

# 5주차 강의

## 자바웹프로그래밍(1)

강사 : 최도현

# 트렌드 분석

## •웹 서비스 및 개발 트렌드

기술 트렌드 - 자바스크립트



# 자바스크립트 변화

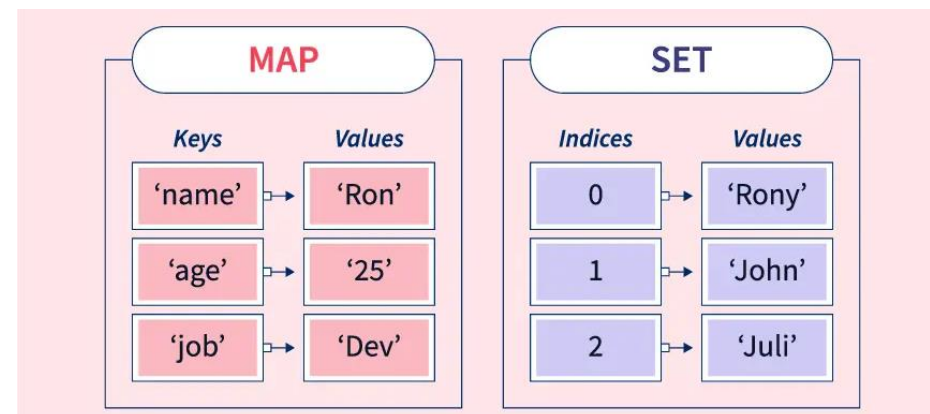
- ECMAScript 6 이후(JS 표준, 모던 자바스크립트)
  - 2015 이후 ES6 이상
  - 최근 웹 사이트 약 97% 지원
- 웹 사이트에서(브라우저 별) 확인
  - 기능 지원 차이가 존재
  - <https://caniuse.com/?search=es6>
- 어떤 변화가 있었는가?
  - LET, CONST** (지난주)
  - 변수 선언 방식
  - 클래스 및 자료형 지원
  - IMPORT, EXPORT
    - 변수, 함수, 클래스(모듈 시스템)
  - MAP/SET 등**
    - 새로운 자료구조 지원



판	출판 연도	새롭게 추가된 주요 문법
6	2015	<ul style="list-style-type: none"> <li>class 선언</li> <li>모듈</li> <li>for...of 반복문</li> <li>제너레이터</li> <li>화살표 함수</li> <li>변수 선언 키워드 let과 const</li> <li>이진 데이터와 TypedArray</li> <li>맵, 셋, 위크맵</li> <li>프라미스</li> <li>Proxy와 Reflect</li> <li>태프린 리터럴</li> </ul>
7	2016	<b>JS의 변화 및 핵심 기능들</b> <ul style="list-style-type: none"> <li>구조 분해(destructuring) 패턴</li> <li>지수 계산을 위한 연산자 **</li> <li>비동기 프로그래밍을 위한 await와 async 키워드</li> </ul>
8	2017	<ul style="list-style-type: none"> <li>제너레이터와 프라미스를 위한 async/await와 구조체</li> <li>atomic과 concurrency관련 휘처</li> </ul>
9	2018	<ul style="list-style-type: none"> <li>나머지/전개(rest/spread) 문법 ...</li> <li>비동기 이터레이션</li> <li>Promise.prototype.finally()</li> </ul>
10	2019	<ul style="list-style-type: none"> <li>Array.prototype.flat을 비롯한 배열 메서드</li> <li>Object.fromEntries</li> </ul>

# MAP과 SET 자료구조

- 새로운 자료구조 살펴보기
  - MAP : 키와 값(쌍), 고유 키, 순서 X
    - 키 값 제한 없음, 메모리 사용량 증가
  - SET : 중복 제거(집합), 순서 유지,
    - 키 값 제한 없음, 속도가 빠름
- 기존 자료구조 활용
  - 객체(Object) : 키(문자 전용), 순서 유지
    - 키 타입 검증 필요
    - 멤버 변수 형태 저장(메모리 증가)
  - 배열(Array) : 중복 O, 크기/순서 제어 어려움
- 활용 예) 데이터 유형, 순서 고려
  - MAP : 사용자 정보 등 고유 정보 저장
  - SET : 단순 값만 저장, 중복 제거



