

개요

- ▶ jquery를 사용하기 위해 필요한 내용을 안다.
- jquery에서 selector의 개념과 활용방법을 안다.
- 기본 선택자(태그, id, class)에 대하여 안다.
- 속성 선택자를 활용할 수 있다.
- 계층 구조상 자손 및 후손 선택자를 활용할 수 있다.
- 필터 선택자의 개념을 활용할 수 있다.

- 브라우저에서 동작하는 자바스크립트 라이브러리!!
- 오픈소스 라이브러리
- 기능
 - 문서 객체 모델과 관련된 처리를 쉽게 구현
 - _ 일관된 이벤트 연결을 쉽게 구현
 - _ 시각적 효과를 쉽게 구현
 - AJAX 애플리케이션을 쉽게 개발
- 다운로드
 - http://jquery.com

jquery 시작:

- 라이브러리 사용
 - CDN(Content Delevery Network)
 - 직접 network상 라이브러리 호출..
 - http://code.jquery.com/jquery-1.10.2.js
 - Local에서 사용.
 - 라이브러리 다운
 - 로컬상 해당 위치지정 설정..
 - <script src="../a01_com/jquery-1.10.2.js">
 - <script src="/clientPrj/a01_com/jquery-1.10.2.js">

초기화면 loading:

- \$(document).ready(함수);
 - 문서(document)가 준비되면 함수를 통해서 해당 자바스크립트 코드 실행..
 - window.onload=함수와 동일한 처리를 하고 있다.
 - \$: jquery를 의미한다.
- \$("선택자").css("속성", "속성값");
 - *: 전체, #id명, .class명, tag명
 - ex) \$("*").css("color","red"); //전체 글자색상
- \$("선택자").attr("기본속성","기본속성값");
 - <input name="nam01" value="홍길동" >

jquery 접근 실습

- \$(document).ready(function(){
- ▶ // 전체 document객체의 글자 색상을 빨강으로 처리한다.
- \$("*").css("color","red");
- \$("*").css("background","yellow");
- \$("h2").css("color","blue"); // \$("tag로 접근")
- \$("#title03").css("color","navy");//\$("#id 로 접근")
- \$(".class04").css("color","purple"); // \$(".class를 접근")

jquery 접근 실습 :

- <body>
- <h1>안녕하세요!!! jquey는 빨강으로 시작!!??</h1>
- <h2 >안녕하세요!!! jquey는 파랑으로 tag접근!!??</h2>
- <h3 id="title03">안녕하세요!!! jquey는 id로 접근!!??</h3>
- <h4 class="class04">안녕하세요!!! jquey는 class으로 접근!!??</h4>
- <h5 class="class04">안녕하세요!!! jquey는 class으로 접근!!??</h5>
- <h6 class="class04">안녕하세요!!! jquey는 class으로 접근!!??</h6>





행복 마트

물건명	가격	현재고량	비고
사과	3,000	30	입고 중
바나나	4,000	300	재고 충분
딸기	12,000	50	

전공자는 data json 배열로 처리 객체.html("")

\$(document).ready(function(){

```
var list=[
{name: "사과", price: 3000, cnt: 30, etc: "입고 중"},
{name:"바나나",price:4000,cnt:300,etc:"재고
충분"},
{name:"딸기",price:12000,cnt:50,etc:""}
$("th").css("background","blue");
$("th").css("color","white");
$(".name").css("text-align","center");
$(".price").css("text-align","right");
$(".cnt").css("text-align","right");
```



```
var
names=document.querySelectorAll(".name");
var prices=document.querySelectorAll(".price");
var cnts=document.querySelectorAll(".cnt");
var etcs=document.querySelectorAll(".etc");
for(var idx=0;idx<list.length;idx++){</pre>
names[idx].innerHTML=list[idx].name;
prices[idx].innerHTML=list[idx].price;
cnts[idx].innerHTML=list[idx].cnt;
etcs[idx].innerHTML=list[idx].etc;
```

예제 풀이 🖁

```
<body><center>
<div >행복마트</div>
<br>
물건명가격현재고량<t
h>비고
<td
class="cnt">
<td
class="cnt">
<td
class="cnt">
```

