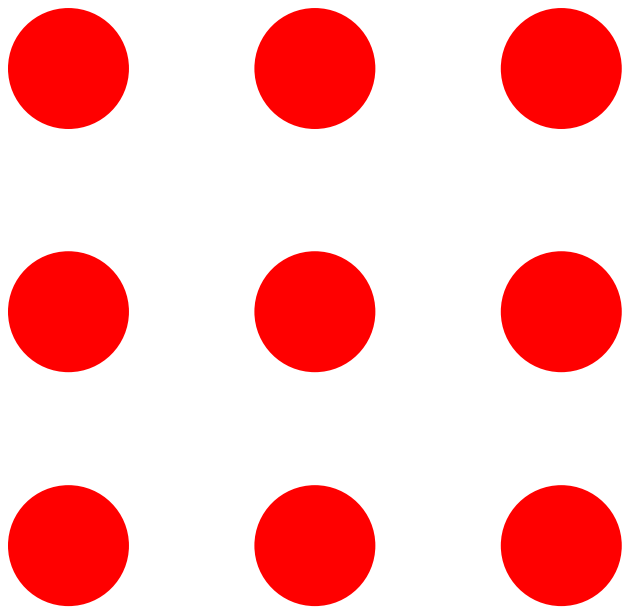




웹 프로그래밍

4개의 선을 이어 그려서 모든 점을 지나가도록 만들기





학습 내용

- I. 웹 표준 및 HTML 소개
- II. HTML
- III. CSS
- IV. JavaScript
- V. jQuery
- VI. ajaxs

학습목표

- 문제해결을 위한 알고리즘을 기술한다.
- 선택정렬, 버블정렬 알고리즘을 이해하고, javaScript로 구현한다.
- javaScript DOM 객체를 조작할 수 있다.
- javaScript의 익명함수, 화살표 함수를 이해한다.
- javaScript의 Number, String, JSON 객체의 유용한 메소드 사용하기

자바스크립트(4-1)

- 알고리즘 이해 및 표현

알고리즘 이해

- 알고리즘

- 어떤 문제를 해결하기 위한 **실행 절차, 방법, 명령어들의 집합**
- **논리적인** 처리 순서 정리

- 알고리즘 표현 방법

- 자연어 : 특별한 형식 없이 **일상적인 언어**로 알고리즘 표현
- 의사코드 : **문법적인 제약을 받지 않고, 프로그래밍 언어**로 알고리즘 표현
- 순서도 : **약속된 기호**를 이용하여 시각적으로 알고리즘 표현

알고리즘 이해

- 자연어 : 특별한 형식 없이 일상적인 언어로 알고리즘 표현



① 물 550ml를 끓입니다.



② 스프와 면을 넣고, 4분 동안 더 끓입니다.



③ 계란, 파 등을 기호에 따라 넣습니다.



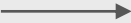
알고리즘 이해

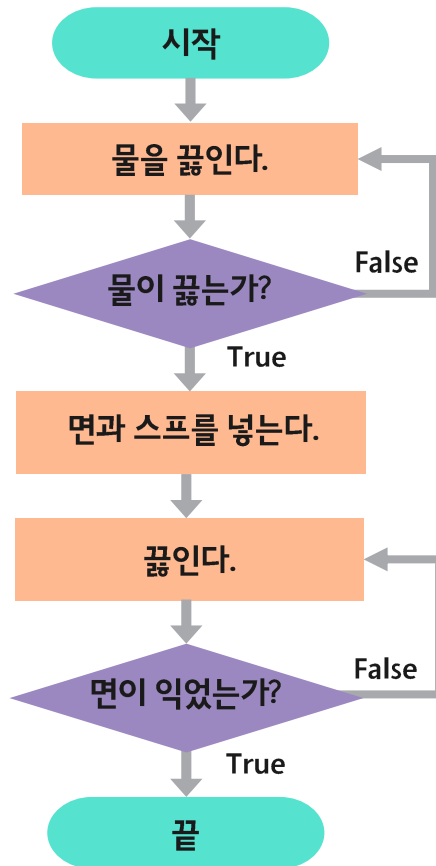
- 문법적인 제약을 받지 않고, 프로그래밍 언어로 변환하기 쉽게
알고리즘 표현

```
시작
물을 끓인다
if (물이 안 끓었으면){
    계속 끓인다
}
면과 스프를 넣는다
if (면이 안 익었으면){
    계속 끓인다
}
끝
```


순서도(FlowChart)