## 복잡한 데이터 제어/관리에 유용

구분	Map	Set
저장 가능한 값	키-값 쌍 (key-value pair)	중복 없는 값
순서	삽입 순서 유지	삽입 순서 유지
키 타입	모든 타입 가능 (심볼 포함)	모든 타입 가능 (심볼 포함)
값 타입	모든 타입 가능	모든 타입 가능
탐색 방식	키 기반	값 기반

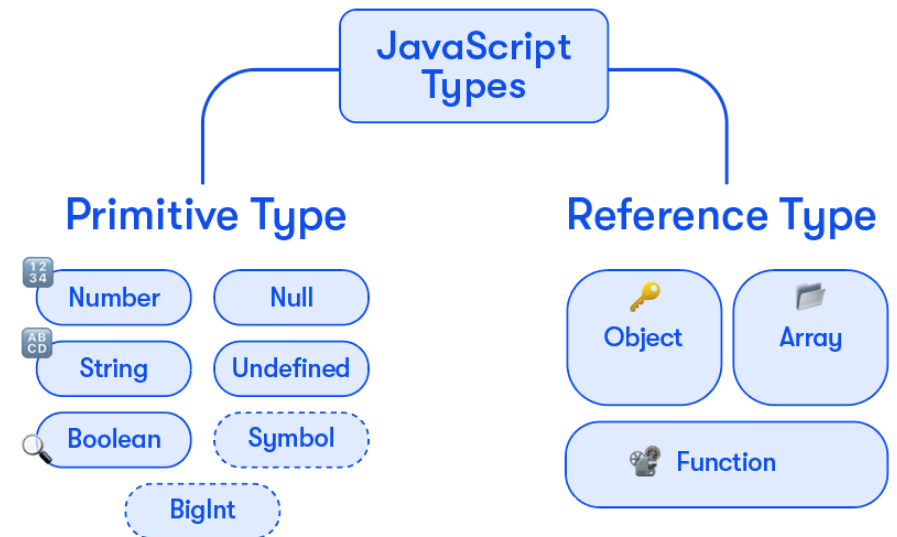
# PART1

## • 자바스크립트 기초

자바스크립트 - 데이터 타입

검색창(확장)

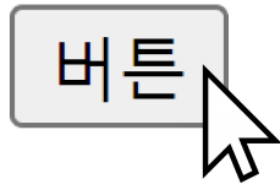
팝업창, 마우스 호버



# 지난주 내용 살펴보기

- 수정된 LOL 메인 화면, 자바스크립트 추가
  - 검색창 및 메시지 창 기능 확인

1. 사용자가 Page 에서 어떠한 상호작용을 함



2. 이벤트(Event) 발생



**자바스크립트 연동, 이벤트 동작 살펴보기**

4. 페이지에서 어떠한 변화가 생김



3. 자바스크립트 코드 실행

```
function myCode() {
  // do something
  return;
}
```

항목	태그 이름/설명
자바스크립트를 추가하는 태그는?	
파일로 작성된 자바스크립트는 어디에 추가되는가?	
자바스크립트가 제어하는 트리구조 문서를 무엇이라 하는가?	
자바스크립트 변수 선언 방식 3가지	
ES6 표준 이후 안전성을 위해 추가된 변수 선언 방식은?	
문서의 최상위 문서의 객체의 이름은?	
문서의 객체 식별자를 인식하는 메서드는?	
마우스 이벤트 중 더블 클릭의 이름은?	

