

Лекц 11

AJAX, JSON

Object & AJAX & JSON

Объект

- Хувьсагч агуулсан хувьсагч

JavaScript объект үүсгэх

Объект литерал ашиглах

```
var person = {firstName:"John", lastName:"Doe", age:50, eyeColor:"blue"};
```

JavaScript объект үүсгэх

new түлхүүр үг ашиглах

```
var person = new Object();  
person.firstName = "John";  
person.lastName = "Doe";  
person.age = 50;  
person.eyeColor = "blue";
```

JavaScript объект үүсгэх

Байгуулагч функц ашиглах

```
function person(first, last, age, eyecolor) {  
    this.firstName = first;  
    this.lastName = last;  
    this.age = age;  
    this.eyeColor = eyecolor;  
}  
  
var myFather = new person("John", "Doe", 50, "blue");  
var myMother = new person("Sally", "Rally", 48, "green");
```

The this Keyword

- In JavaScript, the thing called this, is the object that "owns" the JavaScript code.
- The value of this, when used in a function, is the object that "owns" the function.
- The value of this, when used in an object, is the object itself.
- The this keyword in an object constructor does not have a value. It is only a substitute for the new object.
- The value of this will become the new object when the constructor is used to create an object.
- Note that this is not a variable. It is a keyword. You cannot change the value of this.

Бэлэн байгуулагч функцүүд

- `var x1 = new Object();` // Шинэ Object объект
- `var x2 = new String();` // Шинэ String объект
- `var x3 = new Number();` // Шинэ Number объект
- `var x4 = new Boolean();` // Шинэ Boolean объект
- `var x5 = new Array();` // Шинэ Array объект
- `var x6 = new RegExp();` // Шинэ RegExp объект
- `var x7 = new Function();` // Шинэ Function объект
- `var x8 = new Date();` // Шинэ Date объект

Зaalт

```
var person = {firstName:"John", lastName:"Doe", age:50, eyeColor:"blue"}
```

```
var x = person;
```

```
x.age = 10;
```

```
// x.age болон person.age хоёулаа өөрчлөгдөнө
```

Шинж чанар /гишүүн өгөгдөл/

- Объекттой холбоотой утгуудыг агуулна.
- Эрэмбэлэгдээгүй
- Ихэвчлэн өөрчлөгдсөн, нэмэгдсэн, устгагдсан төлөвт байх ба заримдаа зөвхөн унших төлөвт байна.

Шинж чанарт хандах

objectName.property // person.age

objectName["property"] // person["age"]

objectName[expression] // x = "age"; person[x]

for...in давталт

```
for (хувьсагч in объект) {
```

```
    Ажиллуулах код
```

```
}
```

```
var person = {fname:"John", lname:"Doe", age:25};
```

```
for (x in person) {
```

```
    txt += person[x];
```

```
}
```

Шинж чанар нэмэх

- Шууд утга олгох замаар нэмнэ.
- `person.nationality = "English";`

Шинж чанар устгах

- **delete** түлхүүр үг ашиглана:
- **delete** person.age;
- **// эсвэл delete person["age"];**

Арга /үйлдэл/

- Дараах бичиглэлийг баримтална
- үйлдлийнНэр : `function()` { кодын мөрүүд }
- Объектын үйлдэлд хандах:
- `объектынНэр.үйлдлийнНэр()`