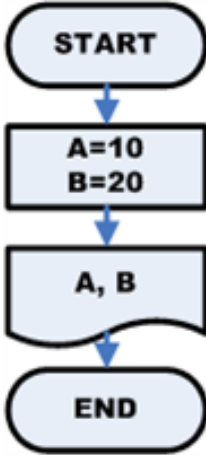
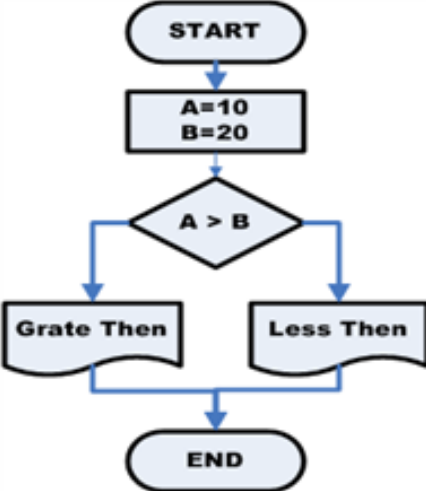
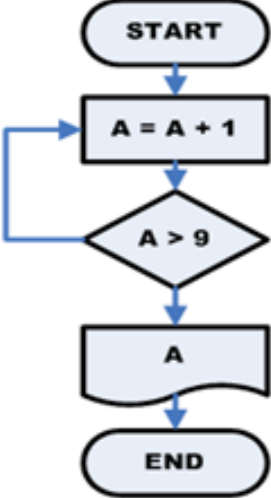
- 프로그램의 실행의 흐름을 한눈에 볼 수 있음.

기호	설명
	시작과 끝
	명령실행, 처리
	조건에 따라 판단 후, 분기
	일반적인 데이터의 입력 또는 출력 (키보드 입력, 화면 출력)
	순서도 연결
	실행 흐름

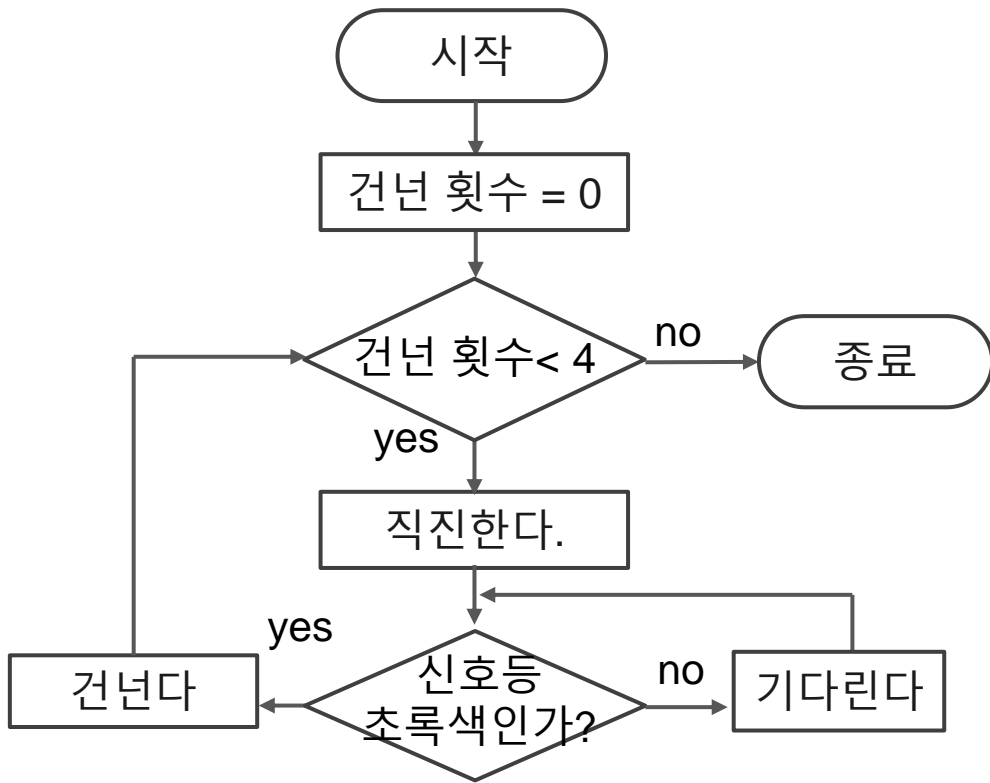
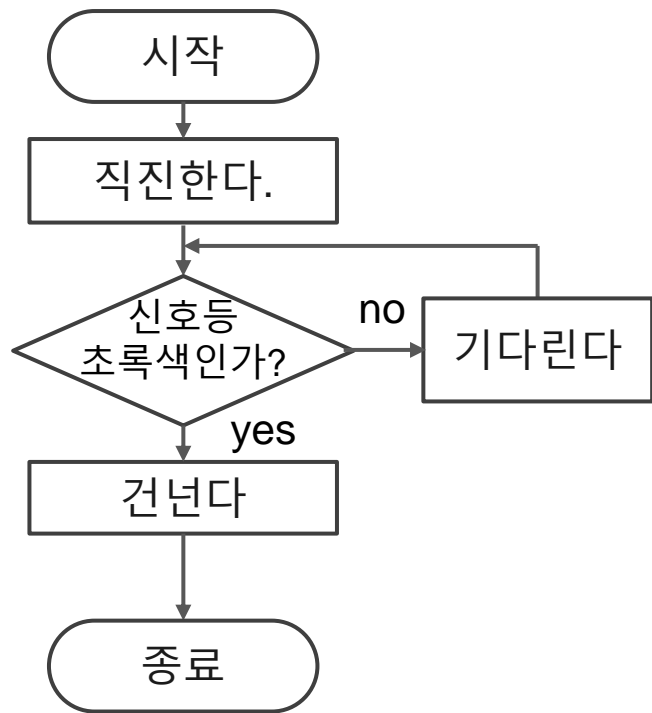