# 자바스크립트 - 데이터 타입



- JS의 데이터 타입 종류
  - 원시 타입, 참조 타입 구분
    - 기존 언어와 차이점 : 타입 지정 없음
    - ES 표준 업데이트 이후 일부 추가됨
- JS 폴더에 파일 생성
  - data\_type.js 를 생성 후 작성한다.
    - head에 파일을 <script> 태그로 연동
- 원시타입 : 숫자, 문자, Boolean 등
  - 직접 선언해서 값을 확인해보자.
    - f12 개발자 모드 콘솔
  - **undefi** 와 **null**은 반드시 직접 값 확인
    - 참고 : typeof 값 (데이터 타입 확인)

```
let number = 5;  
let str = '문자열 입력'; // " "도 묶음 가능  
let prime = 1.5123;  
let is_ok = true; // 참  
let is_not = false; // 거짓  
let undefi; // 변수 이름만, 초기화 x  
let empty = null; // 비어 있음
```

```
console.log(undefi, empty); // 여러 개 출력
```

구분	데이터 타입	설명
원시 타입 (primitive type)	Boolean	논리적 값으로 true, false
	null	존재하지 않거나 유효하지 않은 주소 표시
	undefined	선언 후 값을 할당하지 않은 변수
	number	정수, 실수 등의 숫자, 정수의 한계는 $\pm 2^{53}$
	string	빈 문자열이나 글자들을 표현하는 문자열
	symbol	문자열과 함께 객체 property로 사용, ES6에 추가
객체 타입 (object type)	object	두개 이상의 복잡한 객체 저장 가능

# 자바스크립트 - 데이터 타입

- JS 폴더에 파일 수정
  - data\_type.js 에 추가 작성한다.
- 객체타입 : 객체, 배열
  - 직접 선언해서 값을 확인해보자.
    - f12 개발자 모드에서 콘솔
  - 심볼의 경우 전용 함수 지원
    - 중첩되지 않는 식별값 생성
  - 참고 : 객체는 참조 타입이다.
    - 데이터와 주소는 분리됨, 즉 주소만 저장하는 변수
    - 함수도 객체로 분류된다.

```
const sym1 = Symbol('test'); // 심볼 함수로 값 생성
let symbolVar1 = sym1; // 변수 초기화
```

```
const airline = ["비행기", 320, "airbus", ["V1", true]];
// 다양한 데이터 배열
```

```
// 빈 객체 생성
const obj1 = {};
```

```
// 속성을 추가하여 객체 생성
const obj2 = {
  name: "John Doe",
  age: 30,
  isMale: true,
};
```

```
console.log(symbolVar1.toString()); // 문자열 변환 출력
console.log(obj1, obj2, airline); // 여러 개 출력
```

	기본형	참조형
할당 방식	값을 그대로 할당	값이 저장된 주소값을 할당 (참조)



# 자바스크립트 - 데이터 타입

- JS 폴더에 파일 수정
  - data\_type.js 에 추가 작성한다.
- 자료구조 : MAP, SET
  - 직접 선언해서 값을 확인해보자.
    - f12 개발자 모드에서 콘솔
  - 두 자료구조를 조합하여 활용
    - MAP : 이름은 키, 아이디와 패스워드 조합
    - SET : 이름 중복 X
  - 참고 : 문자열과 변수를 같이 조합하여 출력
    - 템플릿 리터럴이라고 한다.

```
const users = new Map(); // 사용자 정보 Map 객체 생성
users.set("user1", { // 사용자 정보 추가
  id: 1, password: "password123",
});
users.set("user2", {
  id: 2, password: "password456",
});
```

```
// Map 객체의 모든 사용자 정보 반복 출력
for (const [username, user] of users) {
  console.log(`사용자 이름: ${username}`, `ID: ${user.id}`);
  console.log(`비밀번호: ${user.password}`);
}
```

```
// Set 객체 활용 (예), 이름만 저장할 Set 객체 생성
const usernames = new Set();
```

