



Database Systems

DB를 활용한 웹 프로그래밍

- CSS 및 JavaScript 기초



인하대학교



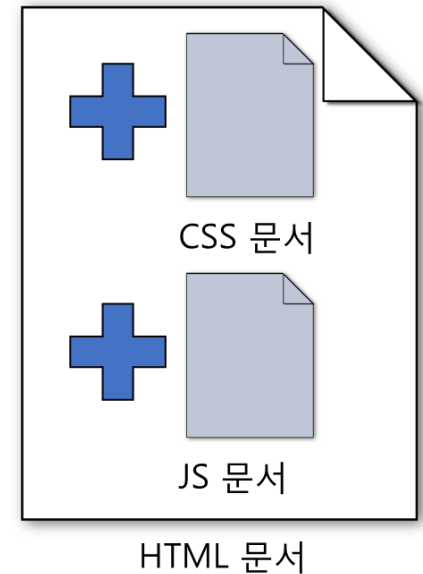
Outline

- 클라이언트 사이드 언어
- DOM 구조의 활용
- CSS 기초
- JavaScript 기초

클라이언트 사이드 언어

- DHTML(Dynamic HTML)이란?

- HTML, CSS, JavaScript를 조합하여 동적 웹사이트를 제작하는 방법

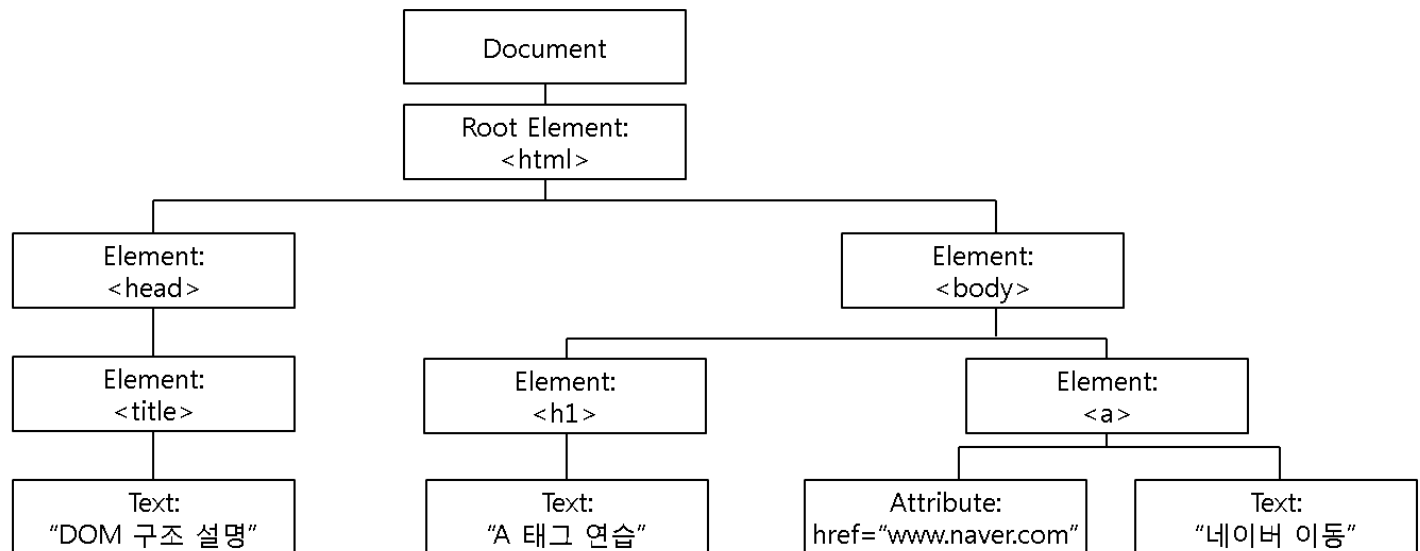


DOM 구조의 활용

• DOM 구조

– 브라우저가 HTML을 관리하기 위해서 사용함

```
<html>
  <head>
    <title>DOM 구조 설명</title>
  </head>
  <body>
    <h1>A태그 연습</h1>
    <a href="www.naver.com">네이버 이동</a>
  </body>
</html>
```

