

jquery

제이쿼리란?

자바스크립트를 이용해 만든 라이브러리 언어

라이브러리란? 자바스크립트로 만들어진 다양한 함수들의 집합

-호환성문제해결 (자바스크립트가 표준을 갖추기전 브라우저에 종속적이었음)

-쉽고편리한 애니메이션효과기능 구현

제이쿼리 라이브러리 연동하기

- 다운로드방식

- 네트워크 전송방식(CDN)

<http://jquery.com/download/>

<https://cdnjs.com/libraries/jquery/>

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.0/jquery.min.js"></script>
```

■jquery사용해 보기

1. jquery **선택자**
2. jquery **속성변경** (css변경하기)
3. jquery **ajax** (비동기통신 , 서버에게 데이터 json으로 받아오기, 화면 전환 없이)

1. 선택자(jquery 선택자)

문서 객체를 불러오기전에 (문서의 로드가 완료되기 전에 자바스크립트로 dom 객체를 통해 요소들을 선택하면 안된다)
문서의 로드가 완료된 시점이후부터 DOM을 다루자 !

■자바스크립트

```
window.addEventListener("load", function(){  
    alert("문서의 모든 외부 리소스 포함 완료");  
} ) ;
```

```
document.addEventListener("DOMContentLoaded", function() {  
    // 이곳에 실행하고자 하는 코드를 작성합니다.  
    alert("문서로드 DOM 완료");  
});
```

■jquery : 아래 2개는 문서로드 완료와 같음

```
$(document).ready( function(){  
    alert( "문서로드 완료");  
});
```

```
$(function(){  
    alert(" 문서로드 완료 ");  
});
```

```
$(window).on('load', function() {  
    alert("문서로드 완료");  
});
```

\$("#선택자") :id
\$(".선택자") :class
\$("선택자") :태그

-CSS변경

```
$('#btn1').on("click",function(){  
    $("div").css("color","green");  
    $("div").css("translate","100px 100px");  
    $("div").css("transform","scale(2.0)");  
});
```

-내용변경

```
$('#btn1').on("click",function(){  
    $("div").html("<h2> div 내용 변경 </h2>");  
    $("#txt").val( "text내용변경 " );  
});
```

-내용가져오기

```
$('#btn1').on("click",function(){  
    let contents = $("div").html();  
    let contents2 = $("#txt").val() ;  
});
```

-추가하기

```
$('#btn1').on("click",function(){  
    $("div").append( "<div>추가 </div>" );  
});
```

-내용지우기

```
$('#btn1').on("click",function(){  
    $("div").empty( );  
});
```

■ Ajax

Ajax란?

동기: 서버로 요청시 응답이 와야 다음 동작을 수행 할 수 있음

비동기: 서버로 요청시 응답과 상관없이 다음 동작을 수행 할 수 있음

사용자의 컴퓨터에서 서버에 자료를 요청할 때 현재페이지의 화면전환 없이
서버에 요청한 데이터를 전송받을 수 있는 Ajax (Asynchronous javascript and xml) 매서드 이다.

`$.ajax();` // 호출시 인자값을 json형태로 넘겨줌 , 통신에 필요한 값들을 json형태 즉 객체형태로 전달함

```
$.ajax( {  
  url: " action url " ,  
  type: "get/post/put/delete" ,  
  data : "전송할 데이터" ,  
  dataType : "요청한 데이터형식, html, text, json,xml " ( 서버가 보내는 형식임)  
  success: function(){  
    //응답이 성공적일 때 수행할 콜백함수  
  } ,  
  error: function(){  
    //응답이 실패했을 때 수행할 콜백함수  
  }  
});
```



- 밥 제육덮밥 9000
- 면 칼국수 8000
- 분식 납작만두 6500

클릭시
아래 정보를 화면전환 없이
서버로부터 받을 수 있음
ajax매서드를 통해서
서버로부터 비동기로
데이터만 화면전환없이 받아올 수 있음
(json으로 받아옴)

■ JSON

자바스크립트로 객체를 표현하는 방법이다 (**객체**는 **속성** , **매서드**)의 집합이다

JSON은 "JavaScript Object Notation"의 약자로, 데이터를 저장하고 전송하기 위한 경량의 데이터 형식
JSON 데이터는 "키-값" 쌍의 집합으로 구성되며, 키와 값은 문자열, 숫자, 불리언, 배열, 객체, 또는 null 값일 수 있다.

예시)