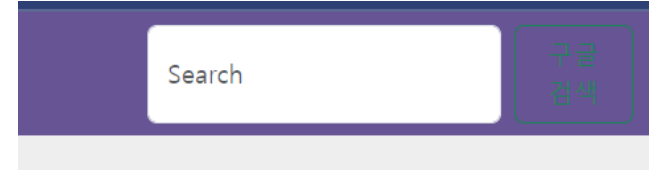
```
usernames.add("user1"); // 사용자 이름 추가
usernames.add("user2");
```

```
// Set 객체의 모든 사용자 이름 반복 출력
for (const username of usernames) {
  console.log(`사용자 이름: ${username}`);
}
```

# LOL 웹 사이트 - 검색창(확장)

- 검색 기능을 추가 구현한다.
  - 입력한 텍스트를 구글에서 검색
- index.html 코드를 수정한다.
  - 기존 form 태그를 수정한다.
    - submit 하면 추가 구글 검색 함수를 호출

사용자 입력 영역 = form



```
<form action="http://www.google.com/search">  
  <div>  
    Let's search Google:  
    <input name="q" />  
    <input type="submit" />  
  </div>  
</form>
```

Let's search Google:

```
<form class="d-flex" role="search" onsubmit="return googleSearch();">  
  <input class="form-control me-2" id="search_input" name="q" type="search" placeholder="키워드 입력" aria-label="Search">  
  <button class="btn btn-outline-success" id="search_btn" type="submit">검색하기</button>  
</form>
```

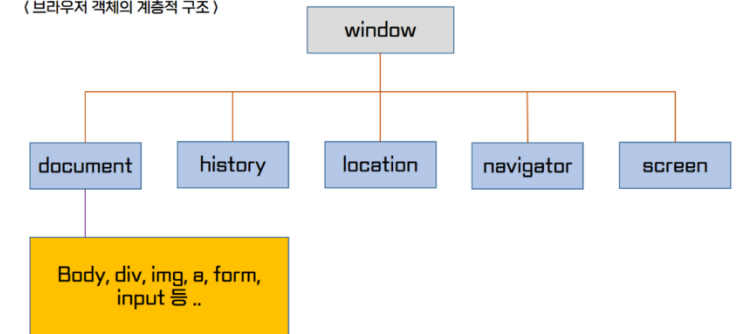
- 구글은 검색 파라미터 이름이 q로 고정되어 있음
- 기존 자바스크립트와 차이점?
  - 이벤트 등록 방식이 아님
  - form 태그에 이벤트 발생

속성	속성값	기능
action	URL 주소	사용자 입력 전송할 위치(처리 프로그램의 주소)를 지정
method	get/ post	사용자 입력 정보의 전송 방식을 지정 get : url 주소에 입력 정보(변수명+변수값)를 추가해서 문자열로 보내는 방식 post : url 주소가 아닌 내부적(트랜잭션)으로 보내는 방식 보안성이 높고 전송량에 제한이 없음
name	문자열	폼의 이름

# LOL 웹 사이트 - 검색창(확장)

## 기본 브라우저 객체 구조

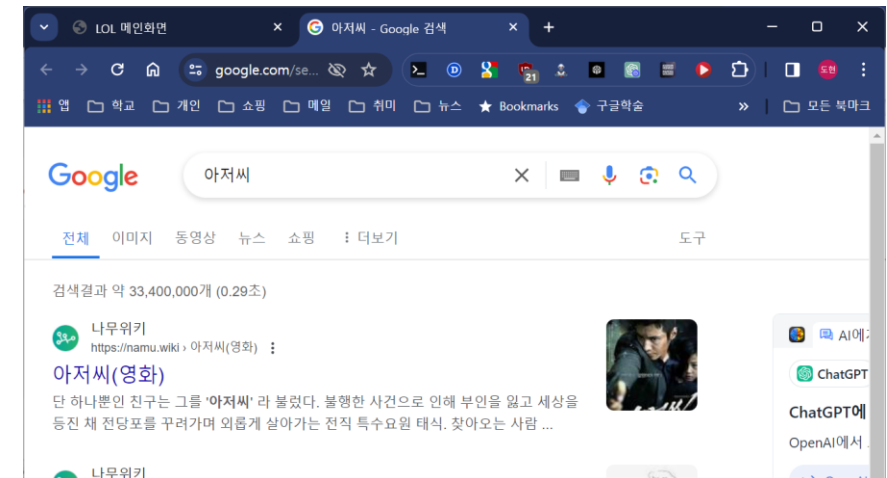
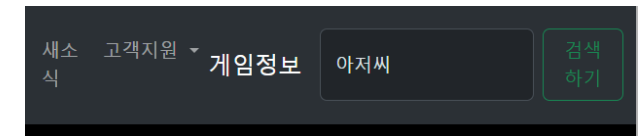
(브라우저 객체의 계층적 구조)