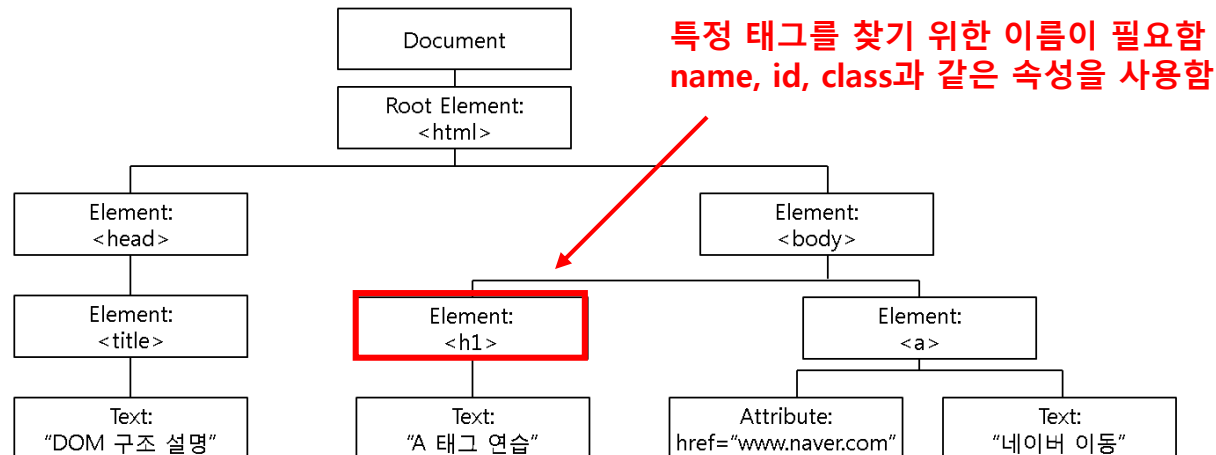


DOM 구조의 활용 (cont'd.)

• HTML, CSS, JavaScript의 통합

- CSS와 JavaScript는 DOM 구조를 탐색하고 특정 태그에 접근하여 디자인 및 기능을 보충함

```
<html>
  <head>
    <title>DOM 구조 설명</title>
  </head>
  <body>
    <h1>A태그 연습</h1>
    <a href="www.naver.com">네이버 이동</a>
  </body>
</html>
```



DOM 구조의 활용 (cont'd.)

• Name, Id, Class 속성

- 모든 태그에서 사용 가능한 속성
- DOM구조에서 특정 태그를 찾을 때 사용됨
 - Name : 태그의 이름(ASP, JavaScript 에서 활용)
 - Id : 태그의 유일한 이름(CSS, JavaScript 에서 활용)
 - Class : 태그의 그룹을 나타내는 이름(CSS, JavaScript 에서 활용)
- 예시

```
<input type="text" name="loginID" id="loginID" class="LoginIDtype1"/>
```

- 보통 name과 id를 같은 것으로 지정하고, class를 그룹명으로 지정
- ASP에서는 Name만 이용 가능하므로 name 속성이 필요
- CSS, JavaScript 에서는 보통 Id, Class 속성을 사용함

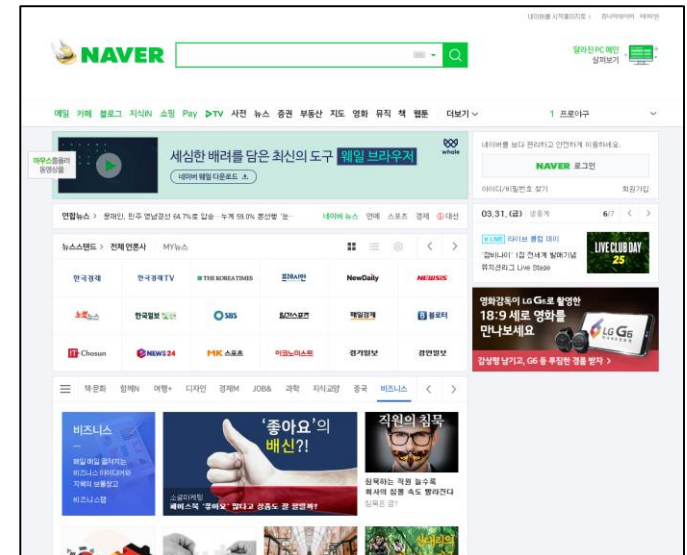
CSS 기초

• CSS이란?