순서도

- 순서도 예

(1) 직선형	(2) 분기형	(3) 반복형
시작부터 종료 명령까지 단계적으로 진행되는 순서	조건에 따라 실행 순서를 달리하는 형태	조건을 만족할 때까지 일정한 내용을 반복 수행하는 형태
 <pre>graph TD; START([START]) --> A10B20[A=10
B=20]; A10B20 --> AB[A, B]; AB --> END([END]);</pre>	 <pre>graph TD; START([START]) --> A10B20[A=10
B=20]; A10B20 --> Decision{A > B}; Decision --> GoToEnd([Go to END]); Decision --> LessThan[Less Than]; LessThan --> END([END]);</pre>	 <pre>graph TD; START([START]) --> Aplus1[A = A + 1]; Aplus1 --> Decision{A > 9}; Decision --> A[A]; A --> END([END]); Decision --> Aplus1;</pre>

신호등 건너기 알고리즘



정렬 알고리즘 – 선택 정렬(오름차순)

1. 정렬되지 않은 인덱스의 맨 앞에서 부터, 그 값을 포함한 이후의 배열값 중 가장 작은 값을 찾기 위해 마지막 값까지 반복.
2. 기준값과 비교하여 가장 작은 값이면, 기준값과 스와핑
3. 다음 인덱스에서 위 과정을 반복함.

정렬 알고리즘 - 선택 정렬(오름차순)

시작

arr = 숫자 리스트 입력

len = 입력 숫자들의 전체 개수

for(기준 i ;전체 개수-1까지){

 제일 작은값 기억하기

 for(j=(i-1)부터 모든 숫자 반복){

 if (i의 값 보다 j의 값이 작다) {

 제일 작은 값으로 저장

 }

 if (기준값과 찾은 값이 다르면) 값 스와핑

}

정렬 값 콘솔에 출력

- 참고 : <https://hsp1116.tistory.com/33>

arr

안쪽 for j = 1 ↑ ↑ ↑ ↑ ↑ ↑ ↑

minIdx = 0

arr[0] ↔ arr[8]

arr

안쪽 for j = 2 ↑ ↑ ↑ ↑ ↑ ↑ ↑

minIdx = 1

arr[1] ↔ arr[3]

정렬 알고리즘 - 선택 정렬(오름차순)

arr	index	0	1	2	3	4	5	6	7	8
	value	9	11	12	23	14	16	20	15	11

바깥 for i = 2

안쪽 for j = 3

minIdx = 2

↑

↑

↑

↑

↑

↑

↑

2

8

arr[2] ↔ arr[8]

arr	index	0	1	2	3	4	5	6	7	8
	value	9	11	11	23	14	16	20	15	12

바깥 for i = 3

안쪽 for j = 4

minIdx = 3

↑

↑

↑

↑

↑

↑

3

4

8

arr[3] ↔ arr[8]

정렬 알고리즘 - 선택 정렬(오름차순)

arr	index	0	1	2	3	4	5	6	7	8
	value	9	11	11	12	14	16	20	15	23

바깥 for i = 4

안쪽 for j = 5

minIdx = 4

↑

↑

↑

↑

↑

4

arr	index	0	1	2	3	4	5	6	7	8
	value	9	11	11	12	14	16	20	15	23

바깥 for i = 5

안쪽 for j = 6

minIdx = 5

↑

↑

↑

↑

5

7

arr[5] ↔ arr[7]

정렬 알고리즘 - 선택 정렬(오름차순)

arr	index	0	1	2	3	4	5	6	7	8
	value	9	11	11	12	14	15	20	16	23

바깥 for i = 6

안쪽 for j = 7

minIdx = 6

↑

↑

↑

6

7

arr[6] ↔ arr[7]

arr	index	0	1	2	3	4	5	6	7	8
	value	9	11	11	12	14	15	16	20	23

바깥 for i = 7

안쪽 for j = 8

minIdx = 7

↑

↑

7

정렬 알고리즘 – 버블 정렬(오름차순)

- 매번 연속된 두개 인덱스를 비교하여, 정한 기준의 값을 뒤로 넘겨 정렬
- 비교시마다 큰 값이 뒤로 이동, 1바퀴 돌 때마다 큰 값이 맨 뒤에 저장
- (전체 배열의 크기 – 현재까지 순환한 바퀴 수) 만큼 반복
- 버블 정렬 기본 로직
 - 현재 index값과 바로 이전 index 값 비교
 - 만약 이전 index의 값이 더 크면, 현재 index의 값과 swapping
 - 현재 index의 값이 더 크면, 교환하지 않고 다음 두 연속 된 배열 값 비교
 - 위 과정을 전체 배열의 크기 – 현재까지 순환한 바퀴 수) 만큼 반복

