*** overview ***

1. jQuery
가. 빠르고 간결한 자바 스크립트 라이브러리
나. 빠른 웹개발 단순화 지원
- HTML문서 트래버싱
- 이벤트 처리
- 비동기 상호작용
나. 코드 작성을 줄여 다양한 작업을 단순화 하기위한 자바스크립트 툴킷

2. jQuery가 지원하는 중요상 핵심 특징
가. 선택자 엔진을 사용한 편리한 DOM 조작: DOM 요소 선택/읽기/변경
나. 이벤트 처리
- 폭넓은 이벤트 캡처
- 복잡한 이벤트 처리 코드 불필요
다. 비동기 지원: 비동기 기술을 사용한 풍부한 사이트 개발 지원
라. 애니메이션: 풍부한 빌트인 애니메이션 효과
마. 작고 가벼운 라이브러리
바. 크로스 브라우저 지원
사. CSS3 선택자 지원

3. 사용방법
가. 로컬 설치
<script type="text/javascript"
src="/jquery_study/jquery/jquery-버전.min.js">
</script>
나. CDN 기반 버전
<script type="text/javascript"
src="http://code.jquery.com/jquery-3.5.1.min.js">

4. jQuery 라이브러리 함수 호출: 웹페이지와 작업할 경우
- 준비 이벤트 등록: $(document).ready() function
- 준비 이벤트 함수 내부: DOM 로딩되자 마자 실행(웹페이지 컨텐츠가 로딩 전: BODY)

5. 커스텀 자바 스크립트: 스크립트 파일을 외부에

6. 여러 자바 스크립트 라이브러리와 사용할 때 충돌되면: $->jQuery로 변경

7. 목차

05. DOM 트래버싱

06. CSS 선택자

07. DOM 조작

08. 이벤트 처리

09. AJAX

10. forminput

*** basics ***

1. 이유

- jQuery는 자바스크립트 기능을 사용해 개발된 프레임워크

2. 다시 기억 하기

가. 문자열(String): immutable(변경할 수 없는) (01)

- "This is JavaScript String"
- 'This is JavaScript String'
- 'This is "really" a JavaScript String'
- "This is 'really' a JavaScript String"

나. 숫자(Numbers): 배정도 64비트 포맷 값, immutable(변경할 수 없는)

- 5350
- 120.27
- 0.26

다. 논리(Boolean)

- true // true
- false // false
- 0 // false
- 1 // true
- "" // false: 빈 문자열(empty string)
- "hello" // true

라. 객체(Objects)

- 객체 생성

```
var emp = {
        name: "Zara",
        age: 10
};
```

- 객체 읽고 쓰기

```
// 속성 읽기
emp.name
emp.age
// 속성 쓰기
```

```
emp.name = "kht"
emp.age = 20
```

마. 배열(Arrays)

- 배열 정의

```
var x = [];
var y = [1, 2, 3, 4, 5];
```

- 배열의 길이

```
var x = [1, 2, 3, 4, 5];
for (var i = 0; i < x.length; i++) {
        x[i]로 작업
}
```

바. 함수(Functions)

- 이름있는 함수: function 함수명() { }
- 이름없는(익명)함수: var handler = function() { }
- jQuery에서의 익명함수 사용

```
$(document).ready(function(){
        작업코드
});
```

사. 아규먼트(Arguments)

- 함수 구현

```
function func(x){
        alert(typeof x, arguments.length);
}
```

- 함수 호출

```
func();                                 //  "undefined", 0
func(1);                                //  "number", 1
```

아. 영역: 전역/지역변수

자. 콜백

```
$("body").click(function(event) {
        alert("clicked: " + event.target);
});
```

차. 클라우저 (02)

- 현재 영역 밖에 정의된 변수를 내부 영역내에서 접근될 때 생성
- 변수 counter가 create, increment, print 함수에서 어떻게 보여지는지
- 외부에서는 보이지 않는 데이터를 조작하는 메소드를 가진 객체를 생성

