

The screenshot shows the Windows Task Manager Performance tab. The CPU section at the top indicates '100%' usage. Below this, a list of processes is shown, with 'System Idle Process' at the top and 'smss.exe' at the bottom. The 'smss.exe' process is highlighted in blue, indicating it is the active process. The 'smss.exe' process is a 64-bit application running under the SYSTEM user, with a PID of 4.

## 학습목표

- jQuery에 관해 이해하고, 이를 설명할 수 있다.
- jQuery를 활용할 수 있다.

## 학습내용

- jQuery 개요
- jQuery 응용
- jQuery 실습

## jQuery 개요

### 1. jQuery란?

#### 1) 개요



jQuery란?



간편하게 사용할 수 있도록 단순화

### 오픈 소스 기반의 자바스크립트 라이브러리

- HTML의 클라이언트 사이드 조작을 단순화하도록 설계됨
- jQuery 라이브러리는 jQuery라는 이름의 객체로 생성됨

## jQuery 개요


### 1. jQuery란?

#### 2) 특징



#### CSS 선택자

- HTML 문서의 구조를 명료하고, 읽기 쉬운 형태로 표현하고 사용 가능



#### 플러그인 아키텍처

- 중복되는 기능과 코드가 엉키는 것을 피함
- 이미 개발된 많은 플러그인을 쉽고 빠르게 이용 가능

## jQuery 개요

### 1. jQuery란?

#### 2) 특징

#### 메서드 체인

- 여러 기능을 한 줄에 나열하고,  
임시 변수의 사용을 최소화  
➡ 불필요한 코드의 반복을 피함

#### 크로스 브라우저

- 브라우저마다 다르게 발생하는 이벤트,  
객체 등을 처리하고자 여러 개의 메서드를  
사용하지 않음  
➡ jQuery에서 제공되는 메서드 또는  
문장으로 간단하게 해결 가능

## jQuery 개요

### 1. jQuery란?

#### 3) 적용 방법

- jQuery 파일은 자바스크립트 파일(.js) 형태로 존재
- 웹페이지에서 사용하기 위해서는 파일을 먼저 웹페이지에 로드해야 함

방법이 무엇인가요?



## jQuery 개요

### 1. jQuery란?

#### 3) 적용 방법

## [ jQuery 파일을 다운받아 로드 ]

```
<head>  
  <script src = “/파일경로/제이쿼리파일명.js”></script>  
</head>
```

## [ CDN(Content Delivery Network)을 이용하여 로드 ]

#### ■ jQuery.com CDN

```
<script src = “https://code.jquery.com/jquery-1.12.4.min.js”></script>
```

#### ■ 구글 CDN

```
<script src = “https://ajax.googleapis.com/ajax/libs/jquery/1.12.4/jquery.min.js”></script>
```

#### ■ MS CDN

```
<script src = “http://ajax.aspnetcdn.com/ajax/jQuery/jquery-1.12.4.min.js”></script>
```

#### ■ CDNJS CDN

```
<script src = “https://cdnjs.cloudflare.com/ajax/libs/jquery/1.12.4/jquery.min.js”></script>
```

#### ■ jsDelivr CDN

```
<script src = “https://cdn.jsdelivr.net/jquery/1.12.4/jquery.min.js”></script>
```

## jQuery 개요

### 2. jQuery 문법

#### 1) 기본 문법

- HTML 요소를 선택하고 선택한 요소에 손쉽게 특정 동작을 설정할 수 있음
- **\$ 기호는 jQuery를 의미**하고 이를 접근할 수 있게 하는 식별자임
- 동작 함수를 정의하여 선택된 요소에 원하는 동작을 설정
- **문법**

```
$ (선택자).동작함수();
```

#### 2) Document 객체의 ready()

- 자바스크립트는 웹브라우저의 모든 요소를 로드한 뒤 실행되어야 함
- **오류가 발생할 경우**

아직 생성되지 않은 HTML  
요소에 속성을 추가할 때

아직 로드되지 않은 이미지의  
크기를 얻으려고 할 때



## jQuery 개요

### 2. jQuery 문법

#### 2) Document 객체의 ready()

[ 요소 로딩여부에 따라 자바스크립트를 실행할 수 있음 ]

- 자바스크립트 Window 객체의 onload() 메소드

```
window.onload = function() {  
    자바스크립트 코드;  
};
```

- jQuery에서 Document 객체의 ready() 메소드

```
$(document).ready(function() {  
    제이쿼리 코드;  
});
```

