

Javascript API

FORM API

- autocomplete
 - : on 과 off 두가지 값을 가진다. 기본값은 on.
 - : <input>요소에 off로 설정하면 자동완성 기능이 비활성화되어 이전 항목을 보여주지 않는다.
- novalidate
 - : html 폼은 자동으로 유효성검증이 내장되어 실행됨.
 - : 이 기능을 실행하지 않으려면 novalidate 속성을 사용.
 - :특정 폼 태그에 사용하려면 formnovalidate 속성을 사용한다.
- <input> - **type**
email, search, url, tel, number, range, date, week, month, time, datetime, color
- <input> - 단일 속성(속성을 표기하면 true로 인식)
placeholder, required, multiple, autofocus(페이지 시작시 자동으로 커서가 이동함.)
, pattern(정규식을 넣어서 데이터를 검증할 수 있다.)
ex) <input pattern="[0-9]{5}" name="post"> → 우편번호 5자리

정규식

- 정규식

- 특정한 규칙을 가진 문자열의 집합을 표현하는데 사용하는 형식 언어

- JavaScript에서 사용하는 방법

- JavaScript 버전이 1.1이하 버전일 경우에는 정규표현식을 사용할 수 없다.
- 정규표현식을 사용하는 방법으로는 두 가지가 방법
- 첫 번째로는 'RegExp'객체를 이용하는 방법

// RegExp 객체를 이용하는 방법

```
var objectInitializer = new RegExp('정규표현식',['Flag']);
```

- 두 번째로는 객체초기화(Object Initializer)를 사용하는 방법입니다. 주로 입력된 표현식이 거의 바뀌지 않는 상수 형태의 표현식을 사용할 때 사용합니다.}

// 객체초기화(Object initializer) 방법

```
var regExp = /정규표현식/[Flag];
```

정규식

- Flag의 종류

- 자주 사용하는 Flag는 밑의 3종류가 있으며 Flag를 사용을 하지 않을 수도 있다.
- 만약 Flag를 설정 하지 않을 경우에는 문자열 내에서 검색대상이 많더라도 한번만 찾고 끝나게 됩니다.

Flag	설명
g	Global - 문자열 내의 모든패턴을 찾습니다.
i	Ignore Case -문자열의 대소문자를 구별하지 않습니다.
m	Multi Line - 문자열이 행이 바뀌어도 찾습니다.

정규식

- **exec()**
 - 일치하는 문자열을 찾는 RegExp 메소드. 정보를 가지고 있는 배열을 반환.
- **replace()**
 - 일치하는 문자열을 찾는 String 메소드. 일치하는 문자열을 replacement 로 대체합니다.
- **match()**
 - 일치하는 문자열을 찾는 String 메소드입니다. 정보를 가지고 있는 배열을 반환하거나 일치하지 않는 부분을 null로 반환합니다. i 플래그를 사용해서 대/소문자 구분없이 매치 가능합니다.
- **test()**
 - 일치하는 문자열을 검사하는 RegExp 메소드입니다. true나 false를 반환합니다.

정규식

//사용자가 입력한 ID가 한글이 포함되어 있는지 Check 합니다.

```
function idCheck () {  
    // 입력한 ID를 Check하기 위해 가져옵니다.  
    var titleCheck = $("#titleId").val;  
    // 정규표현식으로 한글만 선택하도록 만듭니다.  
    var languageCheck = /[ㄱ-ㅎ|ㅏ-ㅣ|가-힣]/;  
    // 입력한 ID와 정규표현식을 비교하여 한글 여부를 판단합니다.  
    // test외에도 search ,exec , match등을 사용할 수 있습니다.  
    if (languageCheck.test(titleCheck)) {  
        alert("ID에 한글이 포함되어 있습니다.");  
        return;  
    }  
    ...  
}
```

한글만 입력 가능	var 변수명 = /^[가-힣]+\$/;	
숫자만 입력 가능	var 변수명 = /^[0-9]*\$/;	
영문만 입력 가능	var 변수명 = /^[a-zA-Z]+\$/	
한글 + 영문만 입력 가능	var 변수명 = /^[가-힣a-zA-Z]+\$/;	
아이디(영문+숫자)	var 변수명 = /^[a-zA-Z0-9]{4,12}\$/;	4글자 이상, 12자미만, 영어대소문자,숫자,-, _ 사용가능
비밀번호 (영문/숫자+특문)	var 변수명 = /^(?=.*[a-zA-Z])(?=.*\d)(?=.*[!@#\$%^&*~` ;:,~'`/(){}]).{6,20}\$/	6글자 이상 20자 미만,최소 1개의 숫자 혹은 특수문자 포함
이메일 주소	var 변수명 = /^[a-z0-9_+.-]+@[a-z0-9-]+\.[a-z0-9]{2,4}\$/	
숫자만 입력가능	var 변수명 = /^[0-9]/gi;	숫자만 입력
전화번호	var 변수명 = /^wd{2,3}-wd{3,4}-wd{4}\$/;	ex)123-123-1234
핸드폰입력	var 변수명 = /^01([01 6 7 8 9]?)-?([0-9]{3,4})-?([0-9]{4})\$/;	ex)010-0000-0000
url입력	var 변수명 = /^(file gopher news nntp telnet https? ftp):w/w/([a-z0-9-]+\.[a-z0-9]{2,4}).*\$/;	