- js 폴더의 search.js 코드를 수정한다.
  - 구글 검색을 위한 함수를 추가한다.
    - 입력한 텍스트를 인식하고 검색어로 설정

```
function googleSearch() {  
  const searchTerm = document.getElementById("search_input").value; // 검색어로 설정  
  const googleSearchUrl = `https://www.google.com/search?q=${encodeURIComponent(searchTerm)}`;  
  // 새 창에서 구글 검색을 수행  
  window.open(googleSearchUrl, "_blank"); // 새로운 창에서 열기.  
  return false;  
}
```

- 주소 url에 포함되는 문자는 변환 필요
  - 서버에서 올바르게 해석 x
  - encodeURIComponent UTF-8 인코딩 처리
- 검색어 입력 후 구글 검색 확인
  - 참고 : url에 검색어가 삽입됨(get 방식 특징)



# LOL 웹 사이트 - 팝업창

- 팝업 창 기능을 추가 구현한다.
  - 메인 접속시에 팝업창(공지, 광고 등)
- popup 폴더를 생성한다.
  - popup.html 파일을 생성하고 작성한다.
- Js 폴더 pop\_up.js를 작성한다.
  - <head> 태그에 <script> 태그로 삽입한다.

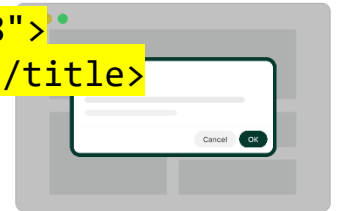
```
function pop_up() {  
    window.open("../popup/popup.html", "팝업테스트", "width=400, height=300, top=10, left=10");  
}
```

- 가로 400, 세로 300, 위 10, 왼쪽 10 여백
- 윈도우.open() 함수는 창을 열거나 닫는(close) 전용 함수

## POPUP과 비슷한 MODAL이 있다.

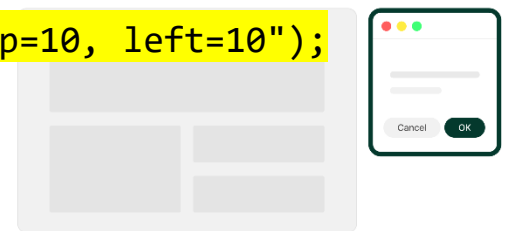
```
<!DOCTYPE html>  
<html>  
  <head>  
    <meta charset="UTF-8">  
    <title>팝업창 테스트</title>  
  </head>  
  <body>  
    <h1>팝업창 확인</h1>  
  </body>  
</html>
```

**Modal** 하나의 브라우저 + 레이어 창



© 2023 Imweb Corp., All rights reserved.

**Pop-up** 두개의 브라우저 창



© 2023 Imweb Corp., All rights reserved.