## jQuery 응용

### 1. 태그 요소 조작

#### 1) 선택자

- CSS 선택자를 사용하여 HTML 요소를 선택함
- `$()` 함수에 전달되는 인수는 따옴표(“ ”)를 사용한 문자열 형태로 전달되어야 함

#### [ 기타 선택자 예 ] 1/4

- Selects all elements

```
$("#*")
```

- Selects the current HTML element

```
$(this)
```

- Selects all `<p>` elements with `class = "intro"`

```
$("#p.intro")
```

## jQuery 응용

### 1. 태그 요소 조작

#### 1) 선택자

#### [ 기타 선택자 예 ] 2/4

- Selects the first `<p>` element

```
$(“p:first”)
```

- Selects the first `<li>` element of the first `<ul>`

```
$(“ul li:first”)
```

- Selects the first `<li>` element of every `<ul>`

```
$(“ul li:first-child”)
```

## jQuery 응용

### 1. 태그 요소 조작

#### 1) 선택자

#### [ 기타 선택자 예 ] 3/4

- Selects all elements with an href attribute

```
$("#[href]")
```

- Selects all <a> elements with a target attribute value equal to “\_blank”

```
$("#a[target = ‘_blank’]”)
```

- Selects all <a> elements with a target attribute value NOT equal to “\_blank”

```
$("#a[target! = ‘_blank’]”)
```

## jQuery 응용

### 1. 태그 요소 조작

#### 1) 선택자

#### [ 기타 선택자 예 ] 4/4

- Selects all `<button>` elements and `<input>` elements of type = "button"

```
$(".:button")
```

- Selects all even `<tr>` elements

```
$(".tr:even")
```

- Selects all odd `<tr>` elements

```
$(".tr:odd")
```



jQuery 선택자란?

선택한 요소들을 변수에 저장하여 사용 가능

## jQuery 응용

### 1. 태그 요소 조작

#### 1) 선택자

#### [ 필터링에 사용 가능한 선택자 ] 1/2

**:eq(n)**    ▪ 선택한 요소 중에서 인덱스가 n인 요소를 선택함

**:gt(n)**    ▪ 선택한 요소 중에서 인덱스가 n보다 큰 요소를 모두 선택함

**:lt(n)**    ▪ 선택한 요소 중에서 인덱스가 n보다 작은 요소를 모두 선택함

**:even**    ▪ 선택한 요소 중에서 인덱스가 짝수인 요소를 모두 선택함

**:odd**    ▪ 선택한 요소 중에서 인덱스가 홀수인 요소를 모두 선택함

**:first**    ▪ 선택한 요소 중에서 첫 번째 요소를 선택함

**:last**    ▪ 선택한 요소 중에서 마지막 요소를 선택함

**:animated**    ▪ 선택한 요소 중에서 애니메이션 효과가 실행 중인 요소를 모두 선택함

**:header**    ▪ 선택한 요소 중에서 h1부터 h6까지의 요소를 모두 선택함

**:lang(언어)**    ▪ 선택한 요소 중에서 지정한 언어의 요소를 모두 선택함

# jQuery 응용

## 1. 태그 요소 조작

### 1) 선택자

## [ 필터링에 사용 가능한 선택자 ] 2/2

- :not(선택자)**    ▪ 선택한 요소 중에서 지정한 선택자와 일치하지 않는 요소를 모두 선택함
- :root**    ▪ 선택한 요소 중에서 최상위 루트 요소를 선택함
- :target**    ▪ 선택한 요소 중에서 웹 페이지 URI의 fragment 식별자와 일치하는 요소를 모두 선택함
- :contains  
(텍스트)**    ▪ 선택한 요소 중에서 지정한 텍스트를 포함하는 요소를 모두 선택함
- :has(선택자)**    ▪ 선택한 요소 중에서 지정한 선택자와 일치하는 자손 요소를 갖는 요소를 모두 선택함
- :empty**    ▪ 선택한 요소 중에서 자식 요소를 가지고 있지 않은 요소를 모두 선택함
- :parent**    ▪ 선택한 요소 중에서 자식 요소를 가지고 있는 요소를 모두 선택함

