

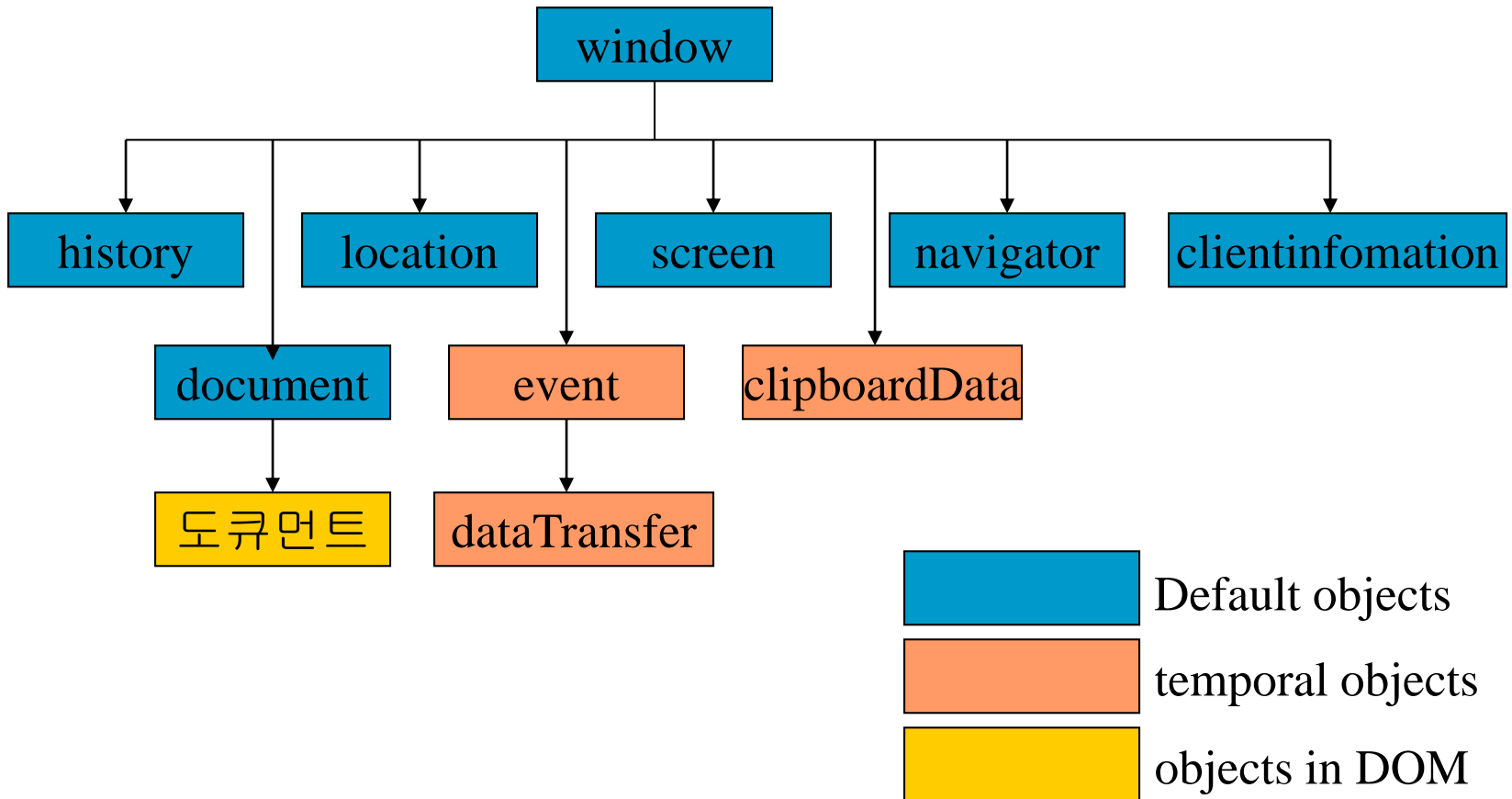
# 9th week



## Javascript Objects(2)

# Objects in DOM

## ✚ Objects hierarchy



# Objects in DOM

## ✚ Definition

- ◆ Every elements in the user html document becomes objects
- ◆ Specially can be identified by id attribute  
`<hr id="hr_1st">`  
...  
`<hr id="hr_2nd">`