카. 내장함수

*** selectors ***

1. jQuery 선택자

- CSS의 선택자 기능을 활용: DOM(HTML문서)내에 존재하는 요소들에 쉽게 접근

- DOM안에 존재하는 요소들과 일치하는 것을 찾아내기 위해 표현식을 사용하는 함수($()) ->jQuery

2. $() Factory 함수 (01)

가. jQuery가 가지고 있는 모든 타입의 선택자는 항상 $()로 시작

나. 주어진 문서(웹페이지)에 존재하는 요소를 찾는 동안 세 개의 주요한 블럭을 사용

- 태그이름(Tag Name)

: DOM에 있는 이름을 가진 태그를 표현

예) $('p'): 문서내 모든 패러그래프 <p>를 선택

- 태그아이디(Tag ID): DOM에 있는 아이디를 가진 태그를 표현

예) $('#some-id'): 문서내 단일 요소를 선택

- 태그클래스(Tag Class): DOM내 클래스를 가진 태그를 표현

예)$(.클래스명): 문서내에서 클래스를 가진 모든 요소를 선택

3. 기타

- 선택자는 독립적 또는 조합

- 팩토리 함수 $()는 jQuery()함수와 동의어

- 다른 자바 스크립트와 동시에 사용할 때 $ 충돌이 발생하면 $()->jQuery()로 변경

4. 기본 선택자 사용법

4.1 Name (02)

가. 정의: 요소이름과 일치하는 모든 엘리먼트 선택

나. 문법:$('tagname')

다. 파라메터: tagname: 표준 HTML 태그 이름, div, p, img, li 등

라. 리턴값: 발견된 요소들의 배열

4.2 #ID (03)

가. 정의: 아이디 속성을 가진 단일 엘리먼트 선택

나. 문법:$('#elementid')

다. 파라메터: elementid: 특수문자(: .)는 백슬래시(₩) 사용

라. 리턴값: 발견된 요소들의 배열

4.3 Class (04)

가. 정의: 클래스와 일치하는 모든 엘리먼트 선택

나. 문법:$('.classid')

다. 파라메터: classid

라. 리턴값: 발견된 요소들의 배열

마. 예)

$('.big')

$('p.small'): small 클래스 아이디를 가진 모든 패러그래프 선택

$('.big.small'): big과 small 클래스 아이디를 가진 모든 요소 선택

4.4 Universal(*) (05)

가. 정의: 문서내 모든 엘리먼트 선택

나. 문법: $('*')

다. 파라메터

라. 리턴값: 발견된 요소들의 배열

4.5 멀티플 요소 E, F, G (06)

가. 정의: 모든 지정된 선택된 E, F 또는 G 결과를 조합해서 선택

나. 문법: $('E, F, G, ...')

다. 파라메터: E(임의의선택자), F(임의의선택자), G(임의의선택자)

라. 리턴값: 발견된 요소들의 배열

마. 예

- $('div, p'): div 또는 p와 일치하는 모든 요소
- $('p strong, .myclass'): p 하위에 있는 strong 속성과 일치 또는 myclass
- $('p strong, #myid')

5. 기타 선택자 예

- $('*')
- $('p > *'): 패러그래프 엘리먼트의 모든 자식요소들
- $('#elementid')
- $('.classid')
- $('li:not(.myclass)'): 클래스아이디가 myclass가 아닌 모든 li 요소들
- $('a#elementid.classid'): 아이디 그리고 클래스 가진 link 요소
- $('p a.classid'): p안에 클래스 아이디를 가진 link 요소
- $('ul li:first'): ul안에 첫번째 li 요소
- $('#container p'): 아이디 안에 p 요소
- $('li > ul'): li 안에 모든 ul 요소들
- $('code, em, strong'): 또는 요소들
- $('p strong, .myclass')
- $(':empty'): 자식이 없는 모든 요소들
- $('p:empty'): 자식이 없는 모든 p 요소들
- $('div[p]'): p와 일치하는 요소를 포함하는 div와 매치되는 모든 요소들
- $('p[.myclass]')
- $('a[@rel]'): rel 속성을 가진 a와 일치하는 모든 요소
- $('input[@name=myname]'): 일치
- $('input[@name^=myname]'): 시작
- $('a[@rel$=self]'): self로 끝나는 rel 속성값을 가지는 a와 일치하는 모든 요소들
- $('a[@href*=domain.com]'): 포함
- $('li:even'): even 인덱스를 가진
- $('tr:odd'): odd 인덱스를 가진
- $('li:first'): 첫번째 li 요소
- $('li:last'): 마지막 li 요소
- $('li:visible'): 보이는 li 요소
- $('li:hidden'): 숨겨진 li 요소
- $(':radio'): 폼내 모든 레디오 버튼