## jQuery 응용

### 1. 태그 요소 조작

#### 1) 선택자

[ input 요소를 선택할 수 있는 선택자 ] 1/3

:button

- type 속성값이 “button”인 요소를 모두 선택함

:checkbox

- type 속성값이 “checkbox”인 요소를 모두 선택함

:file

- type 속성값이 “file”인 요소를 모두 선택함

:image

- type 속성값이 “image”인 요소를 모두 선택함

:password

- type 속성값이 “password”인 요소를 모두 선택함



## jQuery 응용

### 1. 태그 요소 조작

#### 1) 선택자

#### [ input 요소를 선택할 수 있는 선택자 ] 2/3

:radio

- type 속성값이 “radio”인 요소를 모두 선택함

:reset

- type 속성값이 “reset”인 요소를 모두 선택함

:submit

- type 속성값이 “submit”인 요소를 모두 선택함

:text

- type 속성값이 “text”인 요소를 모두 선택함

:input

- <input>, <textarea>, <select>, <button> 요소를 모두 선택함

## jQuery 응용

### 1. 태그 요소 조작

#### 1) 선택자

#### [ input 요소를 선택할 수 있는 선택자 ] 3/3

:checked

- type 속성값이 “checkbox” 또는 “radio”인 요소 중에서 체크되어 있는 요소를 모두 선택함

:selected

- <option> 요소 중에서 선택된 요소를 모두 선택함

:focus

- 현재 포커스가 가지고 있는 요소를 선택함

:disabled

- 비활성화 되어있는 요소를 모두 선택함

:enabled

- 활성화 되어있는 요소를 모두 선택함

## jQuery 응용

### 1. 태그 요소 조작

#### 2) 탐색

### [ 조상 요소 탐색 ]

- 선택한 요소의 부모(parent) 요소를 선택함

`.parent()`

- 선택한 요소의 조상(ancestor) 요소를 모두 선택함

`.parents()`

- 선택한 요소의 조상 요소 중에서 지정한 선택자에 해당하는 요소 바로 이전까지의 요소를 모두 선택함

`.parentsUntil()`

- 선택한 요소를 포함한 조상 요소 중에서 지정한 선택자에 해당하는 요소 중 가장 첫 번째 요소를 선택함

`.closest()`

## jQuery 응용

### 1. 태그 요소 조작

#### 2) 탐색

#### [ 형제 요소 탐색 ] 1/2

- 선택한 요소의 형제(sibling) 요소 중에서  
지정한 선택자에 해당하는 요소를 모두 선택함

`.siblings()`

- 선택한 요소의 바로 다음에 위치한 형제 요소를 선택함

`.next()`

- 선택한 요소의 다음에 위치한 형제 요소를 모두 선택함

`.nextAll()`

## jQuery 응용

### 1. 태그 요소 조작

#### 2) 탐색

#### [ 형제 요소 탐색 ] 2/2

- 선택한 요소의 형제 요소 중에서 지정한 선택자에 해당하는 요소 바로 이전까지의 요소를 모두 선택함

`.nextUntil()`

- 선택한 요소의 바로 이전에 위치한 형제 요소를 선택함

`.prev()`

- 선택한 요소의 이전에 위치한 형제 요소를 모두 선택함

`.prevAll()`

- 선택한 요소의 형제 요소 중에서 지정한 선택자에 해당하는 요소 바로 다음까지의 요소를 모두 선택함

`.prevUntil()`

## jQuery 응용

### 1. 태그 요소 조작

#### 2) 탐색

#### [ 자손 요소 탐색 ]

- 선택한 요소의 자식(child) 요소를 모두 선택함

`.children()`

- 선택한 요소의 자손(descendant) 요소 중에서 전달받은 선택자에 해당하는 요소를 모두 선택함

`.find()`

## jQuery 응용

### 1. 태그 요소 조작

#### 2) 탐색

#### [ 기타 탐색 ] 1/2

- 선택한 요소의 집합에 전달받은 요소를 추가함

`.add()`

- 선택한 요소의 집합에 바로 전에 선택했던 요소를 추가함

`.addBack()`

- 선택한 요소 집합의 요소마다 전달받은 콜백 함수를 실행함

`.each()`

## jQuery 응용

### 1. 태그 요소 조작

#### 2) 탐색

#### [ 기타 탐색 ] 2/2

- 마지막으로 실행한 메소드의 실행 전 상태로 선택한 요소의 집합을 복원함

`.end()`

- 선택한 요소를 위치시킬 때 기준이 되었던 조상 요소를 선택함

`.offsetParent()`

- 선택한 요소의 자식(child) 요소를 모두 선택함(텍스트 노드와 주석 노드까지 모두 포함함)

