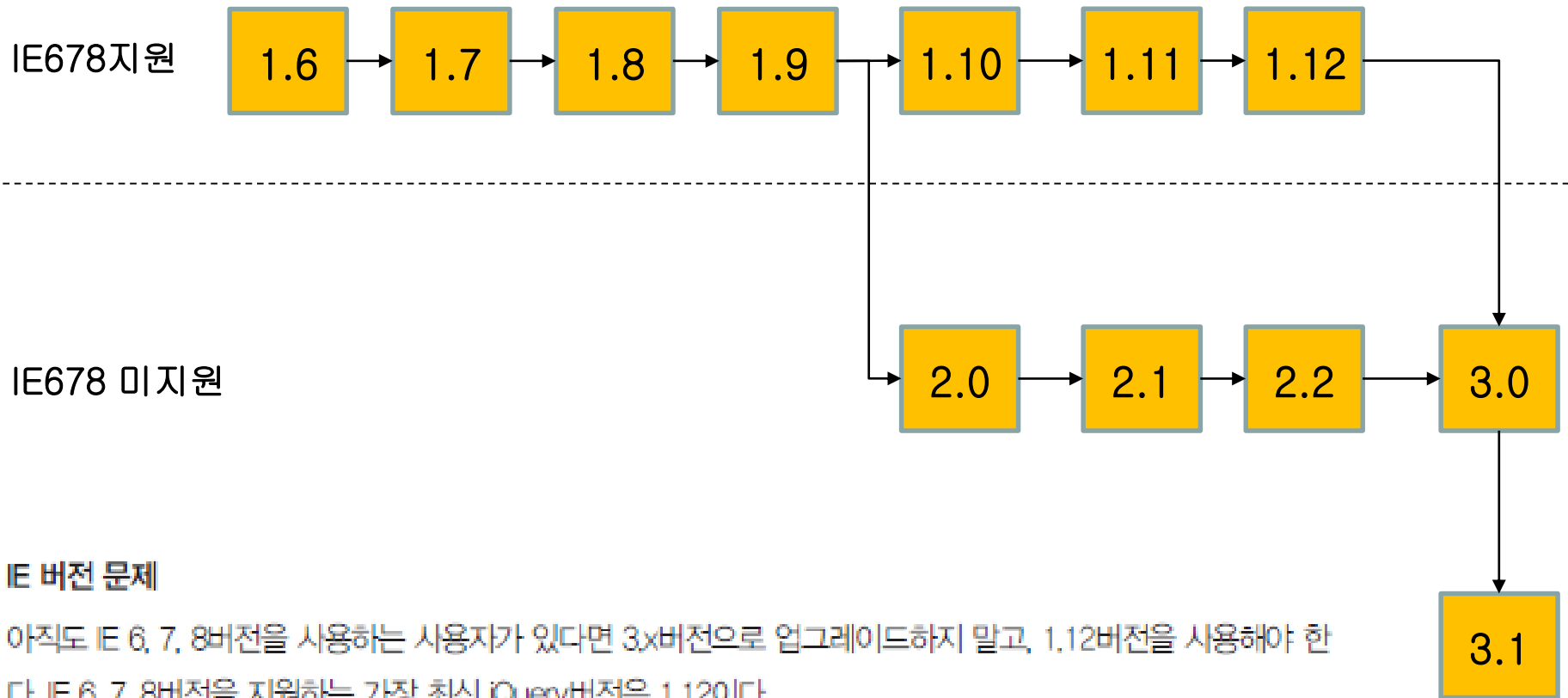


# jQuery 버전별 특징



## jQuery 버전



# jQuery 3.X(1)



## ■ 기능 변경

- `show()`, `hide()` 메서드의 기능이 변경되었다.
- jQuery Deferred 객체가 ECMAScript 6의 Promise, Promise/A+ 기능으로 변경되었다.
- 이벤트 핸들러 일부는 기능 지원이 중단되었다
- 몇몇 Selector의 대폭적인 성능 향상이 이루어졌다.

```
01: <!DOCTYPE html>
02: <html lang="">
03: <head>
04: <meta charset="utf-8">
05: <title></title>
06: <style>
07:   #hello { font-size:30pt }
08: </style>
09: <script type="text/javascript" src="https://code.jquery.com/jquery-3.0.0.js"></script>
10: <script type="text/javascript">
11: $(document).ready(function() {
12:   $("#toggle").click(function() {
13:     $("#hello").toggle();
14:   });
15: });
16: </script>
17: </head>
```