속성 선택자

- 기본 선택자(input, select, h1) 뒤에 붙어 사용하는 속성(name, style, bgcolor)을 지정하는 것을 말한다.
 - \$('input[name=name01]').val("");
 - ps) val("@@"): value값으로 할당, 호출
 - 할당 \$(객체).val("himan");
 - 호출 var nameVal = \$(객체).val();
 - html("@@@"): innerHTML="" 할당과 호출
 - 할당 -\$(객체).html("안녕..");
 - 호출 var showList= \$(객체).html();



요소 선택 기본 예제 🕻

```
function chgSome(){
  // body 태그의 바로 밑에 있는 객체들..
  $("body > *").css("color","red");
<body>
  <h1 onclick="javascript:chgSome();">화면 변경</h1>
  <div style="color:blue;">
  ul>
  사과
  강아지
  가방
  코끼리
  </div>
```

여러 속성 선택자

- 요소[속성=값]
 - ex) \$("input[name=id01]").val("himan");
 - <input type="text" name="id01"/>
- 요소[속성!=값]: 해당 속성에 값이 같지 않는 것
 전체 호출..→ 배열을 지칭
 - ex) \$("input[type != password]").val("");
 - <input type="text" value="himan"/>
 - <input type="checkbox" value="ch01"/>
 - <input type="password" value="7777" />
 - 위의 객체 중에 type이 password가 아닌 input객체는 value을 ""으로 처리..

여러 속성 선택자

- 요소[속성=값]
 - ex) \$("input[name=id01]").val("himan");
 - <input type="text" name="id01"/>
- 요소[속성!=값]: 해당 속성에 값이 같지 않는 것
 전체 호출..→ 배열을 지칭
 - ex) \$("input[type != password]").val("");
 - <input type="text" value="himan"/>
 - <input type="checkbox" value="ch01"/>
 - <input type="password" value="7777" />



속성 선택예제 🖁

```
function chgAttr(){
   // $("요소[속성=속성값]").val("데이터 할당");
   $("input[name='id01']").val("비가 올 듯 말듯!! 기분이 좋을 듯
   말듯..");
   // $("요소[속성!=속성값]") : 해당 객체가 아닌 것에 value값을
   8888로 입력처리
   $("input[type!='password']").val("8888");
<input type="text" name="id01"/>
<input type="password"/>
<input type="text" name="id02"/>
<input type="radio"/>
<input type="checkbox"/>
<input type="file"/><br>
<input type="button" value="Shall we change?"</pre>
onclick="chgAttr()"/>
```

filter selector:

- 요소:button: 해당 속성에 type이 button 인것에 글자색상을 blue로 처리하세요.
 - \$("input:button").css("color","blue");
 - <input type="button"/>
 - 요소:checkbox, 요소:file, 요소:image
- 요소:checked : 주로 type이 radio, checkbox,
 - \$("input[name='gender']:checked").val();
 - <input type="radio" name="gender" value="남자"/>
 - <input type="radio" name="gender" value="여자"/>
- 요소:selected: select에서 활용되는 것으로 해당 요소가 선택되어진 것에 대한 처리를 할 때 활용된다.
 - \$("select > option:selected").val();
 - <select>
 - <option>서울</option>
 - <option selected>부산</option>
 - <option>대구</option>



filter기본예제 :

```
function filterFun(){
  alert("성별에 대한 처리 확인!!");
  // name값으로 gender이 있는 것중에..
  alert("성별은?"+$("input[name='gender']:checke
  d").val());
<body>
  <input type="button" value="filterData!!"</pre>
  onclick="filterFun()"/><br>
  남자<input type="radio" name="gender"
  value="gentlman"/>
  여자<input type="radio" name="gender"
  value="lady" checked/><br>
```

기타 filter요소들

- 요소:disabled → 비활성화된 입력양식 선택
 - \$("input:disabled")
- 요소:enabled → 활성화된 입력양식 선택
 - \$("input:enabled").val("입력준비완료!!");
- 요소:focus → 포커스 즉 클릭하는 순간 처리할 내용...
 - <input value="id입력하세요"/>
 - <input value="이름을 입력하세요!!"/>
 - \$("input:focus").val("");
 - // 마우스 클릭하는 순간에 value을 삭제