## ✚ To access objects

- ◆ Using id , all, children, this, etc
- ◆ `getElementById()`, `getElementsByTagName()`, `getElementsByClassName()`
- ◆ `createElement()`

# Form related objects in DOM

## + Objects

- ◆ form: generated when use `<form>` element
- ◆ textArea: generated when use `<textArea>` element
- ◆ text, password, file, radio, checkbox, button, submit, reset, image: generated when use `<input type="control_type">` element
- ◆ select: generated when use `<select>` element

## + Characteristics

- ◆ These objects except `<form>` have a special property named as 'form'
- ◆ Form property is a reference to `<form>` object which is the parent object



# Practice 1

- ✚ Execute following 6 programs and understand the way to handle DOM tree



```
<!DOCTYPE html >
```

```
<html>
```

```
<head>
```

```
<title>Focus</title>
```

```
<script type="text/javascript">  
    window.onload=setObjects;
```

```
    function setObjects( ) {  
        document.personData.lastName.focus( );  
    }
```

```
</script>
```

```
</head>
```

```
<body>
```

```
    <form name="personData">
```

```
    First Name: <input type="text" name="firstName" /><br /><br />
```

```
    Last Name: <input type="text" name="lastName" />
```

```
    </form>
```

```
</body>
```

```
</html>
```



```
<!DOCTYPE html >
```

```
<html>
```

```
<head>
```

```
<title>Event Handlers and this</title>
```

```
<script type="text/javascript">
```

```
    window.onload=setObjects;
```

```
    function setObjects( ) {
```

```
        document.personData.firstName.onblur=testValue;
```

```
    }
```

```
    function testValue( ) {
```

```
        alert("Hi " + this.value);
```

```
    }
```

```
</script></head>
```

```
<body>
```

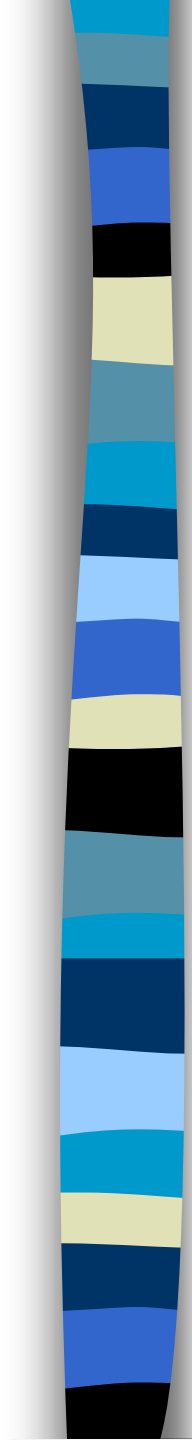
```
<form name="personData">
```

```
First Name: <input type="text" name="firstName" /><br /><br />
```

```
Second Name: <input type="text" name="secondName" />
```

```
</form>
```

```
</body></html>
```



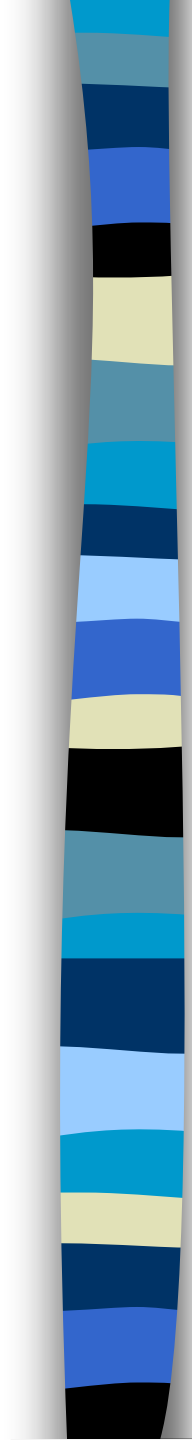
```
<html><head><title>입력한 메일 주소의 형식 확인하기</title>
<script language="javascript">
function chkMail()
{
    var txtMail=document.frm.email.value;
    var result=txtMail.indexOf("@");
    if(result!=-1)

        alert("메일 주소형식이 틀렸습니다");
    else
        alert("메일링에 가입되었습니다")
}
</script></head>
```



