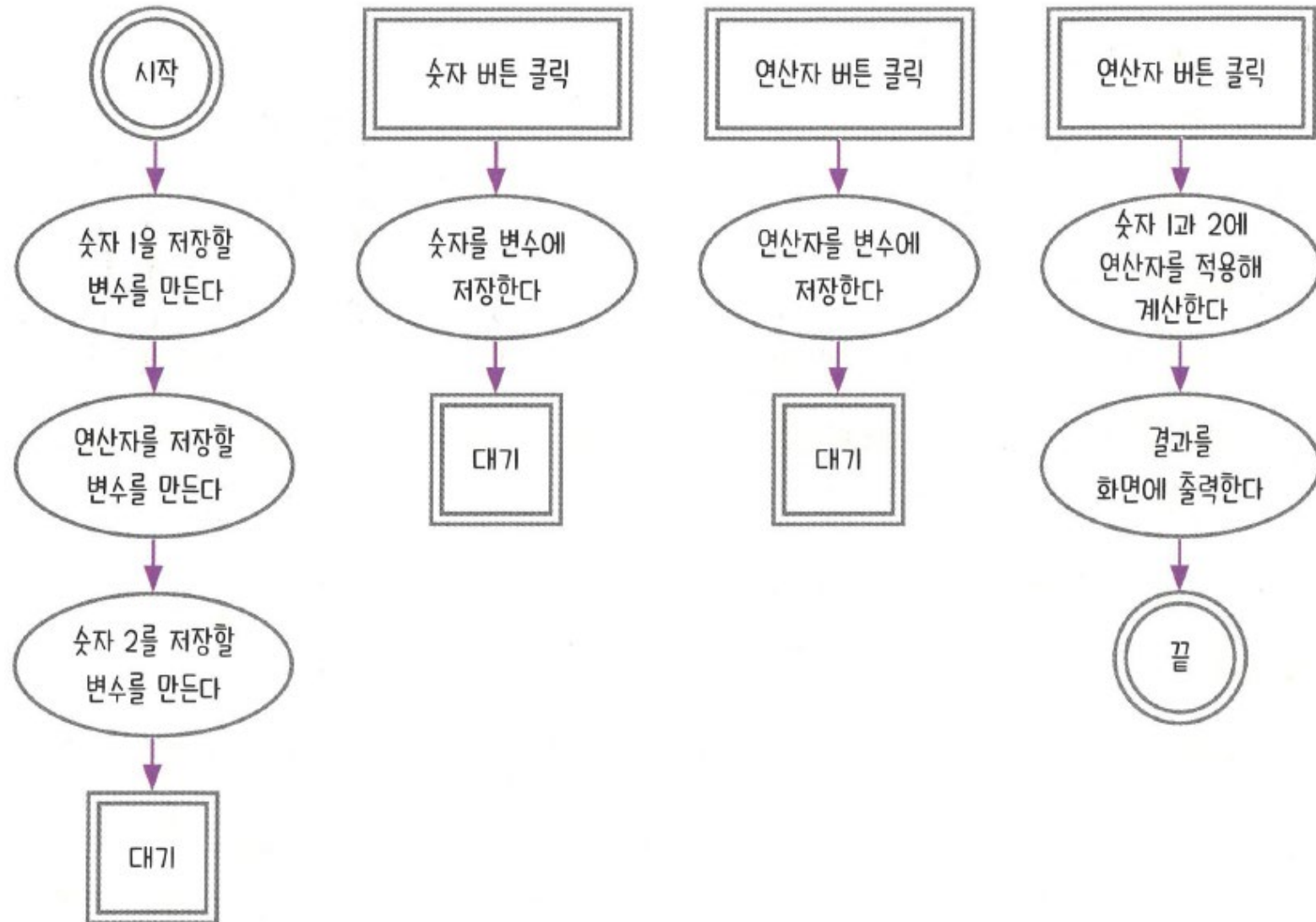
6.1 이 장에서 만드는 프로그램

- 계산기 순서도



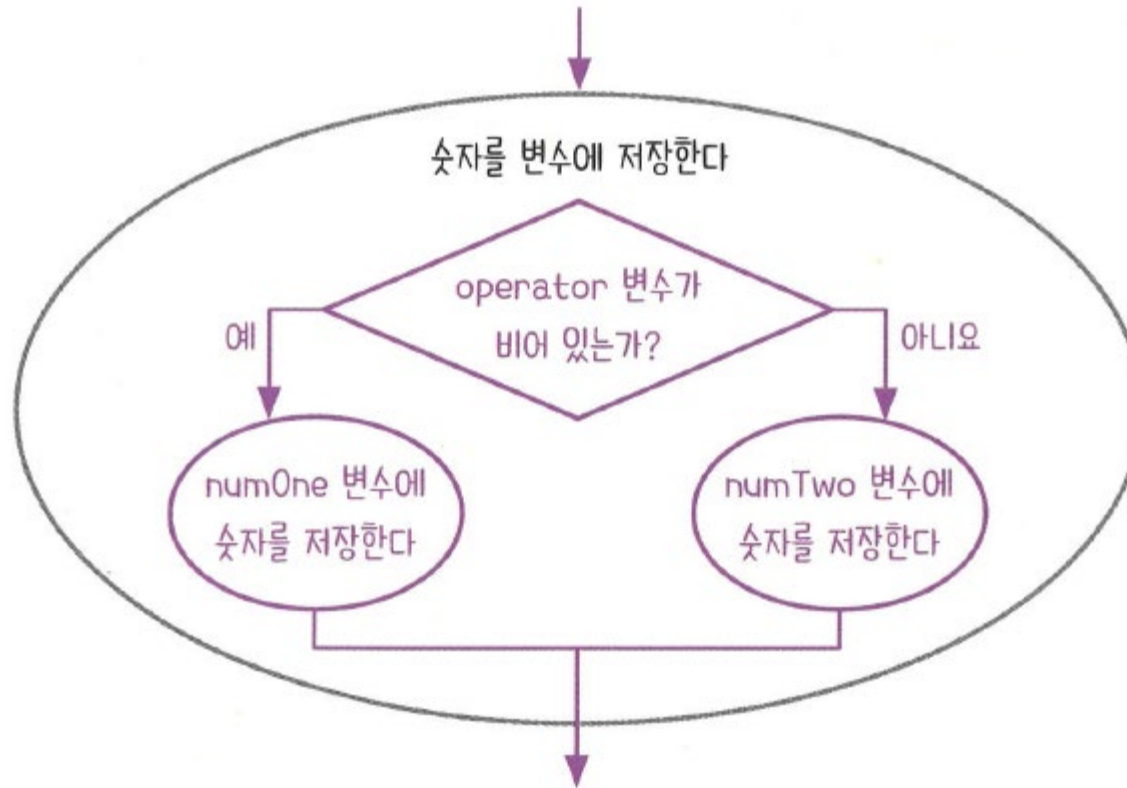
6.1 이 장에서 만드는 프로그램

- 계산기 순서도



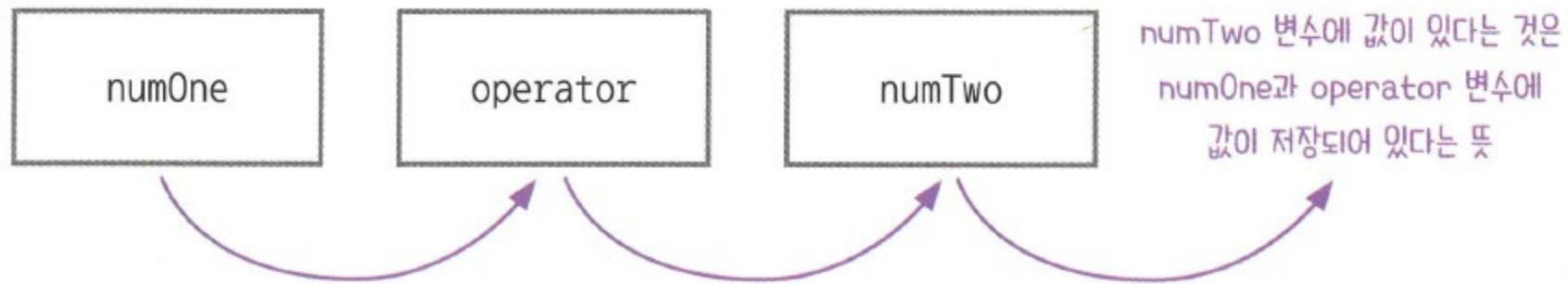
6.1 이 장에서 만드는 프로그램

- 계산기 순서도



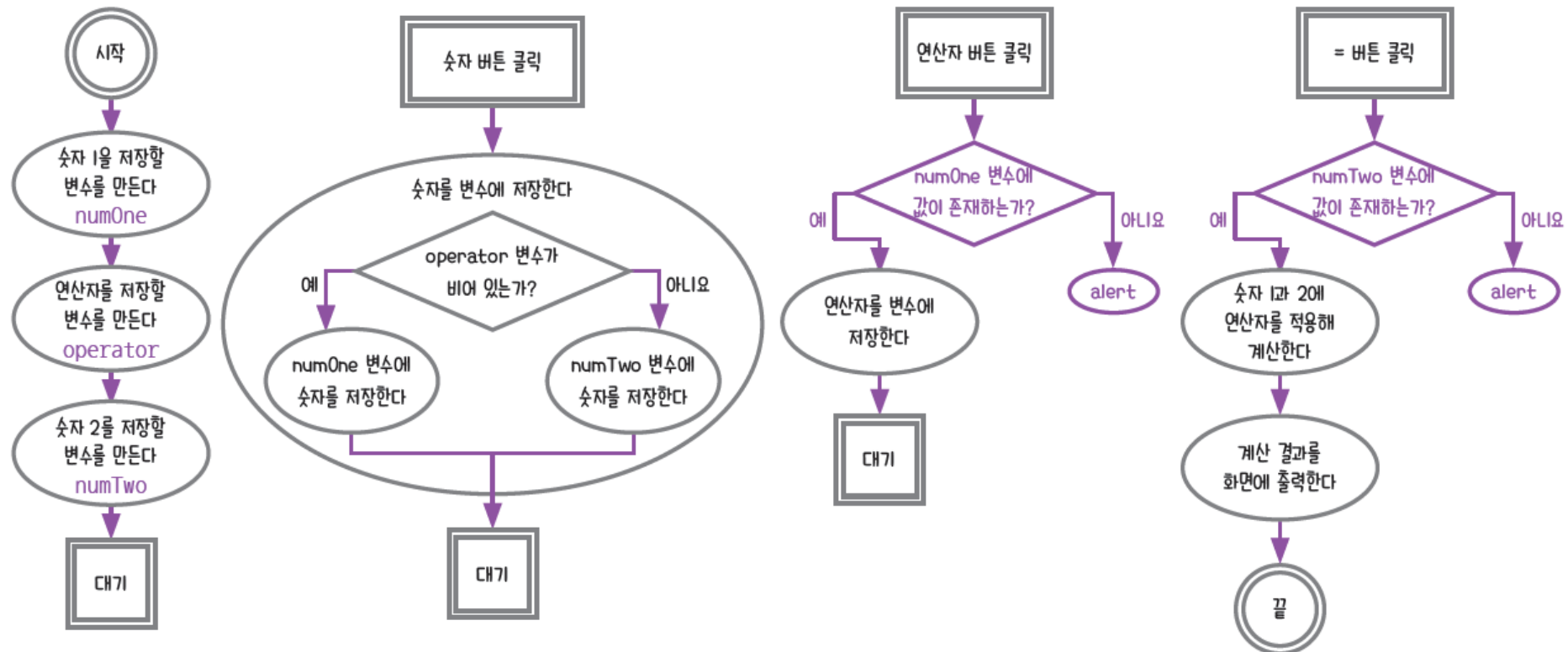
6.1 이 장에서 만드는 프로그램

- 계산기 순서도