- HTML를 꾸미는 역할을 하는 마크업 언어
- ID와 CLASS 속성을 통해 DOM 구조를 탐색하여 특정 태그를 찾고 해당 태그를 꾸밈



<HTML>

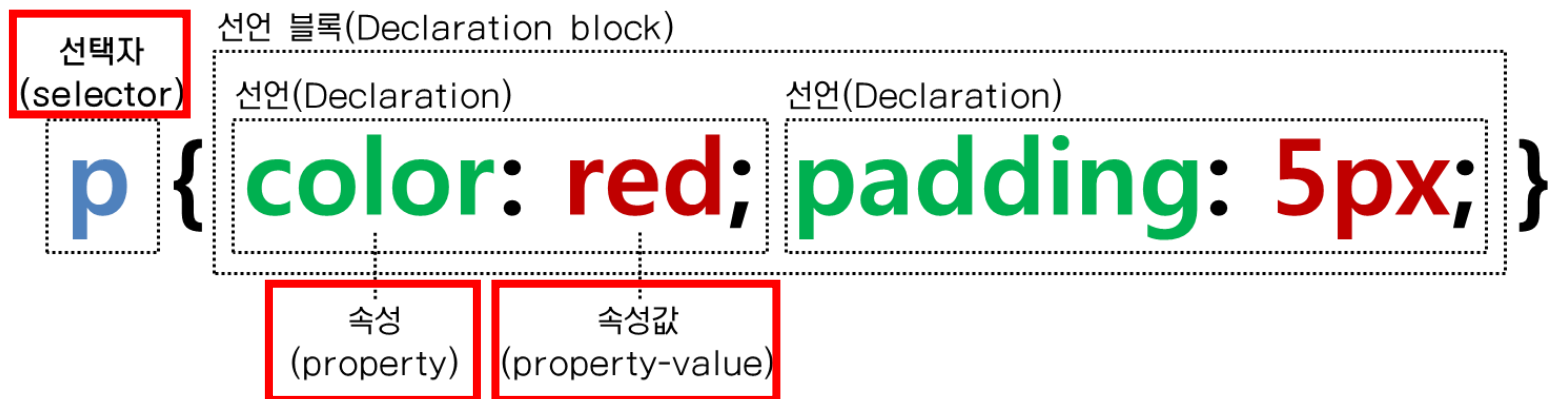


<HTML+CSS>

CSS 기초 (cont'd.)

• CSS의 구성 요소

- 선택자 : DOM구조에서 특정 태그를 찾는 방법
- 특정 태그를 찾고, 속성과 속성값을 통해 색상, 글자 크기, 간격 등 다양한 디자인적 요소를 설정



Selector 참고) <http://www.nextree.co.kr/p8468/>

Property 참고) <https://ofcourse.kr/css-course/속성>

CSS 기초 (cont'd.)

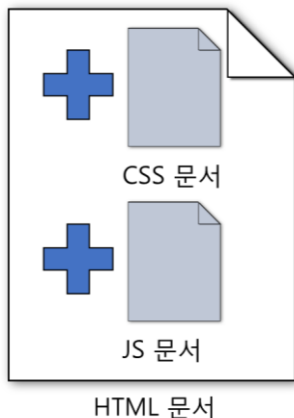
• CSS Selector

| 선택자 이름 | 패턴 | 의미 |
|----------------------------------------------------|-------------|----------------------------|
| 전체 선택자 | * | 모든 태그를 선택 |
| 태그 선택자 | Tag_Name | 태그 이름이 Tag_Name인 태그 선택 |
| 클래스 선택자 | .Class_Name | 클래스 속성값이 Class_Name인 태그 선택 |
| 아이디 선택자 | #ID_Name | 아이디 속성값이 ID_Name인 태그 선택 |
| - 이외에도 많은 선택자가 있지만, 위의 4가지 선택자만으로도 충분히 프로그래밍할 수 있음 | | |

CSS 기초 (cont'd.)

• 예제 1) 로그인 화면 꾸미기 (코드)

– HTML 문서(login.html)



```
<html>
  <head>
    <title> CSS 적용하기 </title>
    <link rel="stylesheet" type="text/css" href= " CSS문서경로 ">
  </head>
  <body>
    <div id="Layer1">
      ID : <input id="loginID" type="text"/> <br/>
      PW : <input id="loginPW" type="password"/> <br/>
      <input id="loginButton" type="button" value="로그인"/>
    </div>
  </body>
</html>
```

– CSS 문서(loginStyle.css)

```
#Layer1 {left:300px; top:300px; width:300px; height:100px; background:#737373;}

#loginID {color:red; width:150px; height:20px;}
#loginPW {color:blue; width:170px; height:20px;}
#loginButton {color:white; background:black;}
```

CSS 기초 (cont'd.)