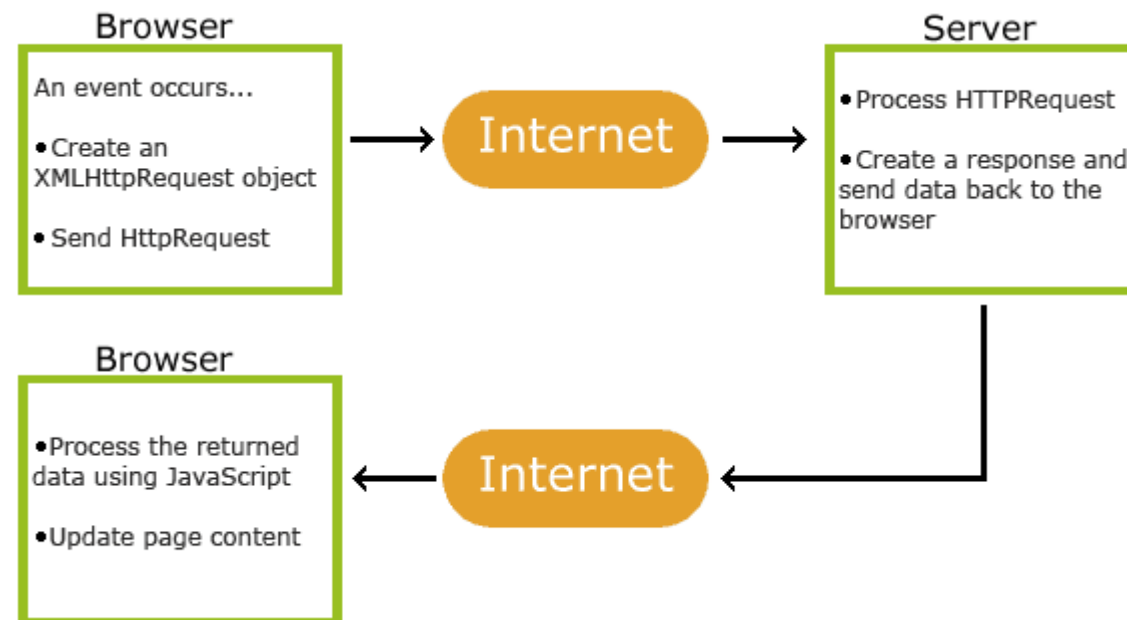
ҮЙЛДЭЛ НЭМЭХ ЖИШЭЭ

```
function person(firstname, lastname, age, eyecolor) {  
  this.firstname = firstname;  
  this.lastname = lastname;  
  this.age = age;  
  this.eyecolor = eyecolor;  
  this.changeName = function (name) {  
    this.lastname = name;  
  }  
}  
myMother.changeName("Doe");
```


AJAX гэж юу вэ?

- AJAX = **A**synchronous **J**avaScript and **X**ML.
- Уян хатан, хурдан ажиллагаатай вэб хийх арга барил
- Вэб хуудсыг бүхэл д биш, тодорхой жижиг хэсгийн мэдээллийг шинэчлэн ачаална

AJAX-ын ажиллагаа



```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<div id="myDiv"><h2>Let AJAX change this  
text</h2></div>
```

```
<button type="button"  
onclick="loadXMLDoc()">Change Content</button>
```

```
</body>
```

```
</html>
```

```
<head>
```

```
<script>
```

```
function loadXMLDoc()
```

```
{
```

```
.... AJAX script энд байна ...
```

```
}
```

```
</script>
```

```
</head>
```

XMLHttpRequest объект

```
var xmlhttp;  
if (window.XMLHttpRequest)  
    {  
        // code for IE7+, Firefox, Chrome, Opera, Safari  
        xmlhttp=new XMLHttpRequest();  
    }  
else  
    {  
        // code for IE6, IE5  
        xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");  
    }
```

Үйлдэл

- **open(method,url,async)** - Specifies the type of request, the URL, and if the request should be handled asynchronously or not.
 - **method**: хүсэлтийн төрөл: **GET** эсвэл **POST**
 - **url**: сервер дээрх файлын байршил
 - **async**: **true** (зэрэгцээ) эсвэл **false** (цуваа)
- **send(string)** хүсэлтийг сервер лүү илгээнэ.
 - **string**: Зөвхөн **POST** хүсэлтэд ашиглана

GET хүсэлтийн жишээ

Энгийн хүсэлт:

```
xmlhttp.open("GET","demo_get.asp",true);
```

```
xmlhttp.send();
```