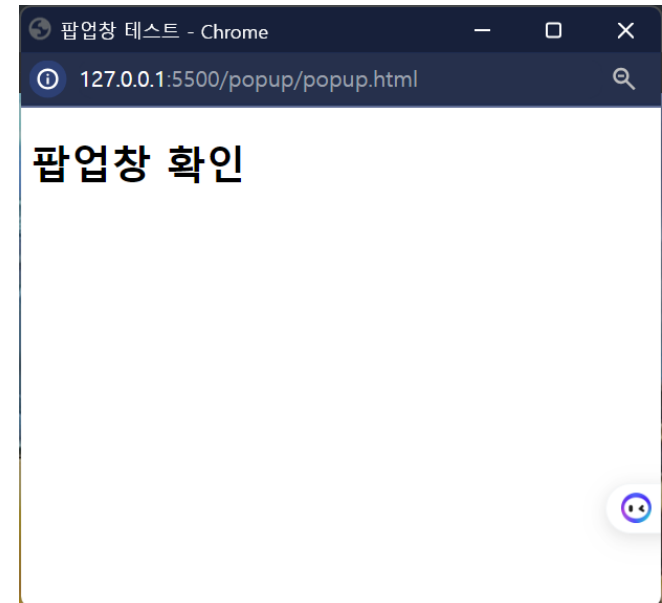
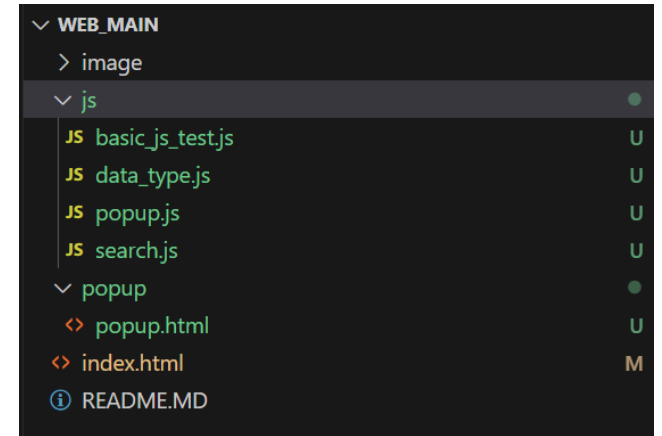
# LOL 웹 사이트 - 팝업창

- index.html 코드를 수정한다.
  - 기존 body 태그를 수정한다.
    - Body 태그 로딩 시에 팝업 함수 호출

```
<body style="background-color: black;" onload="pop_up();">
```

- 적용 이후 웹 브라우저 창 확인
  - 차단된 팝업을 허용한다.
  - 팝업 차단 해제([크롬 웹 브라우저 기준](#))
- 상대 경로의 이해
  - Js 폴더안에 popup.js 기준
    - popup.html의 위치는?
  - 한단계 상위 폴더에서 popup 폴더 안에 위치

상대 경로를 잘 이해하자.



# LOL 웹 사이트 - 팝업창

- 팝업 창 기능을 추가 구현한다.
  - 팝업창에 날짜를 출력한다.
- Popup.html 파일을 수정한다.
  - 시간 표시 부분 div 영역을 추가
    - Popup.js에 show\_clock() 함수도 구현한다.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>팝업창 테스트</title>
    <script type="text/javascript" src="../js/popup.js"
defer></script>
  </head>
  <body onload="show_clock();">
    <h1>팝업창 확인</h1>
    <div id="divClock" class="clock">
</body>
</html>
```

## 1초마다 갱신을 수행하는 날짜 함수

```
function show_clock(){
  let currentDate = new Date(); // 현재 시스템 날짜 객체 생성
  let divClock = document.getElementById('divClock');
  let msg = "현재 시간 : ";
  if(currentDate.getHours()>12){ // 12시 보다 크면 오후 아니면 오전
    msg += "오후";
    msg += currentDate.getHours()-12+"시";
  }
  else {
    msg += "오전";
    msg += currentDate.getHours()+"시";
  }

  msg += currentDate.getMinutes()+"분";
  msg += currentDate.getSeconds()+"초";
  divClock.innerText = msg;