`.contents()`



## jQuery 응용

### 2. 스타일 및 이벤트

#### 1) 스타일

#### [ 프로퍼티 설정 ] 1/2

- `css()` 메소드는 선택한 요소 집합의 첫 번째 요소의 스타일 속성값을 반환하거나, 선택한 요소의 스타일 속성을 인수로 전달받은 값으로 설정함

`.css()`

- 선택한 요소 집합의 첫 번째 요소의 지정된 속성(attribute)값을 반환하거나, 선택한 요소의 지정된 속성을 전달받은 값으로 설정함

`.attr()`

## jQuery 응용

### 2. 스타일 및 이벤트

#### 1) 스타일

#### [ 프로퍼티 설정 ] 2/2

- 선택한 요소 집합의 첫 번째 요소의 지정된 프로퍼티(property)값을 반환하거나, 선택한 요소의 지정된 프로퍼티를 전달받은 값으로 설정함

`.prop()`

- 선택한 요소에서 지정된 속성(attribute)을 제거함

`.removeAttr()`

- 선택한 요소에서 지정된 프로퍼티(property)를 제거함

`.removeProp()`

## jQuery 응용

### 2. 스타일 및 이벤트

#### 1) 스타일

#### [ 클래스 설정 ]

- 선택한 요소에 인수로 전달받은 클래스를 추가함

`.addClass()`

- 선택한 요소에서 인수로 전달받은 클래스를 제거함

`.removeClass()`

- 선택한 요소에 클래스가 없으면 인수로 전달받은 클래스를 추가하고, 전달받은 클래스가 추가되어 있으면 제거함

`.ToggleClass()`

- 인수로 전달받은 값이 선택한 요소의 클래스 이름과 일치하는지를 확인함

`.hasClass()`

## jQuery 응용

### 2. 스타일 및 이벤트

#### 2) 이벤트



#### 이벤트란?

사용자의  
마우스  
움직임  
(클릭 등)

사용자가  
텍스트 박스에  
글 쓰는  
행위 등

#### 모든 동작이 이벤트를 발생

- 이벤트가 발생하면 웹페이지에서 특정 동작이 발생하여 웹 브라우저가 그 사실을 알려줌

## jQuery 응용

### 2. 스타일 및 이벤트

#### 2) 이벤트



#### 이벤트 핸들러란?

---

- 특정 요소에 발생하는 이벤트를 처리하기 위해서 이벤트 핸들러라는 함수 작성
- 이벤트 핸들러가 연결된 특정 요소에서 지정된 타입의 이벤트가 발생하면 웹 브라우저는 연결된 이벤트 핸들러를 실행



#### 이벤트 객체란?

---

- 이벤트 핸들러는 이벤트 객체를 인수로 전달받을 수 있음
- 전달받은 이벤트 객체를 이용하여 이벤트의 성질을 결정하거나 이벤트의 기본 동작을 막을 수 있음

## jQuery 응용

### 2. 스타일 및 이벤트

#### 2) 이벤트



#### 이벤트 연결이란?

이벤트와 이벤트 핸들러를  
특정 요소에 연결하는 것

#### [ 이벤트 연결 ]

`.on()` : 특정 요소에 이벤트 연결을 위해 사용

- 선택한 요소에 어떤 타입의 이벤트라도 연결할 수 있음
- 하나의 이벤트 핸들러에 여러 개의 이벤트를 동시에 연결할 수 있음
- 선택한 요소에 여러 개의 이벤트 핸들러와 여러 개의 이벤트를 같이 연결할 수 있음
- 사용자 지정 이벤트를 위해 이벤트 핸들러로 데이터를 넘길 수 있음