- $(':checked'): 폼내 체크된 모든 박스
- $(':input'): 오로지 폼 요소만(input, select, textarea, button)
- $(':text'): text 요소만(input[type=text])
- $('li:eq(2)'): 세번째 li 요소
- $('li:lt(2)')
- $('p:lt(3)')
- $('li:gt(1)')
- $('p:gt(2)')
- $('div/p'): div의 자식 p
- $('div//code'): div후손 code 매치
- $('//p//a')
- $('li:first-child')
- $('li:last-child')
- $('li:contains(second)'): 텍스트 second를 포함하는 li

*** attributes ***

1. 정의 : DOM의 요소에 할당되어 요소의 특성을 조작할 수 있는 컴포넌트

2. 자바 스크립트에서 활용가능한 속성 구조: 이름=값
가. 예
- <img id="mylmage" src="image.gif" alt="이미지"
class="클래스명" TITLE="이것은 이미지이다." />
- 태그이름: img
- 속성: id="mylmage", src="image.gif", alt="이미지"
class="클래스명", title="이것은 이미지이다."

3. 속성값 가져오기(01)
가. 대상 샘플
- 첫번째 문단이다.
-
나. 속성값 가져오기
- 변수명1 = $('em').attr('title')
- $('#divid').text(변수명1)

4. 속성값 설정하기(02)
가. 대상 샘플
-
나. 속성값 지정하기
- $('#myimg').attr('src', '/jquery_study/images/Chrysanthemum.jpg')

5. 스타일 적용(03)

가. 대상 샘플

```
<style>
.selected { color:red; }
.highlight { background:yellow; }
</style>
<body>
<em title="Bold and Brave">This is first paragraph.</em>
<p id="myid">This is second paragraph.</p>
</body>
```

나. 스타일 적용

```
$('em').addClass('selected');
$("#myid").addClass("highlight");
```

6. 속성함수

가. attr(속성): 속성들 쓰기(04)

- 문법: 선택자.attr({속성명1:속성값1, 속성명2:속성값2});
- 파라메터: 키(속성명), 값(속성값)

나. attr(키, 함수): 하나의 아규먼트(현재 인덱스)를 가진 함수로 리턴값으로 설정(05)

- 문법: 선택자.attr(키, 함수)
- 파라메터: 키, 함수명

다. removeAttr(속성명): 일치하는 요소의 속성명 삭제(06)

- 문법: 선택자.removeAttr(속성명);
- 파라메터: 속성명

라. hasClass(class): 클래스 존재여부를 논리값 리턴(07)

- 문법: 선택자.hasClass(클래스명)
- 파라메터: CSS 클래스명

마. removeClass(class): 클래스 삭제(08)

- 문법: 선택자.removeClass(클래스명)
- 파라메터: CSS 클래스명

바. toggleClass(class): 있으면 제거하고, 없으면 추가(09)

- 문법: 선택자.toggleClass(클래스명)
- 파라메터: CSS 클래스명

사. html(): 일치하는 첫번째 요소 html 내용 얻기(10)

- 문법: 선택자.html()
- 파라메터: 없음

아. html(val): 일치하는 모든 요소 html 내용 쓰기(11)

- 문법: 선택자.html(val)
- 파라메터: 임의의 문자열

자. text(): 일치하는 모든 요소 내용 읽기(12)

- 문법: 선택자.text()
- 파라메터: 없음
- 예1: text로 읽어와서 html로 쓰기 - (12-1)
- 예2: html로 읽어와서 text로 쓰기 - (12-2)