- 예제 1) 로그인 화면 꾸미기 (웹페이지)

ID :

PW :

CSS 파일 Link

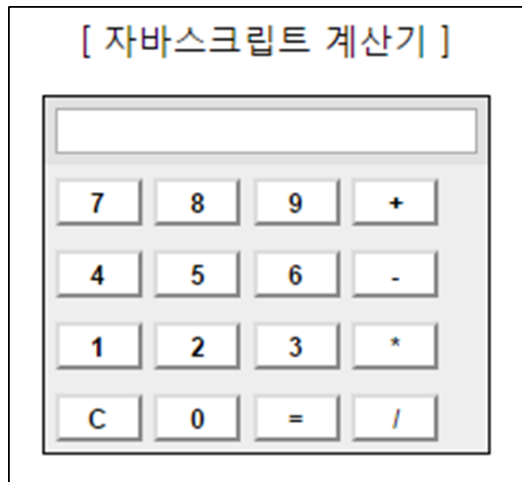
ID :

PW :

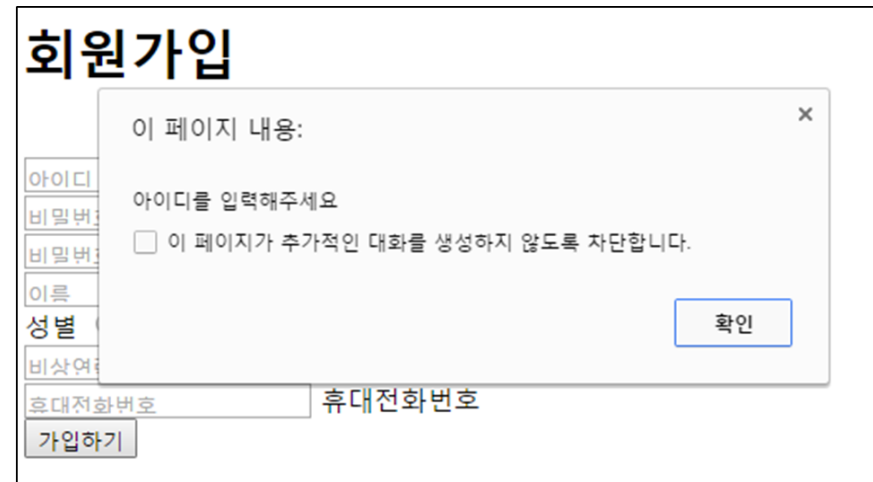
JavaScript 기초

• JavaScript(JS)이란?

- HTML의 기본 골격에 여러 가지 기능을 붙여넣는 역할을 하는 프로그래밍 언어
- ID와 CLASS 속성을 통해 DOM 구조를 탐색하여 특정 태그를 찾고 해당 태그에 기능을 부여



<HTML+JS로 계산 기능 추가>



<HTML+JS로 메시지 박스 기능 추가>

JavaScript 기초 (cont'd.)

• JavaScript의 문법

– 리터럴 자료형

- String (문자형) 예) “우리 나라”, ‘우리 나라’
- Number (숫자형) 예) 1534(정수), 3.14(실수)
- Bool (논리형) 예) true(참), false(거짓)
- Null (비워진 데이터) 예) null

– 변수

- 입력되는 리터럴 값에 따라 변수의 자료형이 변경된다.
 - var 변수이름=값 또는 변수 이름;
 - var index=10;
 - var str=“자바스크립트”;

– 배열

- 특정 데이터 타입이 없음

```
var myArray = new Array();  
myArray[0] = new Array("a", "b", "c");  
myArray[1] = new Array(1, 2, 3);
```



| | | | |
|------------|--------------------|--------------------|--------------------|
| myArray[0] | myArray[0][0] a | myArray[0][1] b | myArray[0][2] c |
| myArray[1] | myArray[1][0] 1 | myArray[1][1] 2 | myArray[1][2] 3 |

JavaScript 기초 (cont'd.)