## jQuery 응용

### 2. 스타일 및 이벤트

#### 2) 이벤트

#### [ 이벤트 연결 - 마우스 이벤트 메소드 ] 1/4

- 자바스크립트의 “click” 이벤트와 이벤트 핸들러를 연결하거나, 해당 요소에 “click” 이벤트를 발생시킴

`.click()`

- 자바스크립트의 “dblclick” 이벤트와 이벤트 핸들러를 연결하거나, 해당 요소에 “dblclick” 이벤트를 발생시킴

`.dblclick()`

- 자바스크립트의 “mouseenter”와 “mouseleave” 이벤트를 같이 이벤트 핸들러와 연결함

`.hover()`

## jQuery 응용

### 2. 스타일 및 이벤트

#### 2) 이벤트

#### [ 이벤트 연결 - 마우스 이벤트 메소드 ] 2/4

- 자바스크립트의 “mousedown” 이벤트와 이벤트 핸들러를 연결하거나, 해당 요소에 “mousedown” 이벤트를 발생시킴

`.mousedown()`

- 해당 요소 위로 마우스가 진입할 때 발생하는 이벤트와 이벤트 핸들러를 연결하거나, 해당 요소에 마우스 진입 이벤트를 발생시킴

`.mouseenter()`

- 해당 요소에서 마우스가 빠져나갈 때 발생하는 이벤트와 이벤트 핸들러를 연결하거나, 해당 요소에 마우스가 빠져나가는 이벤트를 발생시킴

`.mouseleave()`



## jQuery 응용

### 2. 스타일 및 이벤트

#### 2) 이벤트

#### [ 이벤트 연결 - 마우스 이벤트 메소드 ] 3/4

- 자바스크립트의 “mousemove” 이벤트와 이벤트 핸들러를 연결하거나, 해당 요소에 “mousemove” 이벤트를 발생시킴

`.mousemove()`

- 자바스크립트의 “mouseout” 이벤트와 이벤트 핸들러를 연결하거나, 해당 요소에 “mouseout” 이벤트를 발생시킴

`.mouseout()`

- 자바스크립트의 “mouseover” 이벤트와 이벤트 핸들러를 연결하거나, 해당 요소에 “mouseover” 이벤트를 발생시킴

`.mouseover()`

## jQuery 응용

### 2. 스타일 및 이벤트

#### 2) 이벤트

#### [ 이벤트 연결 - 마우스 이벤트 메소드 ] 4/4

- 자바스크립트의 “mouseup” 이벤트와 이벤트 핸들러를 연결하거나, 해당 요소에 “mouseup” 이벤트를 발생시킴

`.mouseup()`

#### [ 이벤트 연결 - 키보드 이벤트 메소드 ] 1/2

- 자바스크립트의 “keydown” 이벤트와 이벤트 핸들러를 연결하거나, 해당 요소에 “keydown” 이벤트를 발생시킴

`.keydown()`

- 자바스크립트의 “keyup” 이벤트와 이벤트 핸들러를 연결하거나, 해당 요소에 “keyup” 이벤트를 발생시킴

`.keyup()`

## jQuery 응용

### 2. 스타일 및 이벤트

#### 2) 이벤트

#### [ 이벤트 연결 – 키보드 이벤트 메소드 ] 2/2

- 자바스크립트의 “keypress” 이벤트와 이벤트 핸들러를 연결하거나, 해당 요소에 “keypress” 이벤트를 발생시킴