- 예3: html로 읽어와서 html로 쓰기 - (12-3)
- 예4: text로 읽어와서 text로 쓰기 - (12-4)

차. text(val): 임의의 문자열을 쓰기(13)

카. val(): 첫번째 매치된 input 값 읽기

- 문법: 선택자.val() (14)
- 파라메터: 없음

타. val(val): 모든 매치된 input 값 쓰기 (15)

- 문법: 선택자.val(val)
- 파라메터: 임의의 문자열

*** domtraversing ***

1. 인덱스로 요소들 찾기 (01)

- 모든 리스트는 자신의 인덱스를 가짐.
- 시작은 0부터
- eq(index) 함수 활용방법: 선택자.eq(index)

2. 요소 필터링 (02)

- 매치된 요소를 필터링
- 사용방법: 선택자.filter(선택자)

3. 하위요소 찾기: 선택자.find(선택자) (03)

4. jQuery DOM 필터 함수들

가. eq(인덱스): 일치된 요소를 단일요소로 줄여 준다. (04)

- 문법: 선택자.eq(인덱스)
- 파라메터: 인덱스(일치된 요소들의 위치: 0부터 시작)

나. filter(선택자): 일치하지 않는 모든 요소 삭제 (05)

- 문법: 선택자.filter(선택자)
- 파라메터: 선택자(".class1, .class2") -> 하나 이상 가능

다. filter(fn): 일치하지 않는 모든 요소 삭제 (06)

- 문법: 선택자.filter(fn)

라. is(선택자): 하나 이상의 일치하는지 확인 (07)

- 문법: 요소.is(선택자)

```
if($('li').is(':first-child')) {

} else if($('li').is(':last-child')) {

} else if($('li').is('.middle0, middle1')) {
하단 패러그래프에 출력
}
```

- 파라메터
- 리턴: true/false

마. map(콜백): jQuery 객체내의 요소집합을 jQuery 배열로 변환 (08)
- 문법: 선택자.map(콜백)
- 파라메터: 콜백 -> 각각의 요소에 대해 실행하는 함수
바. not(선택자): 필터 (09)
- 문법: 선택자.not(선택자)
- 파라메터: not(".class1, class2")
사. slice(시작, [끝]): 일치된 요소의 서브 집합을 선택 (10)
- 문법: 선택자.slice(시작, 끝)
- 파라메터
아. add(선택자) (11)
- 문법: 선택자.add(선택자)

*** cssselector ***

1. 개요
- jQuery 라이브러리는 CSS1부터 3까지에 포함된 거의 모든 선택자를 지원
- DOM 요소에 CSS 속성을 적용하는데 사용

2. CSS 속성 적용
- 단일: 선택자.css(속성명, 속성값)
- 다중: 선택자.css({속성명1:속성값1, 속성명2:속성값2...}) (01)

3. 요소 폭과 넓이 설정: width(val), height(val) (02)

4. jQuery CSS 함수들
가. css(이름): 첫번째 일치된 요소(element) 스타일 속성 리턴 (03)
- 문법: 선택자.css(이름)
- 파라메터: 이름(접근하는 속성 이름)
나. css(이름, 값): 모든 일치된 요소에 대해 속성값이 설정 (04)
- 문법: 선택자.css(이름, 값)
- 파라메터: 이름(접근하는 속성 이름), 값(속성값)
다. css(속성들): 속성들 설정 (05)
- 문법: 선택자.css(속성들) -> 선택자.css({키1:값1, 키2:값2...})
- 파라메터
라. height(): 첫번째 일치된 요소의 현재 높이 속성값 읽기 (06)
- 문법: 선택자.height()
- 파라메터: 없음
- 기타: $(window).height(), $(document).height()
마. position(): 요소의 left, top값 읽기 (07)
- 문법: 선택자.position()
- 파라메터: 없음