```
let snack = { name : "맛동산" , price : 3700 };
```

■ 리터럴방식으로 객체생성하기

```
let score = { name:"hong" ,  
              kor:90 ,  
              eng:90 ,  
              tot:0 ,  
              calcTotal:function(){  
                  this.tot = this.kor+this.eng;  
              }  
            } ;
```

```
console.log( score.tot );
```

```
let test = {  
    key1: " value1",  
    key2: " value2",  
    key3: " value3",  
    success: function(){  
        alert( "성공" );  
    },  
    error: function(){  
        alert("실패" );  
    }  
}
```

■ 생성자함수를 이용한 객체만들기

```
var Person = function(name, age){  
    this.name = name; // this는 자신이 속한 객체를 참조하는 '자기 참조 변수'다.  
    this.age = age;  
    this.hello = function(){  
        return `이름은 ${this.name}이고, 나이는 ${this.age}입니다.`;  
    }  
}  
var hong = new Person('홍길동', 20);  
console.log(hong);  
let result = hong.hello();  
alert( result ) ;
```

JSON 라이브러리 , 기본으로 제공되는 라이브러리

```
JSON.parse( 객체문자열);    // 객체문자열 => 객체  
JSON.stringify( 객체);      // 객체 => 객체문자열 (json의 구조를 가진 문자열 )
```

```
// {"name":"맛동산","price": 3800}  
// "" 쌍따옴표 안에 문자열속에 쌍따옴표를 표현할 때 쌍따옴표앞에 \ 문자 추가한다. )
```

```
let snackStr = "{\name\\":\"맛동산\\\",\\price\\\":3800}";  
alert( snackStr ) ;
```

■ 자바스크립트 정리

자바스크립트란?

- ▶ 자바스크립트 선언문 :<script></script>
 - ▶ 자바스크립트 외부로 분리하기 :<script src="test.js" > </script>
 - ▶ 변수 : 하나 값을 저장할 수 있는 메모리 공간
 - let a ;
 - const b;
 - ▶ 자료형: 문자(열)형, 숫자, 논리형, 객체형, undefined (형이 정해지지 않았음을 의미하는 값)
자바스크립트는 변수선언시 값이 할당될 때 자료형이 결정된다.
- ```
let c;
alert(c); // undefined 출력됨
```

### ▶ 연산자

- 산술연산자 : +, -, \*, /, % (나머지)
- 문자열 연결연산자 : +
- 대입연산자 : =, +=, -=
- 증감연산자 : ++, --
- 비교연산자 : >, <, >=, <=, ==, !=
- 논리연산자 : &&, ||, !(논리부정)
- 삼항연산자 : 조건식?참코드:거짓코드 ;

### ▶ 제어문

순차, 분기, 반복

- 분기문:if, switch
- 반복문 : for, while, do~while
- break, continue

### ▶ 객체

- 내장객체 : String, Date, Array, Math (자바스크립트 제공)
- 문서객체 모델 : DOM (브라우저가 제공)
- 브라우저객체모델 : BOM

## ■ 문자열

```
let str ="hello";
let str2 = new String("hello");
```

## ■ 날짜

```
let today = new Date();
let nowMonth = today.getMonth(); // (현재월-1) 월정보 가져옴
let nowDate = today.getDate();
let nowDay = today.getDay(); // 일:0 ~ 토:6
```

```
document.write("오늘날짜 정보 출력");
```

```
let worldCup = new Date(2002,4,31) ;
let theMonth = worldCup.getMonth();
let theDate = worldCup.getDate();
let theDay = worldCup.getDay();
```

오늘부터 마지막 종료일까지 몇 일 남았을까요?  마지막까지 힘내보아요 !