• JavaScript의 문법

– 연산자

- 산술 연산자
 - +, -, *, /, %
- 문자열 연산자
 - “문자열A” + “문자열B” = “문자열A문자열B”
- 증감 연산자
 - A++, ++A, A--, --A
- 비교 연산자
 - ==, !=, <, >, <=, >=
- 논리 연산자
 - A && B, A || B, !A, A ^ B
- 대입 연산자
 - A=B, A+=B, A-=B, A*=B, A/=B, A%=B
- 조건 연산자
 - C=(조건식)? A : B

– 연산자 우선순위

1. ()
2. 단항 연산자(--, ++, !)
3. 산술 연산자(*, /, %, +, -)
4. 비교 연산자(>, >=, <, <=, ==, ===, !=, !==)
5. 논리 연산자(&&, ||)
6. 대입(복합 대입) 연산자(=, +=, -=, *=, /=, %=)

JavaScript 기초 (cont'd.)

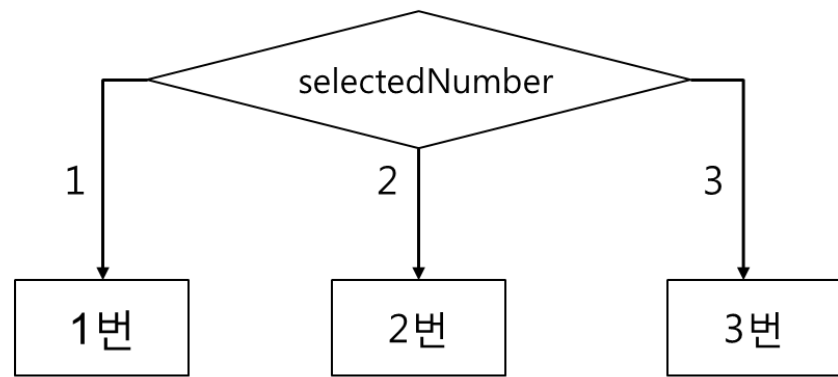
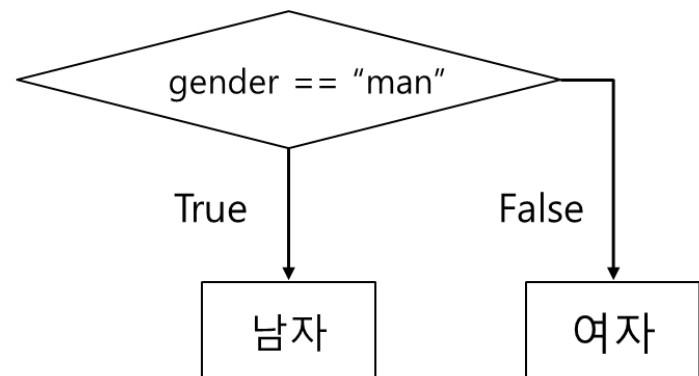
• JavaScript의 문법

– 조건문

```
if ( 조건식 ) { 실행내용 }
```

```
if ( 조건식 )  
{  
    조건식이 참인 경우  
}  
else  
{  
    조건식이 거짓인 경우  
}
```

```
switch( 변수 )  
{  
    case "입력 값 A" : 처리할 내용 1  
        break;  
    case "입력 값 B" : 처리할 내용 1  
        break;  
    case "입력 값 C" : 처리할 내용 1  
        .....  
    default : 기본적으로 처리할 내용  
}
```



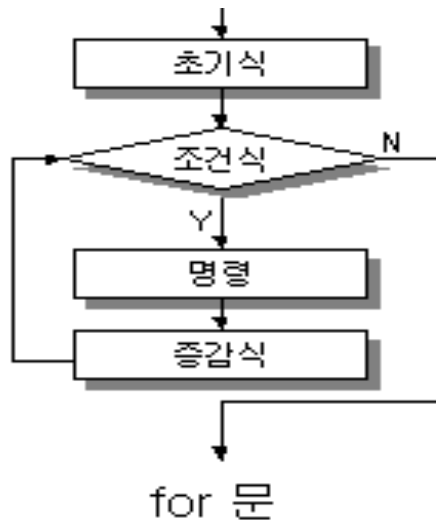
JavaScript 기초 (cont'd.)