6.1 이 장에서 만드는 프로그램

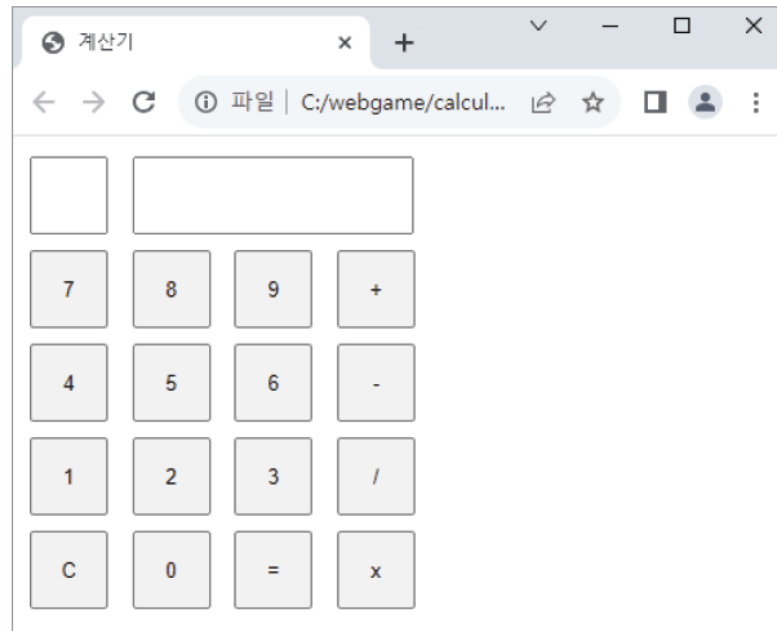
- 계산기 순서도



6.2 계산기 화면 만들고 이벤트 달기

6.2.1 계산기 화면 만들기

- 이 장에서 만드는 계산기 화면



6.2 계산기 화면 만들고 이벤트 달기

6.2.2 버튼에 이벤트 리스너 추가하기

- 각 버튼을 클릭하면 이벤트가 발생할 수 있게 이벤트 리스너 달기
- 반복되는 부분이 많으니 일부만 작성한 다음 규칙 찾기

6.3 고차 함수로 중복 제거하기

- 고차 함수로 중복을 제거해야 하는 이유