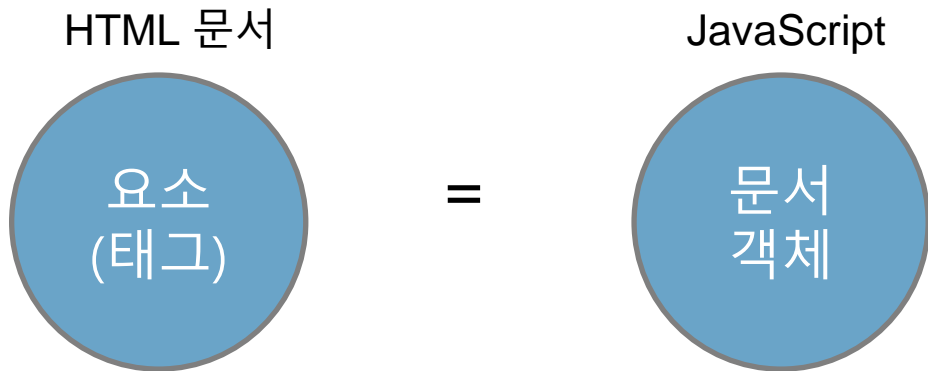
정렬 참고 : <https://hsp1116.tistory.com/33>

<https://godgod732.tistory.com/11?category=659135>

javaScript DOM 조작 및 함수

DOM 이해

- 문서 객체 모델(DOM; Document Objects Model)
 - HTML의 태그(요소:Element)를 자바스크립트에서는 문서 객체라고 함.
 - 문서를 객체로 조작한다는 것은 HTML요소들을 조작한다는 의미
 - 문서를 객체로 조합해서 만든 전체적인 형태를 DOM이라고 함.



HTML 문서 load 이벤트

DOMContentLoaded 이벤트

- 웹 브라우저가 HTML 문서를 모두 읽고 나서 실행하는 이벤트
- DOMContentLoaded 상태가 되었을 때 콜백 함수 호출

```
document.addEventListener('DOMContentLoaded', () => {  
  // callback 함수 정의  
  const $h1 = (text_str) => `<h1>${text_str}</h1>`  
  // callback 함수 실행  
  document.body.innerHTML += $h1('1번째 script 태그(head안)')  
  document.body.innerHTML += $h1('2번째 script 태그(head안)')  
})
```

[Tip] 콜백함수 : 함수를 변수에 저장하고, 그 변수를 함수의 인수로 사용하여 다른 함수의 매개변수에 전달하는 함수

addEventListener() 메소드

document.addEventListener('DOMContentLoaded', () => {})

- document 문서 객체의 DOMContentLoaded 이벤트가 발생했을 때, 매개변수로 지정한 콜백 함수를 실행하라는 의미
- HTML5에서 추가됨

정렬 알고리즘 - 웹으로 표현하기

- 선택정렬 web으로 표현하기 : 4-2-1_select_sort.html
 - 텍스트 박스로 숫자를 입력 받기
 - 입력된 숫자 배열로 저장
 - 정렬하고, 그 결과를 textarea에 출력하기
 - 선택 정렬 예시 확인.
- [문제 해결] 버블 정렬을 web으로 표현하기

Javascript 함수

Callback 함수

- Javascript는 함수도 하나의 자료형이므로, 매개변수로 전달할 수 있음.
- 매개변수로 전달하는 함수를 콜백(callback) 함수라고 함.

```
01 <script>
02 // 함수를 선언합니다.
03 function callThreeTimes (callback) {
04   for (let i = 0; i < 3; i++) {
05     callback(i) → callback이라는 매개변수는 함수이므로 호출할 수 있음
06   }
07 }
08
09 function print (i) {
10   console.log(` ${i}번째 함수 호출`)
11 }
12
13 // 함수를 호출합니다.
14 callThreeTimes(print)
15 </script>
```

0번째	함수	호출
1번째	함수	호출
2번째	함수	호출

Javascript 함수

Callback 함수 : 익명함수

```
01 <script>
02 // 함수를 선언합니다.
03 function callThreeTimes(callback) {
04   for (let i = 0; i < 3; i++) {
05     callback(i)
06   }
07 }
08
09 // 함수를 호출합니다.
10 callThreeTimes(function (i) {
11   console.log(`${i}번째 함수 호출`)
12 })
13 </script>
```

0번째	함수	호출
1번째	함수	호출
2번째	함수	호출

→ 익명 함수 사용하기

Javascript 함수

Callback 함수

- 콜백 함수를 활용하는 함수: `forEach()`
- `forEach()` 메소드는 배열이 갖고 있는 메소드로써 단순하게 배열 내부의 요소를 사용해서 콜백 함수를 호출

```
function (value, index, array) { }
```

```
01 <script>
02  const numbers = [273, 52, 103, 32, 57]
03
04  numbers.forEach(function (value, index, array)
05  {
06      console.log(`${index}번째 요소 : ${value}`)
07  })
08 </script>
```

0번째 요소 : 273

1번째 요소 : 52

2번째 요소 : 103

3번째 요소 : 32

4번째 요소 : 57

매개변수로 `value`, `index`, `array`를
갖는 콜백 함수를 사용

Javascript 함수

Callback 함수

- 콜백 함수를 활용하는 함수: map()
- map() 메소드는 콜백 함수에서 리턴한 값들을 기반으로 새로운 배열을 만드는 함수

```
01 <script>
02 // 배열을 선언합니다.
03 let numbers = [273, 52, 103, 32, 57]
04
05 // 배열의 모든 값을 제공합니다.
06 numbers = numbers.map(function (value, index, array) {
07     return value * value
08 })
09
10 // 출력합니다.
11 numbers.forEach(console.log)
12 </script>
```