*** dommanupulation ***

1. 개요

가. jQuery는 효율적인 방법으로 DOM을 조작하는 방법을 제공

나. 편의성 제공: 많은 코드 작성 불필요

- 임의 요소의 속성값을 변경
- 패러그래프나 디비전으로부터 HTML 코드 추출

다. DOM 요소 정보 가져오는 getters: .attr(), .html(), and .val()

2. 내용 조작

- 선택자.html(): 일치되는 첫번째 html 내용 (01)

3. 돔 요소 대체

- 문법: 선택자.replaceWith(내용) (02)

4. 돔 요소 제거

가. 선택자.remove(): 모든 일치된 요소를 삭제 (03)

나. 선택자.empty(): 일치된 요소의 모든 자식노드를 삭제 (04)

5. DOM 요소 추가: 존재하는 DOM 요소에 1개이상의 새로운 DOM 요소 추가

가. 선택자.before(내용) (05)

나. 선택자.after(내용) (06)

6. 기타 함수들

가. append(내용): 모든 일치된 요소들안에 내용을 추가 (07)

- 문법: 선택자.append(내용)

나. appendTo(선택자): 모든 일치된 요소들안에 선택된 요소에 추가 (08)

- 문법: 선택자.append(선택자)

다. clone(bool): 일치된 선택된 요소를 해당 요소 다음에 추가 (09)

- 문법: 선택자.clone(true) 또는 선택자.clone()

라. html(값): 모든 일치된 요소의 html 내용을 설정 (10)

- 문법: 선택자.html(값)

마. html(): 일치된 첫번째 요소의 html 내용을 읽기 (11)

- 문법: 선택자.html()

*** eventshandling ***

1. 이벤트

가. 다이내믹 웹페이지를 만드는 것이 가능

나. 웹 어플리케이션에 의해 감지될 수 있는 액션들

다. 예

- 마우스 클릭
- 웹페이지 로딩
- 요소(엘리먼트)상의 마우스 오버
- HTML 폼 보내기
- 키보드에 의한 키 입력

라. 절차

- 이벤트 발생(트리거)
- 사용자 함수를 사용해 원하는 이벤트를 처리
- 사용자정의 함수: 이벤트 핸들러

마. 이벤트 핸들러 바인딩

- jQuery모델을 사용해 bind()함수로 DOM 요소에 대한 이벤트 핸들러 설정 (01)
- 문법: 선택자.bind(이벤트 타입[, 이벤트 데이터], 이벤트 핸들러)
- 파라메터

이벤트 타입: 자바스크립트 이벤트 타입을 포함하는 문자열

이벤트 데이터: 이벤트 핸들러에 넘겨지는 데이터(옵션)

이벤트 핸들러: 이벤트가 발생(트리거)될 때마다 실행하는 함수

바. 이벤트 핸들러 제거

- 이벤트 핸들러 주기: 웹페이지의 라이브 사이클과 동일(페이지 종료) (02)
- 문법: 선택자.unbind(이벤트 타입)

2. 이벤트 타입

- blur: 요소가 포커스를 잃을 때
- change: 요소값이 변할 때
- click: 마우스 클릭
- dblclick: 마우스 더블 클릭
- mouseenter: 마우스가 요소 영역으로 들어갔을 때: bind("mouseenter", function()
- mouseleave: 마우스가 요소영역에서 벗어날 때: bind("mouseleave", function()
- Mouseover: 마우스 포인터가 요소위를 이동할 때
- Select: 텍스트가 선택될 때
- Submit: 폼이 전송될 때

...

3. 이벤트 객체란?

- 이벤트 핸들러: 콜백함수
- 콜백함수: 하나의 파라메터를 가짐

- 그 하나의 파라메터가 이벤트 객체
- 불필요해서 파라메터가 생략되는 경우가 많음

4. 이벤트 객체 속성 (03)
- type: 발생된(트리거된) 이벤트 타입
- pageX: 마우스 이벤트에서 웹페이지 원점에서의 X좌표
- pageY: 마우스 이벤트에서 웹페이지 원점에서의 Y좌표
- target: 이벤트가 발생된 요소