 - 내부 코드가 모두 같은 함수보다 대부분 다 비슷한데 특정 부분만 다른 함수가 더 많기 때문

6.4 중첩 if 문 줄이기

- 중첩 if 문을 제거하는 방법

- ① if 문 다음에 나오는 공통된 절차를 각 분기점 내부에 넣는다.
- ② 분기점에서 짧은 절차부터 실행하게 if 문을 작성한다.
- ③ 짧은 절차가 끝나면 return 문(함수 내부일 때)이나 break 문(for 문 내부일 때)으로 중단한다.
- ④ else 문을 제거한다(이때 중첩 하나가 제거된다).
- ⑤ 다음 중첩된 분기점이 나오면 ①~④의 과정을 반복한다

6.5 결과 계산하기

- = 버튼이 클릭될 때 결과를 출력하도록 #calculate 태그에 단 이벤트 리스너 작성
 - operator 변수는 +, -, *, / 같은 연산자를 문자열로 저장
 - 문자열은 연산자로 바뀌야 계산할 수 있음
 - 문자열은 자료형이고 연산자는 문법 기호라서 문자열을 연산자로 직접 바꿀 수는 없음
 - 문자열에 맞는 연산자를 사용하도록 조건문을 사용해 분기 처리

6.6 계산기 초기화하기

- #clear 태그(C 버튼)가 클릭되는 것을 감지할 이벤트 리스너 작성
 - 초기화한다는 것은 처음 상태로 되돌린다는 것
 - 변수와 함수 모두 초기 상태로 설정