[illegible]

```
<script>
function check(text, n)
{
    if(text.value.length != n) {
        alert(n+ " 개의 숫자를 입력하세요!!"); // 경고창 출력
        text.value = ""; // 창의 내용을 지운다.
        return false; // 이벤트 고유 행위를 취소
    }
    else return true; // 이벤트 발생이 유효
}
</script></head><body >
<h3>onchange 핸들러를 이용하여 주민등록 번호를 예제</h3>
<hr noshade>
주민등록 번호 13자리를 다음과 같이 입력하세요. <p>
<form name="f">
<input type="text" id="txt1" name="t" value="" onchange="return
    check(this, 6)"> -
<input type="text" id="txt2" name="t" value="" onchange="return
    check(this, 7)"><br>
</form></body>
```



```
<script>
function compute(f) {
    // f는 fo 객체를 가리킨다.
    var res=eval(f.e.value)
    f.v.value=res
}
</script></head><body >
<h3> onclick 예 : 계산기 만들기</h3><hr noshade>
계산하고자 하는 수식을 입력하고 계산 버튼을 누르시
오.<p>
<form id="fo" name="cal">
식 <input type="text" id="e" name="exp" value=""
size=30><p>
값 <input type="text" id ="v" name="val" value="" size=10>
<input type="button" value=" 계산 "
onclick="compute(this.form)"></form></body>
```

# Form object in DOM

## ✚ Key properties

- ◆ Action, elements, encoding, method
- ◆ Target, length, name

## ✚ Methods

- ◆ reset(), submit()

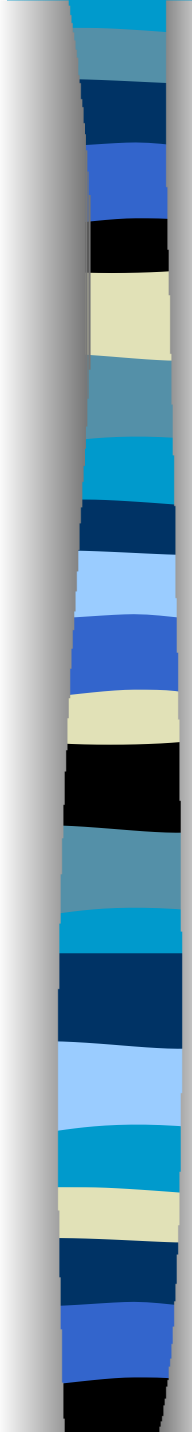
## ✚ Collections

- ◆ All, children, elements



## Practice 2

- ✚ Execute following programs and understand the way to handle form and form related objects



```
<html>
<head><title>예 제</title>
<script>
function printForm(f) {
    var name;
    var index;
    win = window.open(""); for(
    var i=0; i<f.length; i++) {
        name = f.elements[i].name; if(f.elements[i].type == "select-one") {
            index = f.elements[i].selectedIndex;
            value = f.elements[i].options[index].value;
        } else value = f.elements[i].value; win.document.
        writeln(name + " = " + value + "<br>");
    }
    win.document.title="예 제 8-1";
    win.document.close();
    return false; // do not submit
}
</script>
</head>
```