• JavaScript의 문법

– 반복문

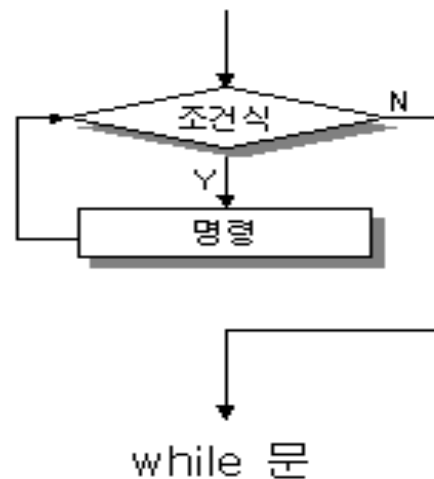
• for문

```
for(초기식; 조건식; 증감식)
{
    반복 내용(명령)
}
```



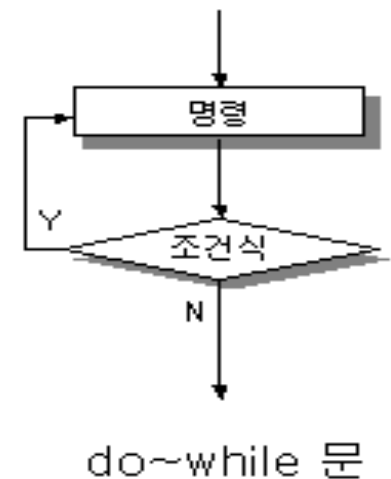
• while문

```
while(조건식)
{
    반복 내용(명령)
}
```



• do~while문

```
do
{
    반복 내용(명령)
} while(조건식)
```



JavaScript 기초 (cont'd.)

• JavaScript의 문법

– 함수(function)

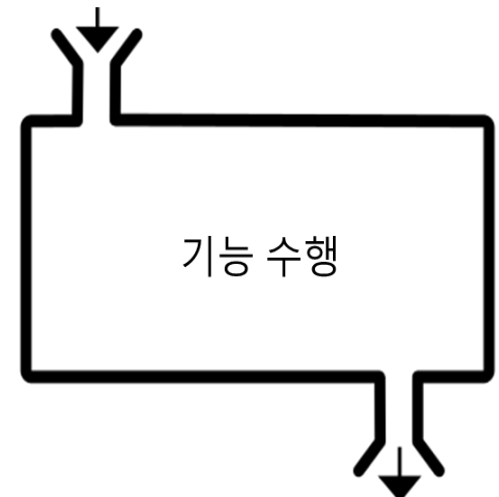
- 가변 자료형을 쓰기 때문에 매개변수와 반환형의 자료형을 명시 안함
- 함수 선언 예
 - 매개변수를 사용한 경우

```
function 함수이름(매개변수 지정, 매개변수 지정)
{
    정의 할 내용
    return 반환 값;
}
```

- 매개변수를 사용하지 않은 경우

```
function 함수이름()
{
    정의 할 내용
    return 반환 값;
}
```

매개 변수



반환 값

JavaScript 기초 (cont'd.)

• JavaScript의 문법

– 이벤트(Event)

- 객체와 상호 작용이 발생할 때 이벤트가 발생함

| 이벤트 유형 | 설명 |
|-------------|-------------------------------------------|
| onload | 문서나 객체의 로딩이 완료되었을 때 발생함 |
| onmouseover | 마우스가 객체 위로 이동할 때 발생 |
| onmousemove | 마우스가 객체 위를 이동할 경우 발생 |
| onmouseout | 마우스가 객체에서 벗어났을 때 발생 |
| onclick | 객체를 클릭하였을 때 발생 |
| onmouseup | 객체를 클릭하고 놓았을 때 발생 |
| ondblclick | 객체를 더블 클릭할 때 발생 |
| onfocus | 객체를 클릭하여 입력 포커스를 얻을 때 발생 |
| onkeypress | 키가 눌렸을 때 발생 (이어서 keydown 이벤트가 발생함) |
| onkeydown | 키가 눌렸을 때 발생 |
| onkeyup | 키에서 손을 뗐을 때 발생 (이어서 keypress 이벤트가 발생함) |
| onchange | 객체가 포커스를 잃고 값이 변경될 때 발생 |
| onsubmit | Form을 제출하였을 때 발생함 |

1) Name:

2) Email address:

3) Comments:

JavaScript 기초 (cont'd.)

• JavaScript의 문법

– 이벤트 핸들러(Event Handler)

- 이벤트를 감지하여 사용자가 정해 놓은 함수를 실행하기 위해 사용함
- 어떤 객체에서 어떤 이벤트가 발생할 때 어떤 일을 할지 설정해야 함

```
특정 객체.특정 이벤트 = function(){
```

```
    //특정 객체에서 특정 이벤트가 발생할 때 함수A를 실행하라는 의미  
    함수A();
```

```
}
```

JavaScript 기초 (cont'd.)