기본예제:

```
function chk(){
   $("input:focus").val("");
// 해당 하는 input에 focus를 위치할 때, 데이터를 삭제..
<body>
<h1>회원가입</h1>
이름:<input type="text" name="name01" value="입력" onclick="chk()"/><br>
생년월일:<input type="text" name="birth" onclick="chk()"/><br>ID:<input type="text" name="id01" onclick="chk()"/>
```

확인예제:



- 1단계
- [@@] + [@@] = [입력하세요]
- 입력하세요에 클릭하는 순간 입력이 가능하게 문자 삭제 처리. 해당 정답이 맞을 때!! 정답, 틀렸을 때 오답..
- 2단계
 - Math.random()을 활용하여 ##(사칙연산임의)
- [@@] ## [@@] = [입력하세요..]
- 위 내용 selector를 이용 선택 처리하게
- 3단계(숙제)
 - 문제 10문제를 시간 count down처리 해서
 - 시간이 지나면 맞은 문제 갯수을 10점 환산에 점수

- 위치에 관련된 선택자를 말하면, 크게 3가지 정도가 있다.
 - 요소:odd → 홀수 번째 위치한 문서 객체 선택
 - \$("tr:odd").css("background","blue");
 - 요소:even → 짝수 번째 위치한 문서 객체 선택
 - \$("tr:even").css("background", "green");
 - 요소:first → 첫번째 위치한 문서 객체를 선택
 - \$("tr:first").css("color","red");
 - 요소:last → 마지막 위치한 문서 객체를 선택
 - \$("tr:last").css("color","pink");
- 주로 list형식: tr, td, h1, h2.., li...

함수 필터 선택자 🖁

- \$("요소:eq(n)"): n번째 위치하는 문서 객체
 - ex) \$("h1:eq(2)").attr("align","center");
 - <h1>의 중 두번째 위치한 것의 정렬을 중앙
- 요소:contains('문자열'): 특정 문자열을 포함한 내용의 객체를 선택..
 - ex) \$("td.ename:contains('홍길동')").css("color","r ed");
 - td중에 class명이 ename에서 '홍길동'이라는 문자열 있는 데이터는 글자색을 빨강색으로 한다.
- 요소:nth-child(Kn+1):K 단위로 해당 요소의 내용을 반복으로 처리하고 할 때, 활용한다.

기본 예제 🖁

주간 단위로 날짜에 대한 내용을 색상을 변경하는 부분을 처리하자.. 클릭했을 시, 일정도 기입되게 하는 내용..

일	월	화	수	목	己	토
2	3	4	5	6	7	8

- 특정한 요일에 배경 색상 표시..객체.eq()
- 3개 단위로 당직일정을 처리하고자 한다.
 - nth-child(3) 홍사원, 김대리, 오과장

소스 처리

- 화면구성
 - ...
- 오늘 날짜를 Date로나 임의의로 날짜 가져와서 해당 날짜에 배경색상을 노랑색으로 처리 \$("td:eq("+num+")").css("background"," yellow"); // \$("td").eq(num).css(....
- 날짜별 당직 인원 처리 표시(3사람)
 - var persons=["오사원","김대리","오과장"];
 - \$("td:nth-child(3n)").html(...+persons[0]);

배열객체관리.

- for, for(in)
- jquery에는 배열을 손쉽게 관리하기 위하여 .each() 메서드를 사용합니다.
 - \$.each(배열객체, function(index, item){
 - index활용, 배열안에 있는 단위 item로 활용
 - });
 - \$(배열객체).each(function(index, item){
 - }); ex) \$("#personList").each()
 - item→ {name:"김길동", age:31} personList[index]

```
{name:"홍길동",{name:"김길동",{name:"신길동",{name:"마길동",age:27}[0]age:31}[1]age:35}[2]age:21}[3]
```





- json데이터로
 - weekplans : 날짜, 시간, 계획내용 → 3개..
 - {date:7,time:"19:00",plan:"불타는금요일시작 "}
 - 아래 형식으로 화면에 나타내세요..
 - This weekend is beautiful!!!

날짜	시간	계획내용
7	19:00	불타는금요일시작



위치 필터 선택자 :