<body>

<h3>폼 요소들과 객체와의 관계</h3>

<hr noshade>

폼의 내용을 입력한 후 O.K 버튼을 누르시오

<p>

<form method="POST" onsubmit="return printForm(this)">

이름 <input type="text" name="username" value="kitae"><br>

암호 <input type="password" name="passwd"><br>

직업 <input type="radio" name="job" value="student">student

<input type="radio" name="job" value="professor" checked>professor

<input type="radio" name="job" value="lawyer">lawyer<br>

좋아하는 수업

<input type="checkbox" name="multi" checked>multimedia

<input type="checkbox" name="arch">architecture<br>

파일을 선택하시오 <input type="file" name="file"><br>

과일을 선택하시오

<select name="fruits">

<option value="1">apple

<option value="2">banana

</select>

<input type="submit" name="sub" value="O.K">

</form></body></html>

# Select and option objects in DOM

## ✚ Key properties of select object

- ◆ type
- ◆ length
- ◆ options
- ◆ selectedIndex

## ✚ Key properties of option object

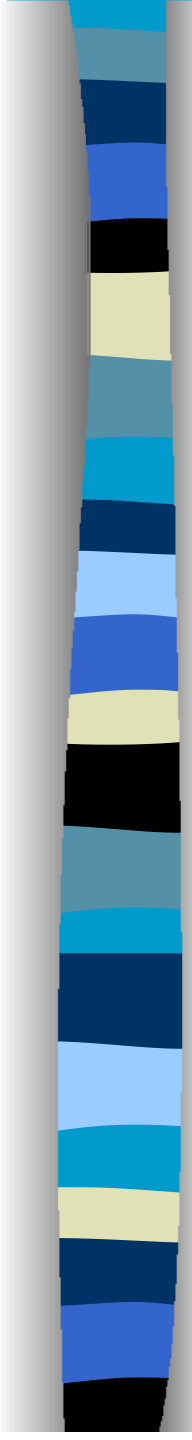
- ◆ form
- ◆ text, value
- ◆ defaultSelected
- ◆ selected



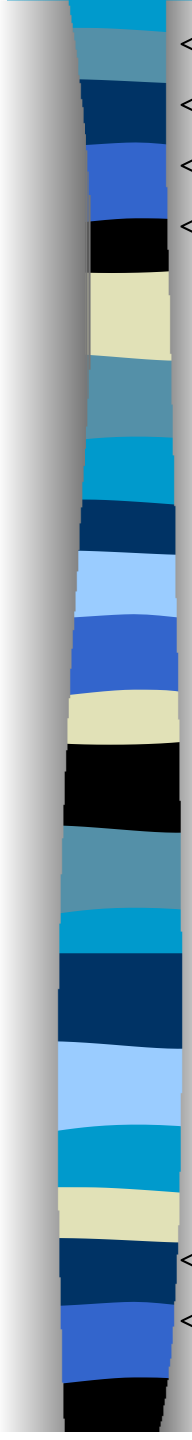


# Practice 3

- ✚ Execute following programs and understand the way to select and option objects



```
<HTML>
<HEAD>
<TITLE> New Document </TITLE>
<SCRIPT LANGUAGE="JavaScript">
    function fnChange(){
        oData.value+=oCars.options[oCars.selectedIndex].text + "\t"+
            +oCars.options[oCars.selectedIndex].value + "\n";
    }
</SCRIPT>
</HEAD>
<BODY>
<SELECT ID="oCars" SIZE="1" onchange="fnChange()">
    <OPTION VALUE="1">BMW
    <OPTION VALUE="2">PORSCHE
    <OPTION VALUE="3" SELECTED>MERCEDES
</SELECT>
<TEXTAREA ID="oData"></TEXTAREA>
</BODY>
</HTML>
```



```
<html><head>
<title>Input form</title>
<meta http-equiv="Content-Type" content="text/html; charset=EUC-KR" />
<script type="text/javascript">
    window.onload=setupEvents;
    function setupEvents(evnt) {
        document.someForm.selectOpts.onChange = checkSelect;
    }
    function checkSelect(evnt) {
        var opts = document.someForm.selectOpts.options;
        for (var i = 0; i < opts.length; i++) {
            if (opts[i].selected) {
                switch(opts[i].value) {
                    case "Opt1" : opts[i + 1].selected = true; break;
                    case "Opt3" : opts[i + 1].selected = true; break;
                    case "Opt4" : opts[i + 1].selected = true; break;
                }
            }
        }
    }
</script>
</head>
```



```
<BODY>
```

```
<select id="oSelect" name="Cars" size="3" multiple>
```

```
<option value="1" selected>BMW
```

```
<option value="2">Porsche
```

```
<option value="3" selected>Mercedes
```

```
</select>
```

```
<script language="JavaScript">
```

```
var oOption = document.createElement("OPTION");
```

```
oOption.text="Ferrari";
```