• JavaScript의 문법

– 특정 태그 선택 방법

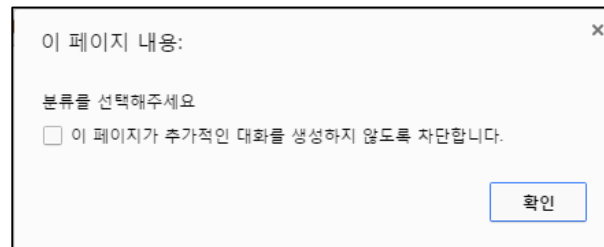
- Form과 Name을 통한 방법 : `document.Form_name.elements[객체_Name]`
- Name을 통한 방법 : `document.getElementsByName(객체_Name)[0]`

– 특정 태그의 속성값 접근 방법

- 속성 값 얻기 : `객체.getAttribute(속성_Name)`
- 속성 값 변경 : `객체.setAttribute(속성_Name, 값)`

– 경고창 띄우는 방법

- `alert('분류를 선택해주세요')`



JavaScript 기초 (cont'd.)

• 예제 2) 이벤트를 통한 기능 부여 (HTML 코드)

- 특정 객체에 접근하기 위해서 Form 태그와 Name 속성을 사용

```
<html>
  <head>
    <title> Javascript 예제</title>
    <script src="ex1.js"></script>
  </head>
  <body>
    <form action="next_page.html" name="test_form">
      <input type="text" name="input1" placeholder="입력상자" />
      <br/><br/>
      <input type="button" name="input2" value="입력 상자의 값을 출력합니다.">
      <br/><br/>
      <input type="button" name="input3" value="1부터 10까지의 값을 모두 더합니다">
      <br/><br/>
      
      <br/><br/>
      <input type="button" name="input5" value="네이버로 이동">
    </form>
  </body>
</html>
```

JavaScript 기초 (cont'd.)

• 예제 2) 이벤트를 통한 기능 부여 (ex1.js)

- 특정 객체에 접근하기 위해서 Form 태그와 Name 속성을 사용

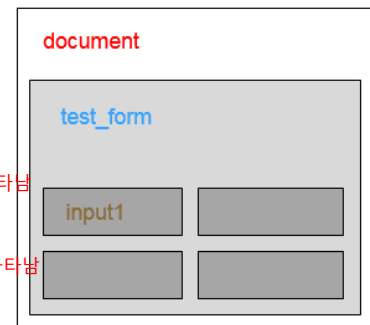
```
window.onload = function () {  
    //이벤트 핸들러 등록  
    document.test_form.input1.onclick = function fct1() {  
        alert('숫자를 입력해주세요');  
    }; //Input1을 클릭하면 숫자를 입력해주세요라는 경고창이 나타남  
    document.test_form.input2.onclick = function fct2() {  
        alert(document.test_form.input1.value);  
    }; //Input2를 클릭하면 input1의 값이 나타남  
    document.test_form.input3.onclick = function fct3() {  
        var temp=10;  
        for(var i=0;i<10;i++)  
        {  
            temp+=i;  
        }  
        alert(temp);  
    }; //Input3을 클릭하면 0부터 9까지의 합이 나타남  
    document.test_form.input4.onmouseover = function fct4_1() {  
        alert("마우스가 이미지 위에 있습니다");  
    }; //Input4에 마우스가 올라가면 마우스가 이미지 위에 있습니다라는 경고창이 나타남  
    document.test_form.input4.onmouseout = function fct4_2() {  
        alert("마우스가 이미지를 벗어 났습니다");  
    }; //Input4에 마우스가 벗어나면 마우스가 이미지를 벗어났습니다라는 경고창이 나타남  
    document.test_form.input5.onclick = function fct5() {  
        location.href = "http://www.naver.com";  
    }; //Input5를 클릭하면 네이버 페이지로 이동됨  
}
```

HTML 코드가 화면에 모두 로드가 되었을때

document.form_name.object_name.attribute

document : 현재문서
form_name : form 태그 이름
object_name : 요소 이름
attribute : 요소의 속성

요소=태그



JavaScript 기초 (cont'd.)

- 예제 2) 이벤트를 통한 기능 부여 (웹페이지)

입력상자

입력 상자의 값을 출력합니다.

1부터 10까지의 값을 모두 더합니다



네이버로 이동

JavaScript 기초 (cont'd.)