매개변수로 value, index, array를
갖는 콜백 함수를 사용

매개변수로 console.log 메소드
자체를 넘김

실행 결과

74529	0	Array(5)
2704	1	Array(5)
10609	2	Array(5)
1024	3	Array(5)
3249	4	Array(5)

Javascript 함수

Callback 함수

```
<script>
// 배열을 선언합니다.
let numbers = [273, 52, 103, 32, 57]
// 배열의 모든 값을 제공합니다.
numbers = numbers.map(function
(value) {
  return value * value
})
// 출력합니다.
numbers.forEach(console.log)
</script>
```

함수 내부에서 value만 사용하므로
value만 매개변수로 넣음

Javascript 함수

Callback 함수

- 콜백 함수를 활용하는 함수 : filter()
- filter() 메소드는 콜백 함수에서 리턴하는 값이 true인 것만 모아서 새로운 배열을 만드는 함수

```
01 <script>
02  const numbers = [0, 1, 2, 3, 4, 5]
03  const evenNumbers = numbers.filter(function (value) {
04    return value % 2 === 0
05  })
06
07  console.log(`원래 배열: ${numbers}`)
08  console.log(`짝수만 추출: ${evenNumbers}`)
09 </script>
```

원래 배열: 0,1,2,3,4,5
짝수만 추출: 0,2,4

Javascript 함수

화살표 함수

- 화살표 함수는 **function** 키워드 대신 **화살표(=>)**를 사용하며, 다음과 같은 형태로 생성하는 간단한 함수

```
(매개변수) => {  
  불 표현식 || 불 표현식이 거짓일 때 실행할 문장
```

```
(매개변수) => 리턴값
```

Javascript 함수

화살표 함수

(매개변수) => {
 } 불 표현식 || 불 표현식이 거짓일 때 실행할 문장

(매개변수) => 리턴값

```
01 <script>
02  // 배열 선언
03  let numbers = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
04
05  // 배열의 메소드를 연속적으로 사용 가능
06  numbers
07    .filter((value) => value % 2 === 0 )
08    .map((value) => value * value)
09    .forEach((value) => {
10      console.log(value)
11    })
12 </script>
```

메소드
체이닝

0
4
16
36
64

Javascript 함수

- 선언적 함수

```
01 <script>
02 선언적함수()
03
04 function 선언적함수 () {
05     console.log('1번째 선언적 함수입니다.')
06 }
07 </script>
08 <script>
09 function 선언적함수 () {
10     console.log('2번째 선언적 함수입니다.')
11 }
12 </script>
13 <script>
14 선언적함수() → 블록 C
15 </script>
```

블록 A

블록 B

1번째 선언적 함수입니다.
2번째 선언적 함수입니다.

Javascript 객체

Number, String, Json

Javascript 객체 - Number

Number.toFixed()

- 숫자 N번째 자릿수까지 출력

```
> const i = 123.456789
```

```
undefined
```

```
> i.toFixed(2)
```

```
"123.46"
```

```
> i.toFixed(3)
```

```
"123.457"
```

```
> i.toFixed(4)
```

```
"123.4568"
```

- NaN 확인 : Number.isNaN()
- infinity 확인하기 : Number.isFinite()

Javascript 객체 – String

String 객체

- 문자열 양쪽 끝의 공백 없애기 : trim()

“ 헬로 자바스크립트 ” → “헬로 자바스크립트”

```
> stringA  
"
```

메시지를 입력하다보니 앞에 줄바꿈도 들어가고"

```
> stringB
```

" 앞과 뒤에 공백도 들어가고 "

```
> stringA.trim()
```

"메시지를 입력하다보니 앞에 줄바꿈도 들어가고"

```
> stringB.trim()
```

"앞과 뒤에 공백도 들어가고"

문자열 앞뒤 공백이 제거

[참조] 모질라 String 객체의 속성과 메소드

https://developer.mozilla.org/ko/docs/Web/JavaScript/Reference/Global_Objects/String

Javascript 객체 – String

String 객체

- 문자열을 특정 기호로 자르기 : split()
- 지정한 문자열을 기준으로 자라서 배열로 만들어 리턴 함.

```
> input = input.split('\n')  
["일자,달러,엔,유로", "02,1141.8,1097.46,1262.37",  
"03,1148.7,1111.36,1274.65",  
"04,1140.6,1107.81,1266.58",  
"07,1143.4,1099.58,1267.8",  
"08,1141.6,1091.97,1261.07"]
```

줄바꿈으로 자르기

```
> input = input.map((line) => line.split(','))  
[Array(4), Array(4), Array(4), Array(4), Array(4), Array(4)]  
> JSON.stringify(input, null, 2)
```

배열 내부의 문자열들을 쉼표로 자르기

```
"[[  
  "일자",  
  "달러",  
  "엔",  
  "유로"  
],
```

이하 생략

Javascript 객체 – JSON

JSON 객체

- 인터넷에서 문자열로 데이터를 주고 받을 때 보통(csv, xml, json) 사용
- 현재 가장 많이 사용한 방식 : json
 - 값을 표현할 때는 문자열, 숫자, 불, 자료형 만 사용할 수 있음.
 - 함수 사용 불가
 - 키(key)도 반드시 따옴표를 붙여야 함.

```
{  
  "name": "자바스크립트 활용",  
  "price": 18000,  
  "publisher": "땡땡미디어"  
}
```

```
[{  
  "name": "자바스크립트 기본",  
  "price": 18000,  
  "publisher": "땡땡미디어"  
},{  
  "name": "HTML5 웹 표준 입문",  
  "price": 26000,  
  "publisher": "웹아카데미"  
}]
```