`.keypress()`

#### [ 이벤트 연결 – 입력 양식 이벤트 메소드 ] 1/3

- 자바스크립트의 “blur” 이벤트와 이벤트 핸들러를 연결하거나, 해당 요소에 “blur” 이벤트를 발생시킴

`.blur()`

- 자바스크립트의 “change” 이벤트와 이벤트 핸들러를 연결하거나, 해당 요소에 “change” 이벤트를 발생시킴

`.change()`

## jQuery 응용

### 2. 스타일 및 이벤트

#### 2) 이벤트

#### [ 이벤트 연결 – 입력 양식 이벤트 메소드 ] 2/3

- 자바스크립트의 “select” 이벤트와 이벤트 핸들러를 연결하거나, 해당 요소에 “select” 이벤트를 발생시킴

`.select()`

- 자바스크립트의 “submit” 이벤트와 이벤트 핸들러를 연결하거나, 해당 요소에 “submit” 이벤트를 발생시킴

`.submit()`

## jQuery 응용

### 2. 스타일 및 이벤트

#### 2) 이벤트

#### [ 이벤트 연결 – 입력 양식 이벤트 메소드 ] 3/3

- 자바스크립트의 “focus” 이벤트와 이벤트 핸들러를 연결하거나, 해당 요소에 “focus” 이벤트를 발생시킴

`.focus()`

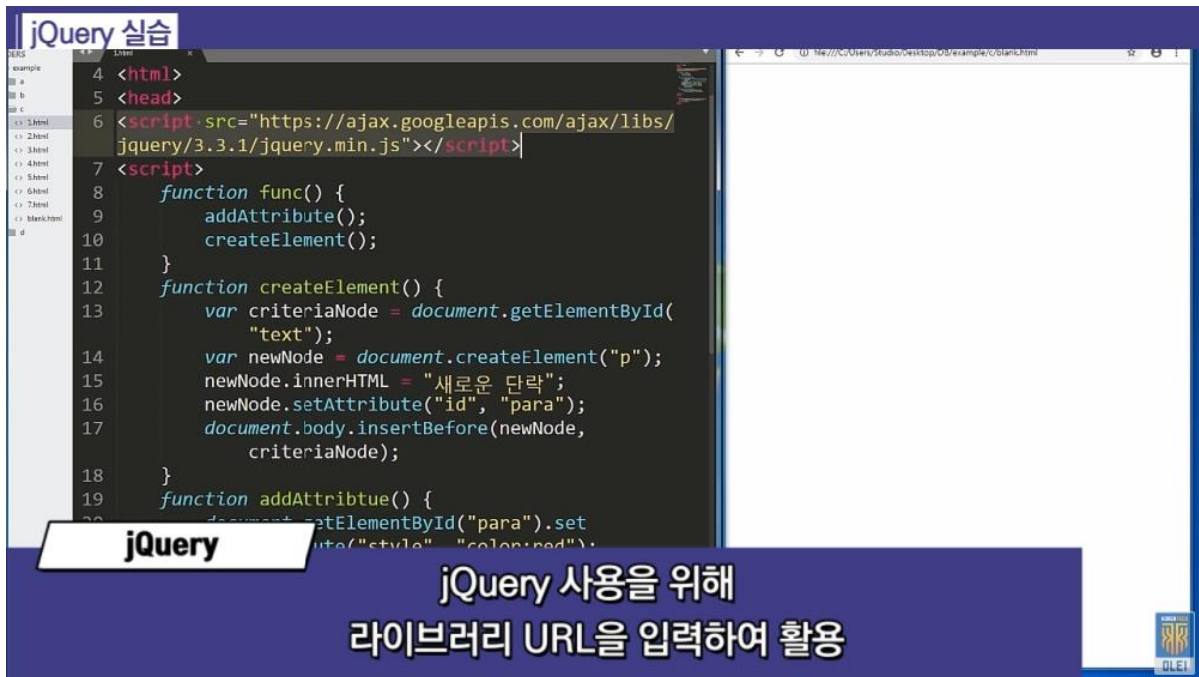
- “focusin” 이벤트와 이벤트 핸들러를 연결하거나, 해당 요소에 “focusin” 이벤트를 발생시킴

`.focusin()`

- “focusout” 이벤트와 이벤트 핸들러를 연결하거나, 해당 요소에 “focusout” 이벤트를 발생시킴

`.focusout()`

# jQuery 실습



※ 실습 영상은 콘텐츠에서 확인할 수 있습니다.

# 핵심정리

## 1. jQuery 개요

- 자바스크립트 언어를 간편하게 사용할 수 있도록 단순화시킨 오픈 소스 기반의 자바스크립트 라이브러리로 HTML의 클라이언트 사이드 조작을 단순화 하도록 설계됨
- jQuery 라이브러리는 jQuery라는 이름의 객체로 생성됨
- 중복되는 기능과 코드가 엉키는 것을 피하고 이미 개발된 많은 플로그인을 쉽고 빠르게 이용할 수 있음
- 브라우저마다 다르게 발생하는 이벤트, 객체 등을 처리하고자 여러 개의 메서드를 사용하지 않고 jQuery에서 제공되는 메서드 또는 문장으로 간단하게 해결 가능함
- jQuery 파일은 자바스크립트 파일(.js)형태로 존재하며 사용하기 위하여 웹페이지에 로드해야 함
- \$ 기호는 jQuery를 의미하고 이를 접근할 수 있게 하는 식별자이고 동작 함수를 정의하여 선택된 요소에 원하는 동작을 설정함

## 2. jQuery 응용

- 선택자를 이용하여 적용할 요소를 선택하고 선택자는 CSS선택자 및 jQuery 선택자를 이용함
- HTML 요소를 탐색할 수 있는 기능을 제공하며 요소의 스타일(프로퍼티) 를 설정할 수 있음
- 사용자가 마우스를 움직이거나 클릭하거나 텍스트 박스에 글을 쓰는 등 모든 동작들을 이벤트라고 하며 특정 요소에 이벤트 핸들러를 연결하여 지정된 이벤트를 웹브라우저에서 동작하게 함