# jQuery 3.X(2)



## ■ readyException 처리

```
$.readyException = function(error) {  
    //ready 이벤트 핸들러에서의 오류 발생 시 에러 처리를 수행할 코드  
}
```

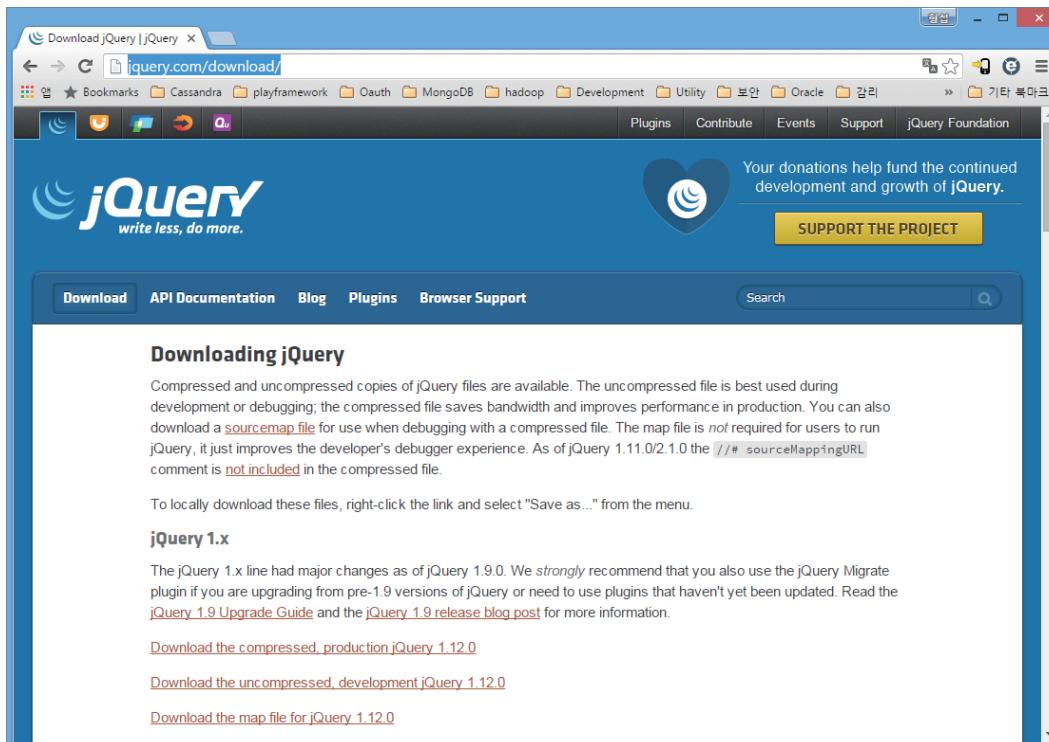
# jQuery 내려 받기(1)



## ■ http://jquery.com/download/에서 다운로드

### ■ 운영버전과 개발 버전

```
<script type="text/javascript" src="[jquery 파일 경로]"></script>
```



# jQuery 내려 받기(2)



## ■ node.js 와 bower를 이용한 다운로드

```
bower install jquery
```

```
bower install http://code.jquery.com/jquery-1.10.2.min.js
```

```
bower install http://code.jquery.com/jquery-1.10.2.js
```

```
<script type="text/javascript" src="bower_components/jquery/dist/jquery.min.js"></script>
```