Утга дамжуулалт:

```
xmlhttp.open("GET","demo_get.asp?t=" + Math.random(),true);
```

```
xmlhttp.send();
```

POST хүсэлт

Энгийн хүсэлт:

```
xmlhttp.open("POST","demo_post.asp",true);  
xmlhttp.send();
```

Утга дамжуулалт:

```
xmlhttp.open("POST","ajax_test.asp",true);  
xmlhttp.setRequestHeader("Content-type","application/x-www-form-urlencoded");  
xmlhttp.send("fname=Henry&lname=Ford");
```

НЭМЭЛТ ҮЙЛДЭЛ

- `setRequestHeader(header,value)` – хүсэлтийн HTTP header-ийг өөрчилнө.
 - **header**: header-ийн нэр
 - **value**: header-ийн утга

Async=true

```
xmlhttp.onreadystatechange = function()  
{  
  if (xmlhttp.readyState == 4 && xmlhttp.status == 200)  
  {  
    document.getElementById("myDiv").innerHTML = xmlhttp.responseText;  
  }  
}  
xmlhttp.open("GET","ajax_info.txt",true);  
xmlhttp.send();
```

Async=false

```
xmlhttp.open("GET","ajax_info.txt",false);
```

```
xmlhttp.send();
```

```
document.getElementById("myDiv").innerHTML = xmlhttp.responseText;
```

AJAX – Серверийн хариу

- **responseText** – хүсэлтийн үр дүнг тэмдэгт мөрөөр авна
- `document.getElementById("myDiv").innerHTML=xmlhttp.responseText;`
- **responseXML** – хүсэлтийн үр дүнг XML-ээр авна
- `xmlDoc = xmlhttp.responseXML;`
- `txt = "";`
- `x = xmlDoc.getElementsByTagName("ARTIST");`
- `for (i = 0; i < x.length; i++)`
- `{`
- `txt = txt + x[i].childNodes[0].nodeValue + "
;`
- `}`
- `document.getElementById("myDiv").innerHTML = txt;`

-
- **onreadystatechange** - Stores a function (or the name of a function) to be called automatically each time the readyState property changes
 - **readyState** - XMLHttpRequest-ийн төлвийг хадгална. 0-с 4 хүртэл утга авна
 - 0: хүсэлт ачаалагдаагүй
 - 1: сервертэй холбогдсон
 - 2: хүсэлтийг хүлээж авсан
 - 3: хүсэлтийг боловсруулж байна
 - 4: хүсэлт дуусч хариу бэлэн болсон
 - **Status**
 - 200: "OK"
 - 404: Page not found

JSON гэж юу вэ?

- JavaScript **O**bject **N**otation гэсэн үг
- Өгөгдөл солилцох авсаархан хэвжүүлэлт
- Хэл харгалзахгүй
- "өөрийгөө-тодорхойлсон" ойлгоход хялбар

JSON Жишээ

```
{"employees":[  
  {"firstName":"John", "lastName":"Doe"},  
  {"firstName":"Anna", "lastName":"Smith"},  
  {"firstName":"Peter", "lastName":"Jones"}  
]}
```

JSON бичиглэл

- Өгөгдөл нь **нэр/утга** нийлэмжээс тогтоно
- Өгөгдөл нь **таслалаар** тусгаарлагдана
- **Угалзан** хаалтаар **объектыг** дүрсэлнэ
- **Дөрвөлжин** хаалтаар **хүснэгт** буюу массивийг дүрсэлнэ

JSON бичвэрийг JavaScript объект руу хөрвүүлэх

```
var text = '{ "employees" : [' +  
  '{ "firstName":"John" , "lastName":"Doe" },' +  
  '{ "firstName":"Anna" , "lastName":"Smith" },' +  
  '{ "firstName":"Peter" , "lastName":"Jones" } ]}';  
  
var obj = JSON.parse(text);  
  
text = JSON.stringify(obj);
```

Анхаарал тавьсанд баярлалаа!

Асуулт байна уу ?