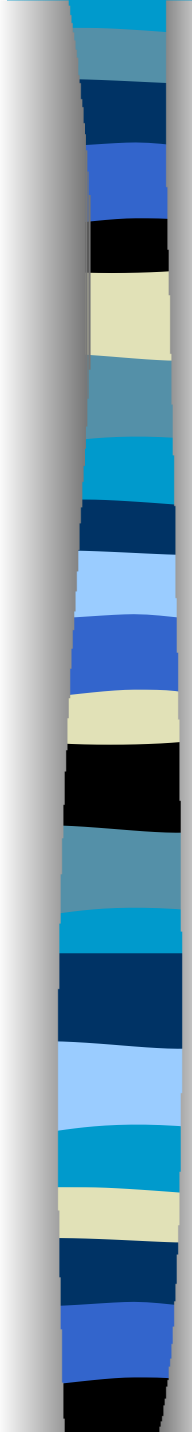
```
oOption.value="4";
```

```
oSelect.add(oOption);
```

```
</script>
```

```
</BODY>
```

```
</HTML>
```



```
<html>
<head><title>양식간 내용 복사</title></head>
<body>
<h3>버튼을 클릭하면 왼쪽 창에 입력된 내용을
오른쪽 창으로 복사합니다.</h3>
<hr noshade>
<form name="f">
<input type="text" name="t1" size="10">
<input type="button" value=" -> " onclick="this.form.t2.
    value=this.form.t1.value">
<input type="text" name="t2" size="10">
</form>
</body>
```



# Exercise 1

- ✚ Upgrade your own registration page submitted for Exercise 1 in 5<sup>th</sup> week material using JavaScript.
  - ◆ When you input any data, the data can be validated
  - ◆ Use event attributes or properties and define handler function written in JavaScript
- ✚ Write down your own program and compare it with the given program in next slide



```
<!DOCTYPE html>
```

```
<html><head><meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
```

```
<title> 회원 가입</title>
```

```
<link rel="stylesheet" type="text/css" href="가입.css" />
```

```
<script>
```

```
    function domainCheck(){
```

```
        var frm=document.fom;
```

```
        if(frm.mail_domain.value==0){
```

```
            frm.mail_main.value="";
```

```
            frm.mail_main.disabled=false;
```

```
        }else{
```

```
            frm.mail_main.value=frm.mail_domain.value;
```

```
            frm.mail_main.disabled=true;
```

```
        }
```

```
    }
```

```
    function pdcheck(){
```

```
        var orgin = document.getElementById("pd")
```

```
        var now = document.getElementById("pd_check")
```

```
        if((orgin.value != "") && (now.value != "")){
```

```
            var pd = document.getElementById("pd_ok")
```

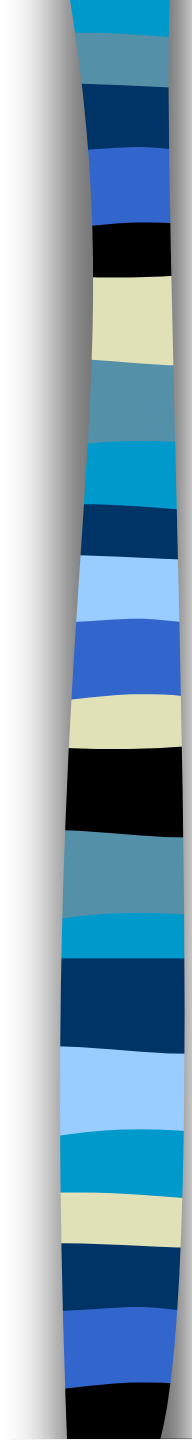
```
            if(orgin.value == now.value)
```

```
                pd.value = "true"
```

```
            else
```

```
                pd.value = "false"
```

```
    }}
```



```
window.onload=start
function start(){
    document.fom.id.focus();
}
</script> </head>
<body>
<form name = "fom">
    <section class = "left">
        아이디:&nbsp;                <p>
        비밀번호:&nbsp;                <p>
        비밀번호확인:&nbsp;            <p>
        이름:&nbsp;                    <p>
        성별:&nbsp;                    <p>
        핸드폰번호:&nbsp;              <p>
        메일:&nbsp;                    <p>
        <input type= "submit" class= "mi" value = "가입">&nbsp;
    </section>
    <section class = "right">
        <input type="text" id = "id" name="id" placeholder = "아이디"><p>
        <input type="password" id = "pd" placeholder="비밀번호"><p>
        <input type="password" id = "pd_check" placeholder="비밀번호확인"
            onblur = "pdcheck()">
```





```
<input type="hidden" id = "pd_ok" value = "false"><p>
<input type="text" id= "name" name="name" placeholder="0 이름"><p>
남<input type="radio" name= "sex" style = "width: 10px;">
여<input type="radio" name= "sex" style = "width: 10px;"><p>
<select name = "ph">
    <option value="010">010</option>
    <option value="011">011</option>
    <option value="016">016</option>
</select> -
<input type = "text" name = "hp1" maxlength = 4 style="width:60px"> -
<input type = "text" name = "hp2" maxlength = 4 style="width:60px"><p>
<input type="text" name="mail_id" style="width:80px" > @
<input type="text" name="mail_main" style="width:80px">
<select name="mail_domain" onchange="domainCheck()">
    <option value = "0" selected="selected">직접입력</option>
    <option value = "naver.com" >naver.com</option>
    <option value = "nate.com">nate.com</option>
</select><p>
<input type= "reset" class= "mi" value = "초기화" >
<input type= "button" class= "mi" value = "취소"
                                onclick="window.close()">
```

```
</section>
```

```
</form>
```