Javascript 객체 – JSON

JSON 객체

- 자바스크립트 객체를 JSON 문자열로 변환할 때 : JSON.stringify()

```
01 <script>
02 // 자료 생성
03 const data = [{
04   name: '자바스크립트 기본',
05   price: 18000,
06   publisher: '땡땡미디어'
07 }, {
08   name: 'HTML5 웹 표준 입문',
09   price: 26000,
10   publisher: '웹아카데미'
11 }]
12
13 // 자료를 JSON으로 변환, 출력
14 console.log(JSON.stringify(data))
15 console.log(JSON.stringify(data, null, 2))
16 </script>
```

```
[{"name": "자바스크립트 기본", "price": 20000, "publisher": "땡땡미디어"},
{"name": "HTML5 웹 표준 입문", "price": 22000, "publisher": "웹아카데미"}]

[
  {
    "name": "자바스크립트 기본",
    "price": 20000,
    "publisher": "땡땡미디어"
  },
  {
    "name": "HTML5 웹 표준 입문",
    "price": 22000,
    "publisher": "웹아카데미"
  }
]
```

2번째 매개변수는 객체에서 어떤 속성만 선택해서 추출하고 싶을 때 사용하나 거의 사용하지 않으며, 일반적으로 null(아무 것도 없음)을 넣음

들여쓰기 2칸으로 설정

Javascript 객체 – JSON

JSON 객체

- JSON 문자열을 자바스크립트 객체로 변환 : `JSON.parse(json_data)`

```
01 <script>
02 // 자료 생성
03 const data = [{
04   name: '자바스크립트 기본',
05   price: 18000,
06   publisher: '땡땡미디어'
07 }, {
08   name: 'HTML5 웹 표준 입문',
09   price: 26000,
10   publisher: '웹아카데미'
11 }]
12 // 자료를 JSON으로 변환, 출력
13 const json = JSON.stringify(data)
14 console.log(json)
15 // JSON 문자열을 다시 자바스크립트 객체로 변환
16 console.log(JSON.parse(json))
17 </script>
```

```
[{"name": "자바스크립트 기본", "price": 20000, "publisher": "땡땡미디어"},
{"name": "HTML5 웹 표준 입문", "price": 22000, "publisher": "웹아카데미"}]
▼ Array(2) ⓘ
  ▶ 0: {name: '자바스크립트 기본', price: 20000, publisher: '땡땡미디어'}
  ▶ 1: {name: 'HTML5 웹 표준 입문', price: 22000, publisher: '웹아카데미'}
    length: 2
  ▶ [[Prototype]]: Array(0)
```