5. 이벤트 함수들
가. preventDefault(): 브라우저의 기본 액션을 중지 (04)
- 문법: 이벤트객체.preventDefault()
나. isDefaultPrevented(): 브라우저 기본액션 중지여부 확인 (05)
- 문법: 이벤트객체.isDefaultPrevented()
다. stopPropagation(): 전파 금지 (06) (07)
- 문법: 이벤트객체.stopPropagation()
라. isPropagationStopped(): 전파 금지 여부 확인
- 문법: 이벤트객체.isPropagationStopped()

6. 기타 이벤트 조작 함수들
가. hover(): 마우스 온, 오프 (08)
나. trigger(): 각가 일치되는 요소에 대해 이벤트를 발생(트리거) (09)

*** ajax ***

1. 개요
- AJAX: 비동기(Asynchronous) 자바스크립트(JavaScript) and XML
- 이점: 웹브라우저 웹페이지를 리프레시 하지 않고 서버로 부터 온 데이터를 로딩하는 기술
- JQuery: 차세대 웹 어플리케이션을 개발용 AJAX 함수집합을 제공 툴

2. 데이터 로딩: jQuery AJAX를 사용한 정적/동적 데이터 로딩 (01)
가. 문법: [선택자].load(URL, [데이터], [콜백]);
나. 파라메터
- URL: 서버측 자원 경로
- 데이터: 부가 옵션 파라메터로 요청시점에 넘겨지는 속성을 가진 객체를 표현
- 콜백
응답데이터가 로딩된 후에 호출되는 콜백함수
함수 첫번째 파라메터: 웹서버로부터 수신하는 응답텍스트
함수 두번째 파라메터: 상태코드

2. JSON 데이터 얻기 (02)
가. 개요

- 웹서버가 요청에 대해 JSON 문자열을 되돌려 주는 상황
- jQuery 유틸리티 함수 중 getJSON()사용:
리턴된 JSON 문자열을 파싱해서 콜백함수에게 결과스트링을 사용할 수 있게 함.
나. 문법: 문법: [선택자].getJSON(URL, [데이터], [콜백]);
다. 파라메터
- URL: GET 방식의 서버측 자원 경로
- 데이터: URL에 추가되는 질의문자열로 사용되는 이름/값쌍 사용되는 객체
- 콜백

3. 웹서버에 데이터 넘기기 (03)
가. 개요
- 사용자로부터 input으로 데이터를 받아 서버로 전송

4. jQuery AJAX 함수들
가. jQuery.ajax(옵션): HTTP 요청으로 원격 페이지 로드 (04)
- 옵션
- complete: 요청이 끝날 때마다 실행되는 콜백 함수
- data: 요청과 함께 서버로 보내지는 문자열
- error: 요청이 실패하면 실행되는 콜백 함수
- success: 요청이 성공하면 실행되는 콜백함수
- Timeout: 요청이 1/1000초가 지나면 실패
- type: GET/POST, 기본값이 GET
- url: 요청이 보내지는 경로
나. jQuery.get(...) (05)
- 문법: jQuery.get(url, [데이터], [콜백], [요청타입])
- 파라메터
url: 요청이 보내지는 경로
데이터: 서버로 보내지는 키/값쌍
콜백: 데이터 로드때마다 실행되는 함수
타입: 리턴 테이터 타입 -> xml, html, script, json, text
다. jQuery.post(...) (06)
- 문법: jQuery.get(url, [데이터], [콜백], [요청타입])
- 파라메터
url: 요청이 보내지는 경로
데이터: 서버로 보내지는 키/값쌍
콜백: 데이터 로드때마다 실행되는 함수
타입: 리턴 테이터 타입 -> xml, html, script, json, text
라. serialize(): input 요소에 있는 값을 문자열 배열로 변환
- 문법: $.serialize() (07)
- 파라메터: 없음

5. jQuery AJAX 이벤트
가. 이벤트를 통한 페이지로드와 ajax종료 (08)

나. 이벤트를 통한 페이지로드와 ajax시작과 종료 (09)

다. 페이지 로딩 실패(10)

라. 이벤트를 통한 페이지로드와 ajax 시작, 보내기, 종료 (11)