### Node.js, npm, bower란?

Node.js는 확장성 있는 애플리케이션을 개발하기 위해 사용하는 자바스크립트 기반 소프트웨어 플랫폼이다. 브라우저가 아닌 단일 머신에서 실행하기 위한 자바스크립트라고 생각하면 된다. 자바스크립트로 복잡한 애플리케이션을 개발하기 위해서 다양한 모듈을 내려받아 사용하는데, 이 모듈을 관리해주는 Node Package Manager가 npm이다. npm을 통해서 필요한 모듈을 네트워크를 통해서 내려받을 수 있다. npm은 자바스크립트를 위한 기본적인 패키지 관리자로 사용된다.

bower는 웹사이트 개발을 위한 다양한 구성 요소를 관리하는 패키지 관리자이다. bower를 사용하기 위해서 node.js, npm으로 bower를 인스톨해야 한다.

# CDN 이용 방법(1)



- CDN(Content Delivery Network)
- 반드시 jquery.js 파일을 내려받을 필요는 없음
  - <script> 태그로 원격서버에 캐싱된 파일을 직접 참조할 수 있다.

- jQuery CDN <http://code.jquery.com>
- Google CDN <https://developers.google.com/speed/libraries/#jquery>
- Microsoft CDN [http://www.asp.net/ajax/cdn#jQuery\\_Releases\\_on\\_the\\_CDN\\_0](http://www.asp.net/ajax/cdn#jQuery_Releases_on_the_CDN_0)
- CDNJS CDN <http://cdnjs.com/libraries/jquery/>
- jsDelivr CDN <http://www.jsdelivr.com/projects/jquery>

## ■ jQuery CDN 참조 예

```
<script type="text/javascript"  
  src="https://code.jquery.com/jquery-3.1.0.js"></script>
```

# jQuery Migration Plugin



## ■ jQuery 1.9 이전 버전에서 최신 버전으로 업그레이드할 때 호환성 문제 발생 가능성 존재

- 이전 버전의 기능을 사용하고 싶다면 jQuery Migration Plugin!!

```
<script type="text/javascript" src="http://code.jquery.com/jquery.js"></script>  
<script src="https://code.jquery.com/jquery-migrate-1.3.0.js"></script>
```

- 최신 버전에서 제거된 기능 테스트(ex:live() 메서드)

```
< function ( types, data, fn ) {  
    migrateWarn("jQuery.fn.live() is deprecated");  
    if ( oldLive ) {  
        return oldLive.apply( this, arguments );  
    }  
    jQuery( this.context ).on( types, this.selector, data, f...
```

# jQuery 작성 템플릿(1)



## ■ 두가지 유형

- HTML 문서가 모두 로딩되어 ready 상태가 되면 초기화 함수가 실행되도록 document 객체에 ready 이벤트를 설정하는 방식
- <script /> 태그를 <body> 태그 안의 마지막에 배치하는 방식

[ 예제 06-02 : \$(document).ready()를 사용한 기본 템플릿 ]

```
<!DOCTYPE html>
<html lang="">
<head>
<meta charset="utf-8">
<title>기본 템플릿</title>
<script type="text/javascript" src="https://code.jquery.com/jquery-3.1.0.js"></script>
<script type="text/javascript">
$(document).ready(function() {

});
</script>
</head>
<body>

</body>
</html>
```



# jQuery 작성 템플릿(2)



## ❖ <body /> 태그 마지막에 코드 배치

[ 예제 06-04 : <body /> 태그 마지막에 배치하는 방식 ]

```
01: <!DOCTYPE html>
02: <html lang="">
03: <head>
04: <meta charset="utf-8">
05: <title>06-03</title>
06: </head>
07: <body>
08:     <button id="btn1">클릭1</button>
09:     <button id="btn2">클릭2</button>
10:     <div id="a"></div>
11:     <script type="text/javascript" src="https://code.jquery.com/jquery-3.1.0.js"></script>
12:     <script type="text/javascript">
13:         var str1 = "hello";
14:         var str2 = "world";
15:         $("#btn1").click(function() {
16:             $("#a").html(str1);
17:         });
18:         $("#btn2").click(function() {
19:             $("#a").html(str2);
20:         });
21:     </script>
22: </body>
23: </html>
```