Bootstrap 및 웹 템플릿

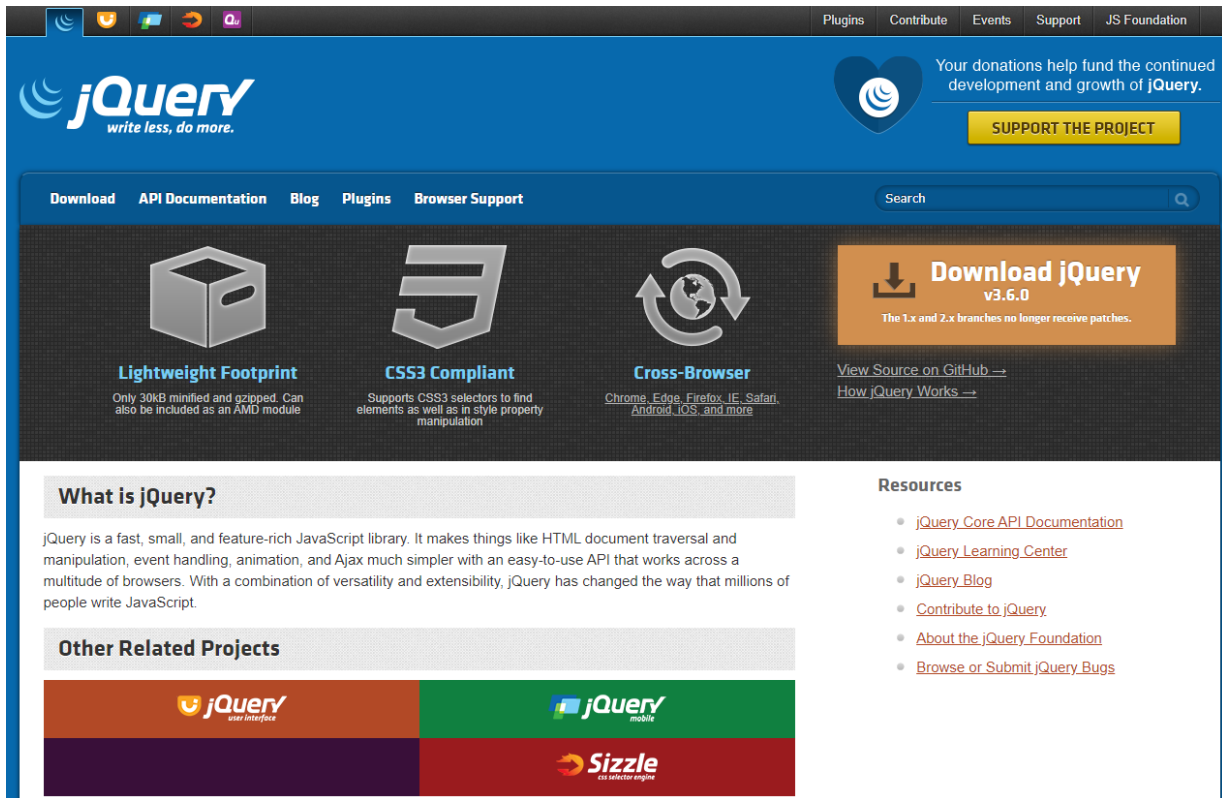
웹 템플릿 사이트

- <https://getbootstrap.com/>
- <https://bootstrapthemes.co/items/free-bootstrap-templates/>
- <https://themewagon.com/theme-price/free>
- <https://html5up.net/>

jQuery와 jQuery UI 활용하기

jQuery 살펴보기

- <https://jquery.com/>



The screenshot shows the jQuery website homepage. At the top, there's a navigation bar with links for Plugins, Contribute, Events, Support, and JS Foundation. Below this is the jQuery logo with the tagline "write less, do more." and a heart icon with the text "Your donations help fund the continued development and growth of jQuery." and a "SUPPORT THE PROJECT" button. The main content area features a search bar and several key features: "Lightweight Footprint" (Only 30KB minified and gzipped. Can also be included as an AMD module), "CSS3 Compliant" (Supports CSS3 selectors to find elements as well as in style property manipulation), and "Cross-Browser" (Chrome, Edge, Firefox, IE, Safari, Android, iOS, and more). A prominent orange box highlights "Download jQuery v3.6.0" with the note "The 1.x and 2.x branches no longer receive patches." and links to "View Source on GitHub" and "How jQuery Works". Below this, there are sections for "What is jQuery?" (describing it as a fast, small, and feature-rich JavaScript library), "Resources" (listing links to jQuery Core API Documentation, jQuery Learning Center, jQuery Blog, Contribute to jQuery, About the jQuery Foundation, and Browse or Submit jQuery Bugs), and "Other Related Projects" (showing logos for jQuery user interface, jQuery mobile, and Sizzle CSS selector engine).

jQuery write less, do more.

Your donations help fund the continued development and growth of jQuery.

SUPPORT THE PROJECT

Download API Documentation Blog Plugins Browser Support

Search

Lightweight Footprint
Only 30KB minified and gzipped. Can also be included as an AMD module

CSS3 Compliant
Supports CSS3 selectors to find elements as well as in style property manipulation

Cross-Browser
[Chrome, Edge, Firefox, IE, Safari, Android, iOS, and more](#)

Download jQuery v3.6.0
The 1.x and 2.x branches no longer receive patches.

[View Source on GitHub →](#)
[How jQuery Works →](#)

What is jQuery?

jQuery is a fast, small, and feature-rich JavaScript library. It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers. With a combination of versatility and extensibility, jQuery has changed the way that millions of people write JavaScript.

Resources

- [jQuery Core API Documentation](#)
- [jQuery Learning Center](#)
- [jQuery Blog](#)
- [Contribute to jQuery](#)
- [About the jQuery Foundation](#)
- [Browse or Submit jQuery Bugs](#)

Other Related Projects

jQuery user interface jQuery mobile Sizzle CSS selector engine

jQueryUI 살펴보기

- <https://jqueryui.com/>

The screenshot displays the jQueryUI website interface. At the top, there's a navigation bar with links for Plugins, Contribute, Events, Support, and JS Foundation. Below this is a header section featuring the jQuery logo, a heart icon with a jQuery logo inside, and a message about donations. A yellow button labeled 'SUPPORT THE PROJECT' is also present. The main navigation bar includes links for Demos, Download, API Documentation, Themes, Development, Support, Blog, and About, along with a search bar. The left sidebar contains two sections: 'Interactions' with links to Draggable, Droppable, Resizable, Selectable, and Sortable; and 'Widgets' with links to Accordion, Autocomplete, Button, Checkboxradio, and Controlgroup. The main content area is titled 'Accordion' and includes a description: 'Displays collapsible content panels for presenting information in a limited amount of space.' Below the description is a visual example of an accordion with two sections: 'Section 1' (expanded) and 'Section 2' (collapsed). To the right of the example is a list of 'Examples' including Default functionality, Collapse content, Customize icons, Fill space, No auto height, and Sortable.

jQuery user interface

Plugins Contribute Events Support JS Foundation

Your donations help fund the continued development and growth of jQuery.

SUPPORT THE PROJECT

Demos Download API Documentation Themes Development Support Blog About

Search

Interactions

- Draggable
- Droppable
- Resizable
- Selectable
- Sortable

Widgets

- Accordion
- Autocomplete
- Button
- Checkboxradio
- Controlgroup

Accordion

Displays collapsible content panels for presenting information in a limited amount of space.