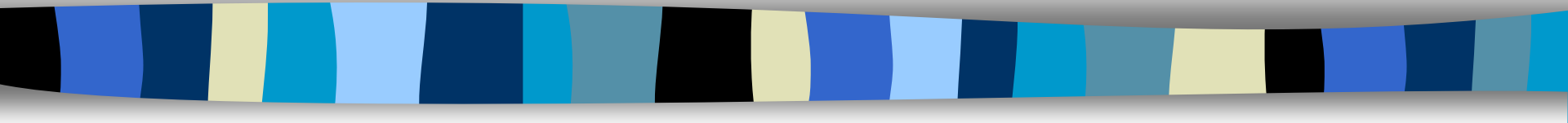
```
</body></html>
```

# 가입.css

```
body{
  background-color:#D5D5D5;
}
input {
  width: 150px;
  height: 11px;
}
.bu{
  width: 100px;
  height: 19px;
}
.mi{
  width: 100px;
  height: 20px;
}
.main{
  max-width:600px;
  height: 430px;
}
```

```
.left{
  width:30%;
  height: 100%;
  float:left;
  text-align:right;
}
.right{
  width:70%;
  height: 100%;
  float:left;
}
```

# 10th week



## Javascript API

# Web storage

## ✚ Definition

- ◆ Data are needed to be stored/used in several web pages within the same domain
- ◆ Before HTML5, cookie or session was used
- ◆ Web storage is supported for storing data in the client in HTML5

## ✚ Cookie vs. Web storage

- Cookie is transferred being embedded with each request and response
- The size of cookie is restricted to 4K
- Larger storage is supported
- Transferred to server when needed

# Web storage

## ✚ Interface([www.w3.org/TR/webstorage/](http://www.w3.org/TR/webstorage/))

```
interface Storage {  
    readonly attribute unsigned long length;  
    DOMString? key(unsigned long index);  
    getter DOMString getItem(DOMString key);  
    setter creator void setItem(DOMString key, DOMString value);  
    deleter void removeItem(DOMString key);  
    void clear();  
};
```

## ✚ Objects supported

- ◆ localStorage: data is stored after a window is closed
- ◆ sessionStorage:
  - Data is stored together with window objects
  - Don't know the value of other window/tabs
  - Data is deleted when window/tab is closed

# Practice 4

- ✚ Do you remember the exercise 2 program in 7<sup>th</sup>? Is there any problem to use?
  - ◆ After going out the program, what is happen to your songs list?
  - ◆ When you connect this program again, are there new songs added? → no
  - ◆ One solution is localStorage
  - ◆ Whenever you add a song, it is stored to local storage. And when you connect the program, every store songs are added automatically
- ✚ Compare the previous program with the program in the next slide



```
<head>
```

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
```

```
<title>음악 추가 시키기</title>
```

```
<script>
```

```
    window.onload= start; //이전 데이터 가져오기
```

```
    function start(){
```

```
        for(var i = 0; i < localStorage.length; i++){//로컬스토리지 길이만큼 반복
```

```
            var key = localStorage.key(i);
```

```
            var li = document.createElement("li");
```

```
            li.innerHTML = localStorage[key];
```

```
            var ul = document.getElementById("playlist");
```

```
            ul.appendChild(li);
```

```
        }
```

```
    }
```

```
    function addSong(){
```

```
        var input= document.getElementById("songName");
```

```
        var add= document.createElement("li");
```

```
        add.innerHTML=input.value;
```

```
        var ul= document.getElementById("playlist");
```

```
        ul.appendChild(add);
```

```
        var str = "list"+localStorage.length;
```

```
        localStorage.setItem(str,input.value);
```

```
    }
```

```
</script>
```

```
</head>
```



# Practice5

- ✚ The addsong program looks like good but it has a big problem?
- ✚ Do you know the problem?
- ✚ One solution to use array
- ✚ Try to solve!!!