  if (currentDate.getMinutes()>58) { //정각 1분전 빨강색 출력
    divClock.style.color="red";
  }
  setTimeout(show_clock, 1000); //1초마다 갱신
}
```

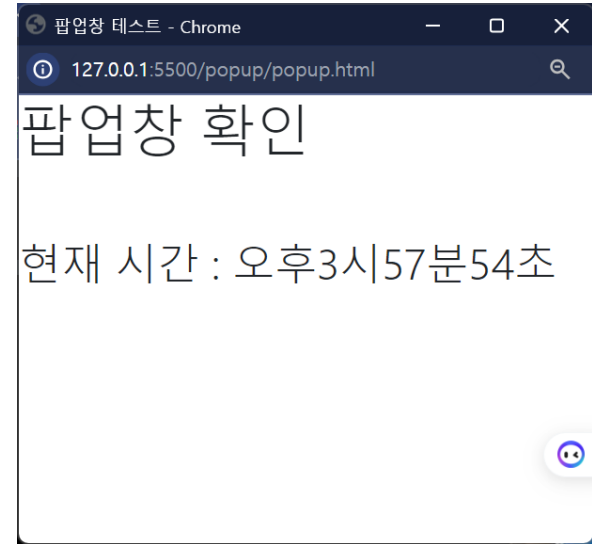


# LOL 웹 사이트 - 팝업창

- 팝업창의 날짜 및 시간을 확인한다.
  - 날짜 클래스는 기본 내장 클래스
    - 참고 : 캘린더(calendar) lib 따로 존재
- 부트스트랩 디자인을 적용한다.
  - popup.html의 코드를 추가 작성한다.
    - index.html의 코드를 재활용한다.
    - meta, link, script 태그를 추가한다.
  - 글자와 시간부분을 아래와 같이 수정한다.

```
<h1 class="display-1"><i class="bi bi-alarm"></i> 팝업창 확인<br><br></h1>  
<h1 class="display-4"><div id="divClock" class="clock"></div></h1>
```

- 적용 후 반응형 페이지로 동작



클래스	리턴 타입	메소드(매개 변수)	설명
LocalDate	int	getYear()	년
	Month	getMonth()	Month 열거값
	int	getMonthValue()	월
	int	getDayOfYear()	일년의 몇 번째 일
	int	getDayOfMonth()	월의 몇 번째 일
	boolean	isLeapYear()	윤년 여부
LocalTime	int	getHour()	시간
	int	getMinute()	분
	int	getSecond()	초
	int	getNano()	나노초 리턴

**자바에서는 함수 = 메소드라 한다.**

# LOL 웹 사이트 – 이미지 호버

- 로고 이미지의 호버 기능을 구현한다.
  - 로고에 마우스 커서를 올리면 이미지가 변화한다.
- popup.js에 호버 함수를 추가한다.
  - 현재 로고 이미지를 확인
    - 구글에서 적당한 크기 이미지 검색
    - 참고 : 사이즈 고정이라 큰 상관은 없다.
- index.html의 로고 이미지 속성을 수정한다.
  - 이미지 태그 속성에 마우스 이벤트를 추가한다.

```
onmouseover="over(this)" onmouseout="out(this)"
```

- 적용 후 마우스 이미지 변화 확인



LEAGUE<sup>OF</sup>  
LEGENDS

```
function over(obj) {  
    obj.src="image/LOGO.png";  
}
```

```
function out(obj) {  
    obj.src="image/LOGO_2.png";  
}
```





- 5주차 연습문제

자바스크립트 수정하기

# 자바스크립트 수정하기

- 검색 창에 문자열을 검사한다.
  - 검사할 항목 : 공백, 비속어 5개
- 현재 **search.js**를 다시 수정한다.
  - `googleSearch()` 함수를 수정
  - 요구 조건 : if문, 반복문, 배열 활용
    - 현재는 무조건 창을 연다.
  - 공백 검사
    - 문자 길이가 0이면 함수 중단
  - 비속어 검사
    - 배열 내에 포함되는 문자열이면 함수 중단



# Q & A

- 다음주 할 일
  - 자바 스크립트 이해
    - 자바 보안
  - LOL 웹 사이트 구현(JS)
    - 로그인/로그아웃
- 무엇이든 물어보세요!