Section 1

Mauris mauris ante, blandit et, ultrices a, suscipit eget, quam. Integer ut neque. Vivamus nisi metus, molestie vel, gravida in, condimentum sit amet, nunc. Nam a nibh. Donec suscipit eros. Nam mi. Proin viverra leo ut odio. Curabitur malesuada. Vestibulum a velit eu ante scelerisque vulputate.

Section 2

Examples

- Default functionality
- Collapse content
- Customize icons
- Fill space
- No auto height
- Sortable

웹 서버 호스팅

무료 호스팅 사이트

- <https://www.dothome.co.kr/>
- 회원가입, 3개월 무료
- ftp 서버 접속
- 자신을 소개하는 파일 웹서버에 올리기
- 알드라이브 설치, 다른 것 써도 됨

끝말잇기 게임 만들기

문제1)끝말잇기 게임 만들기

자전거
인도
리본
드라마



거인
도토리
본드
마중물

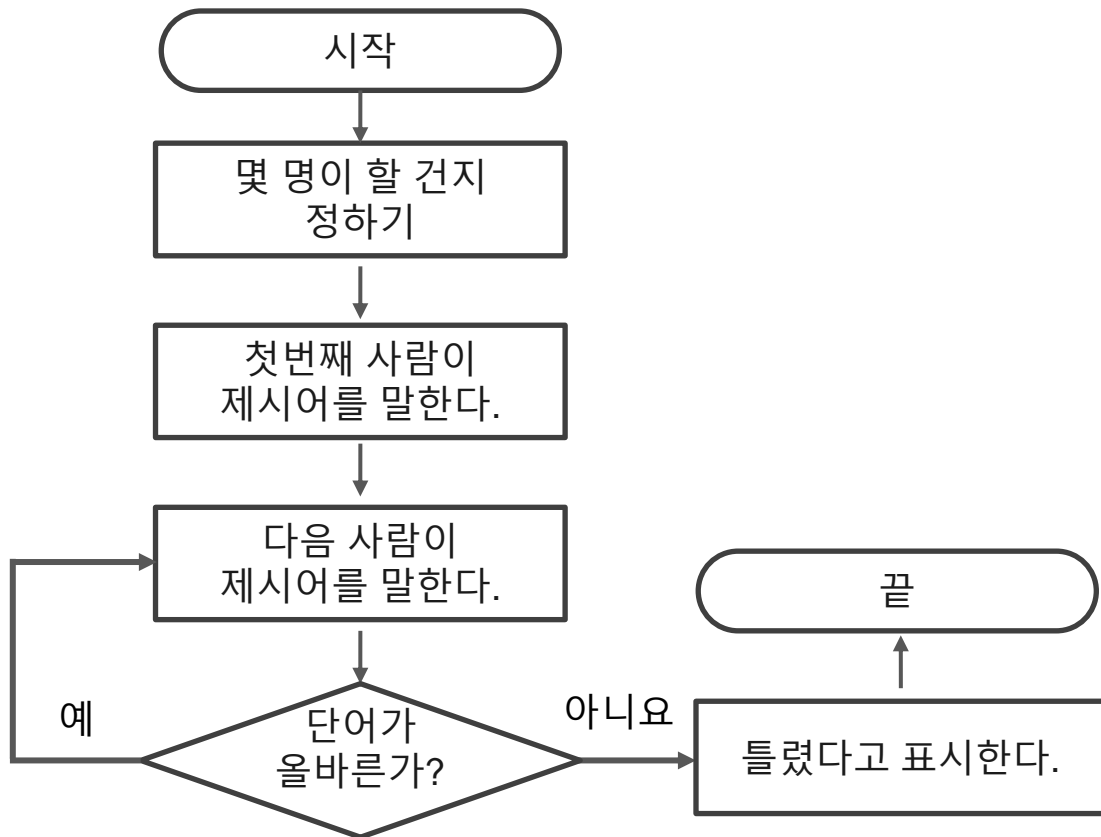


- 끝말 잇기 게임 프로그램 절차
- 몇 명이 진행 할 건지 입력
- 첫번째 사람이 제시어를 말한다.
- 다음 사람이 단어를 말한다.
- 단어가 올바른지 판단한다.
- 올바르면 다음사람이 이어서 말한다.
- 올바르지 않다면 틀렸다고 표시한다.

문제1)끝말잇기 게임 만들기 - 순서도 그리기

- HTML 화면 요소(입력 요소, 버튼, 글자)
- CSS 디자인
- 자바스크립트 핵심 기능: 함수 만들기

문제1) 끝말잇기 게임 만들기 - 순서도 그리기



문제1)끝말잇기 게임 만들기 – 순서도 그리기

- word를 입력하면 어떤 과정을 거치나?
- 첫번째 입력
 - input 요소에 단어 입력하면,
 - 버튼 클릭
 - 입력 단어 word 변수에 저장
 - input 요소 내용 지우기

문제1)끝말잇기 게임 만들기

- word-relay.html 실행화면

2번째 참가자

제시어: 림자

입력

문제해결 프로젝트

문제분석, 알고리즘 표현, 업그레이드