```
<head>
```

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
```

```
<title>음악 추가 시키기</title>
```

```
<script>
```

```
    window.onload= start;    //이전 데이터 가져오기
```

```
    function start(){
```

```
        var value = localStorage.getItem('localSonglist');
```

```
        if(value!=null){
```

```
            var storedStorage=JSON.parse(value);
```

```
            for(var i=0; i<storedStorage.length; i++){
```

```
                var li = document.createElement("li");
```

```
                li.innerHTML = storedStorage[i];
```

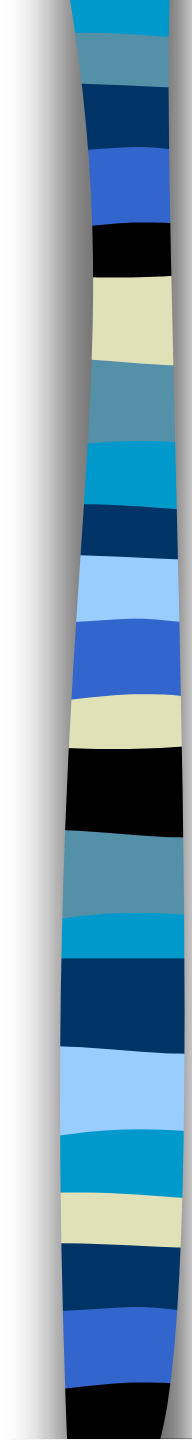
```
                var ol = document.getElementById("playlist");
```

```
                ol.appendChild(li);
```

```
            }
```

```
        }
```

```
    }
```



```
var Songlist = new Array();
function AddSong() {
    var input= document.getElementById("songName");
    if(input.value!=""){
        var li = document.createElement("li");
        li.innerHTML = input.value;
        var ol = document.getElementById("playlist");
        ol.appendChild(li);
        Songlist.push(input.value);
        localStorage.setItem("localSonglist", JSON.stringify(Songlist));
    }
}
</script> </head>
```



## Exercise 2

- ✚ Sticker program is very useful.
- ✚ Write the sticker program
  - ◆ Can add to-do stickers
  - ◆ Can delete to-do sticker when it is finished



```
<title>What to do</title>
```

```
<link rel="stylesheet" href="middle1.css" type="text/css">
```

```
<script>
```

```
    window.onload= start;
```

```
    function start(){
```

```
        for(var i = 0; i < localStorage.length; i++){
```

```
            var key = localStorage.key(i);
```

```
            var li = document.createElement("li");
```

```
            li.innerHTML = localStorage[key];
```

```
            var ul = document.getElementById("list");
```

```
            ul.appendChild(li);
```

```
        }
```

```
    }
```

```
    function addlist(){
```

```
        var input= document.getElementById("name");
```

```
        var add= document.createElement("li");
```

```
        add.innerHTML=input.value;
```

```
        var ul= document.getElementById("list");
```

```
        ul.appendChild(add);
```

```
        var str = "list"+localStorage.length;
```

```
        localStorage.setItem(str,input.value);
```

```
    }
```

```
</script>
```

```
<body>
```



```
<form>
```

```
<input type="text" id = "name">
```

```
<input type="button" id = "addbutton" value ="스티커 노트 추가"  
onclick="addlist()">
```

```
<p>
```

```
<ul id= "list">
```

```
<li>드라이클리닝 찾아오기</li>
```