```
<script>
let today = new Date();
let nowYear = today.getFullYear();
alert(nowYear);

let theDate = new Date(2023,11, 8) ; // 월-1 해야함 12월이라면 11로
let diffDate = theDate.getTime() - today.getTime();
 // getTime() 1970년 1월 1일 기준으로 현재날짜까지의 밀리초(천분의1초) 값 반환
let result = Math.ceil(diffDate/ (1000*60*60*24)) ; // /1000/60/60/24 => 일 구하기
alert(result +"남았어요 !");
</script>
```

## ■ 배열

```
let myFavorites = new Array(3);
 myFavorites[0] = "one";
 myFavorites[1]= "two";
 myFavoritesp2] ="three";

let myFavorites2 = ["one", "two", "three"] ;
```

배열을 이용한 로또만들기 예제

## ■브라우저 객체 모델

브라우저에 내장된 객체를 브라우저 객체라고 한다.  
window는 브라우저 객체의 최상위 객체이다.  
(BOM이라고 함)

▶ open()메서드 : 팝업창  
window.open( "url" , "새창이름" , "새창옵션");  
let url= "https://www.naver";  
let options= "width=350, height=400 , left=50, top=10 , scrollbars=no" ;

▶ close()메서드: 창 닫기  
window.close() ; 현재창 닫기

▶ prompt() ; 입력창      let userName= window.prompt();  
▶ confirm(); 확인창    let confirm = window.confirm("확인하시겠어요");  
▶ alert();      경고창

▶ setInterval()    ;    일정한 시간 간격으로 코드 실행하기  
▶ clearInterval() ; setInterval() 취소할 때 사용한다  
▶ setTimeout();    일정 시간이 지나면 코드를 실행하고 종료한다  
▶ clearTimeout(); setTimeout()메서드를 제거한다

### ▶ location객체

window.location.href="";      url주소변경  
window.location.reload() ;    브라우저의 새로고침 F5

### ▶ history 객체

window.history.back();    이전사이트  
window.history.foward() ;    다음 방문 사이트  
window.history.go(이동숫자) ;    이동숫자에 -2입력하면 2단계 이전 사이트로 이동

### ▶ navigator 객체

window.navigator.appCodeName; 현재브라우저의 코드명 : Mozilla  
window.navigator.appName;    현재브라우저의 이름 : Netscape  
window.navigator.language;    현재브라우저가 사용하고 있는 언어 반환 ko  
window.navigator.platform;    현재컴퓨터의 운영체제 정보  
window.navigator.product ;    현재브라우저의 엔진정보    Gecko  
window.navigator.userAgent;    브라우저와 운영체제에 대한 종합정보

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
</head>
<body>

<button id="find-me">Show my location</button>

<p id="status"></p>

<script>
function geoFindMe() {
 const status = document.querySelector("#status");
 const mapLink = document.querySelector("#map-link");

 mapLink.href = "";
 mapLink.textContent = "";

 function success(position) {
 const latitude = position.coords.latitude;
 const longitude = position.coords.longitude;

 status.textContent = "";
 mapLink.href = `https://www.openstreetmap.org/#map=18/${latitude}/${longitude}`;
 mapLink.textContent = `Latitude: ${latitude} °, Longitude: ${longitude} °`;
 }

 function error() {
 status.textContent = "Unable to retrieve your location";
 }

 if (!navigator.geolocation) {
 status.textContent = "Geolocation is not supported by your browser";
 } else {
 status.textContent = "Locating...";
 navigator.geolocation.getCurrentPosition(success, error);
 }
}

document.querySelector("#find-me").addEventListener("click", geoFindMe);

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