• 예제 3) 유효성 검사 (HTML 코드)

- 아이디, 비밀번호, 이메일 등이 정확한지 유효성을 검사해야 함.

```
<html>
  <head>
    <title>Javascript 예제2</title>
    <script src="ex2.js"></script>
  </head>
  <body>
    <form name="registerForm" action="register_ok.asp" method="post">
      <h1>회원가입</h1><br>
      <input type="text" name="id" placeholder="아이디"> 아이디 <br>
      <input type="text" name="pw" placeholder="비밀번호"> 비밀번호 <br>
      <input type="text" name="pw_ok" placeholder="비밀번호 재확인"> 비밀번호 재확인 <br>
      <input type="text" name="name" placeholder="이름"> 이름 <br>
      성별 <input type="radio" name="gender" value="m" checked>남자 <input type="radio" name="gender" value="w">여자<br>
      <input type="text" name="email" placeholder="비상연락용 이메일"> 비상연락용 이메일 <br>
      <input type="text" name="phoneNumber" placeholder="휴대전화번호"> 휴대전화번호 <br>
      <input type="button" name="submitButton" value="가입하기">
    </form>
  </body>
</html>
```


JavaScript 기초 (cont'd.)

• 예제 3) 유효성 검사 (ex2.js)

```
window.onload = function () {  
    document.registerForm.submitButton.onclick = function () {  
        var registerForm = document.registerForm;  
        var id = registerForm.id.value;  
        var pw = registerForm.pw.value;  
        var pw_ok = registerForm.pw_ok.value;  
        var name = registerForm.name.value;  
        var gender = registerForm.gender.value;  
        var email = registerForm.email.value;  
        var phoneNumber = registerForm.phoneNumber.value;  
        var submitButton = registerForm.submitButton.value;  
        var regex; //유효성 검사를 위한 정규표현식  
  
        //가입하기를 눌렀을 때, 입력값이 모두 입력되었는지 유효성 검사  
        if(id=="") {alert("ID를 입력해주세요"); return;}  
        if(pw=="") {alert("비밀번호를 입력해주세요"); return;}  
        if(pw_ok=="") {alert("비밀번호 확인을 입력해주세요"); return;}  
        if(name=="") {alert("이름을 입력해주세요"); return;}  
        if(gender=="") {alert("성별을 선택해주세요"); return;}  
        if(email=="") {alert("이메일을 입력해주세요"); return;}  
        if(phoneNumber=="") {alert("전화번호를 입력해주세요"); return;}  
  
        //비밀번호와 비밀번호확인이 같은지 유효성 검사  
        if(pw!=pw_ok) {alert("비밀번호와 비밀번호 확인이 다릅니다"); return;}
```

JavaScript 기초 (cont'd.)

• 예제 3) 유효성 검사 (ex2.js)

```
//정규표현식을 이용한 유효성 검사
```

```
//이메일 확인 정규표현식
```

```
regex=/^([\w-]+(?:\.[\w-]+)*)@((?![\w-]+\.)*\w[\w-]{0,66})\.([a-z]{2,6}(?:\.[a-z]{2})?)$/;
```

```
if(regex.test(email) === false) {
```

```
    alert("잘못된 이메일입니다.");
```

```
    return;
```

```
}
```

```
//전화번호 형식 확인 (정규 표현식)
```

```
regex=/^(01[016789]{1}|02|0[3-9]{1}[0-9]{1})-?[0-9]{3,4}-?[0-9]{4}$/;
```

```
if(regex.test(phoneNumber) === false) {
```

```
    alert("잘못된 전화번호입니다.");
```

```
    return;
```

```
}
```

```
alert("회원가입 완료");
```

```
}
```

정규표현식 테스트 사이트 : <https://regexr.com/>

JavaScript 기초 (cont'd.)

- 예제 3) 유효성 검사 (웹페이지)

회원가입

| | |
|-----------------------------------------------------------------|-----------|
| 아이디 | 아이디 |
| 비밀번호 | 비밀번호 |
| 비밀번호 재확인 | 비밀번호 재확인 |
| 이름 | 이름 |
| 성별 <input checked="" type="radio"/> 남자 <input type="radio"/> 여자 | |
| 비상연락용 이메일 | 비상연락용 이메일 |
| 휴대전화번호 | 휴대전화번호 |
| 가입하기 | |



THANK YOU



인하대학교