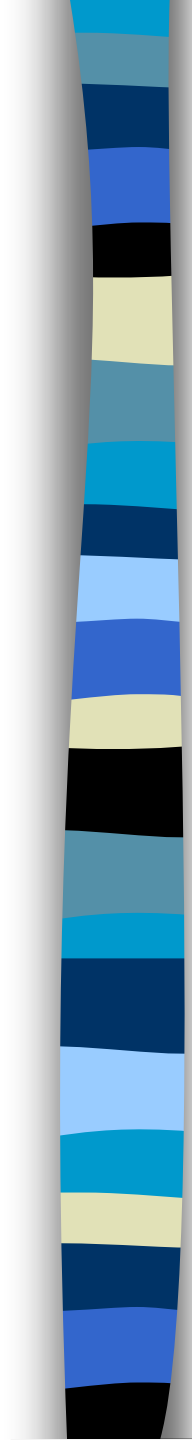
```
<li>케이블tv 취소하기</li>
```

```
</ul>
```

```
</form>
```

```
</body>
```

```
</html>
```



```
*{
  margin : 0.5em;
  padding : 0;
}
BODY {
  background : #acacac;    /*배경 회색*/
}
```

```
#name{
  width : 800px;
  height : 30px;
}
```

```
#addbutton{
  width : 150px;
  height : 30px;
  font-size : 15px;
  font-family: sans-serif;
}
```

```
#list ul{
  width : 1000px;
}
```

```
  #list li{
    background : #ffffd9;    /*배경 노란색*/
    width : 400px;
    height : 300px;
    list-style: none;        /*li의 앞에 *제거*/
    float: left;
    font-size : 30px;
  }
```



# Web worker

- ✚ JavaScript behavior
  - ◆ Single thread
  - ◆ Process several request/response sequentially
  - ◆ While processing request, a long time procedure would make the interaction with users longer
  - ◆ Slow script problem
- ◆ Add multiple thread creation in HTML5: Web worker



# Practice6

```
<!DOCTYPE HTML>
<html>
<head>
<title>Worker example: One-core computation</title>
</head>
<body>
  <p>The highest prime number discovered so far is:
    <output id="result"></output>
  </p>
  <script>
    var worker = new Worker('worker.js');
    worker.onmessage = function (event) {
      document.getElementById('result').textContent = event.data;
    };
  </script>
</body> </html>
```





## worker.js

```
var n = 1;
search: while (true) {
    n += 1;
    for (var i = 2; i <= Math.sqrt(n); i += 1)
        if (n % i == 0) continue search;
    // found a prime!
    postMessage(n);
}
```



## Exercise 3

- ✚ Upgrade your program written in Exercise 1 of 9<sup>th</sup> using web worker

# Ajax

- ✚ Asynchronous Javascript and XML
- ✚ Based on Javascript and DOM
- ✚ New approach for server processing
- ✚ Asynchronous processing in the same page
- ✚ Use the XMLHttpRequest or XMLHttpRequest object instead of form submit



구매자 수	1012명
단가	250원
비용	80원

이익금 : **172,040**

판매수익 현황



# AJAX-programming

## + Step 1

- ◆ Make the object to submit a request– request object
- ◆ Use the built in XMLHttpRequest or XMLHttpRequest class

## + Step 2

- ◆ Write JavaScript function to set server connection and callback function
- ◆ Use the request object made by step 1

## + Step 3

- ◆ Write the callback function to process the server response

# Practice7

```
<!DOCTYPE html >
<html><head><title>Hello Ajax World</title>
<meta http-equiv="Content-Type" content="text/html; charset=EUC-KR" />
<style type="text/css">
div.elem { margin: 20px; }
</style>
<script type="text/javascript">
var xmlhttp = false;
if (window.XMLHttpRequest) {
    xmlhttp = new XMLHttpRequest( );
    xmlhttp.overrideMimeType('text/xml');
} else if (window.ActiveXObject) {
    xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");
}
function populateList( ) {
    var state = document.forms[0].elements[0].value;
    var url = 'ajax.jsp?state=' + state;
    xmlhttp.open('GET', url, true);
    xmlhttp.onreadystatechange = getCities;
    xmlhttp.send(null);
}
```



## Exercise 4

- ✚ How can we use post methods for request submit?



# For post method

```
function populateList( ) {  
    var state = document.forms[0].elements[0].value;  
    var qry = "state="+ state;  
    var url = 'ajax.jsp?';  
    xmlhttp.open( 'POST', url, true);  
    xmlhttp.onreadystatechange = getCities;  
    xmlhttp.setRequestHeader("Content-type", "application/x-www-form-  
        urlencoded");  
    xmlhttp.send(qry);  
}
```