FORM API

- `setCustomValidity()` : 자체적인 유효성 검사도구 개발 (6-23)

- `invalidate` 이벤트

사용자가 폼을 제출할 때 마다 유효하지 않은 필드가 감지되면 이벤트가 발생 (6-24)

- 실시간 유효성 검증

`ValidityState` 객체의 몇가지 속성을 이용 (6-25)

지오 로케이션 API

- 애플리케이션에 접근하는 기기의 물리적인 위치를 확인 하는데 사용된다.
- IP 주소와 같은 네트워크 신호부터 GPS까지 이러한 정보를 얻는 다양한 메서드가 존재한다.
- 리턴값에는 위도와 경도 좌표 정보가 포함되어 있는데, 이는 지오로케이션 API와 구글 지도 같은 외부 지도와 메쉬업되어 실질적인 애플리케이션 구축을 위한 매우 상세한 지역 정보의 접근을 가능하게 한다.

지오 로케이션 API

- 메서드 (9-1)
 - `getCurrentPostion(location, error, configuration)`
 - `watchPostion(location, error, configuration)`
 - `clearWatch(id)`
- 객체
 - **Position**
 - `coords`
 - `timestamp`
 - **PositionError**
 - `message`
 - `error`
 - `PERMISSION_DENIED`
 - `POSITION_UNAVAILABLE`
 - `TIMEOUT`

스토리지 API

- 웹 스토리지 Web Storage와 Indexed Database API는 저장을 목적으로 개발
- 웹 스토리지 : 쿠키에 대한 개선
 - sessionStorage
 - localStorage
- sessionStorage
 - : 페이지 세션 제한시간이 넘어도 일관성을 유지하고 또 다른 윈도우를 여는 등의 잘못된 사용이나 장애가 발생한 경우에도 임시정보를 안전하게 유지 (장바구니) (10-2-6)

스토리지 API

- 반대로 localStorage속성을 사용자의 기기에 큰 파일을 저장할 수 있게 한다.
- 저장된 정보는 영구적이며 보안상의 이유를 제회하고는 만료되지 않는다.
- sessionStorage 와 localStorage 모두 쿠키의 기능을 대체하고 쿠키의 제약사항을 극복한다. (10-2-7)

스토리지 API

- 메서드
 - `setItem(key, value)`
 - `getItem(key, value)`
 - `key(index)`
 - `removeItem(key)`
 - `clear()`
- 프로퍼티
 - `length`
- 이벤트
 - `storage`

파일 API

- 파일을 관리하는 머차지 API를 제공
- 사용자의 기기에서 파일을 읽고, 처리하고, 생성하는 작업이 가능
- 3가지 API사용
- 현재파일 API
- 파일 API : 디렉터리와 시스템
- 파일 API : 라이터

파일 API

- Blob 인터페이스(파일 API)
- size
- type
- slice(start, length, type)
- File 인터페이스 (파일 API)
- name

드레그 엔 드롭 API

- 데스크 톱의 일반적인 액션을 웹에서도 가능하게 개발되었다.
- 몇줄의 코드만으로 요소에서 다른 요소로 드레그 엔 드롭이 가능하다
- 그래픽 뿐 아니라 텍스트, 링크, 파일, 데이터도 가능하다.

드래그 앤 드롭 API

- 드래그 앤드 드롭 작업시 발생 이벤트
 - dragstart
 - drag
 - dragend
- 동일한 작업 동안에 타깃에서 발생하는 이벤트
 - dragenter
 - dragover
 - drop
 - dragleave
- 파일 (8- 9)

드레그 앤 드롭 API

- 메서드
 - setData(type, data)
 - getData(type)
 - clearData(type)
 - setDragImage(element, x,y)
- 프로퍼티
 - types
 - files
 - dropEffect
 - effectAllowed
- 파일 (8- 9)

커뮤니케이션 API

- XMLHttpRequest API 레벨 2
- 크로스 도큐먼트 메시징 API 레벨 2
- 웹소켓 API
- 인터넷 통신의 문제
 - 에이젝스 애플리케이션용 API 는 서로 다른 브라우저에 대해 구현하기 복잡하고 불완전
 - XMLHttpRequest API 레벨 2 로 해결
 - 관련없는 어플리케이션간의 통신은 불가능
 - 크로스 도큐먼트 메시징 API 레벨 2로 해결
 - 실시간으로 서버를 접근하는 효과적인 양방향 통신방법이 없다.
 - 웹소켓 API로 채팅방과 같은 실시간 통신이 가능

캔버스 API

- 캔버스 AIP는 그리기 API로 기본적인지만 강력한 그리기 기능을 제공
- 동적으로 그래픽을 생성하고 렌더링을 하며, 애니메이션을 생성
- 비트맵을 생성
- 픽셀 단위의 이미지는 이러한 목적에 특화된 메서드와 함수의 집합으로 생성되고 처리된다.