# jQuery 작성 템플릿(3)



- document 객체의 ready 이벤트에 여러개 함수를 바인딩

[ 예제 06-05 : 여러 개의 \$(document).ready(func) 사용 ]

```
01: <!DOCTYPE html>
02: <html lang="">
03: <head>
04: <meta charset="utf-8">
05: <title>Template</title>
06: <script type="text/javascript" src="https://code.jquery.com/jquery-3.1.0.js"></script>
07: <script type="text/javascript">
08: $(document).ready(function() {
09:     var str = "hello";
10:     $("#btn1").click(function() {
11:         $("#a").html(str);
12:     });
13: });
14:
15: $(document).ready(function() {
16:     var str = "world";
17:     $("#btn2").click(function() {
18:         $("#a").html(str);
19:     });
20: });
21: </script>
22: </head>
23: <body>
24:     <button id="btn1">클릭1</button>
25:     <button id="btn2">클릭2</button>
26:     <div id="a"></div>
27: </body>
28: </html>
```

# jQuery 작성 템플릿(4)



- `$(document).ready(function() {})` == `$(function() {})`

[ 예제 06-06 ]

```
$(function() {  
    var str1 = "hello";  
    var str2 = "world";  
    $("#btn1").click(function() {  
        $("#a").html(str1);  
    });  
    $("#btn2").click(function() {  
        $("#a").html(str2);  
    });  
});
```

# \$ 식별자 충돌 피하기(1)



## ■ \$.noConflict()

- jQuery에서 \$ 식별자 사용을 포기함

[예제 06-07 : \$.noConflict() 메서드]

```
<script src="https://ajax.googleapis.com/ajax/libs/prototype/1.7.3.0/prototype.js"></script>
<script type="text/javascript" src="https://code.jquery.com/jquery-3.1.0.js"></script>
<script type="text/javascript">
$.noConflict();
</script>
```

```
> $
< function $(element) {
    if (arguments.length > 1) {
        for (var i = 0, elements = [], length = arguments.length; i < length; i++)
            elements.push($(arguments[i]));
        return elements;
    }
    ...
> jQuery
< function ( selector, context ) {

    // The jQuery object is actually just the init constructor 'enhanced'
    // Need init if jQuery is called (just allow error to be thrown if not included)
    return ne...
```

# \$ 식별자 충돌 피하기(2)



## ■ \$ 식별자의 포기가 쉽지 않다면?

- \$(document).ready() 를 이용하여 \$ 사용
- 즉시 실행 함수를 이용해 \$ 사용

[ 예제 06-08 ]

```
<script src="https://ajax.googleapis.com/ajax/libs/prototype/1.7.3.0/prototype.js"></script>
<script type="text/javascript" src="https://code.jquery.com/jquery-3.1.0.js"></script>
<script type="text/javascript">
$.noConflict();
console.log($);    //prototype!!
jQuery(document).ready(function($) {
    console.log($);    //jQuery!!
});
</script>
```

[ 예제 06-09 ]

```
<script src="https://ajax.googleapis.com/ajax/libs/prototype/1.7.3.0/prototype.js"></script>
<script type="text/javascript" src="https://code.jquery.com/jquery-3.1.0.js"></script>
<script type="text/javascript">
$.noConflict();
console.log($);    //prototype!!
(function($) {
    console.log($);    //jQuery!!
})(jQuery);
</script>
```

# 이 책에서 주로 사용할 템플릿은?



## ■ 예제 06-02 형태

- 새로운 예제를 작성할 때 기본적인 언급이 없다면 이 파일에서 시작

[ 예제 06-02 : \$(document).ready()를 사용한 기본 템플릿 ]

```
<!DOCTYPE html>
<html lang="">
<head>
<meta charset="utf-8">
<title>기본 템플릿</title>
<script type="text/javascript" src="https://code.jquery.com/jquery-3.1.0.js"></script>
<script type="text/javascript">
$(document).ready(function() {

});
</script>
</head>
<body>

</body>
</html>
```