JavaScript

Whats it

- Interpreted lang with OO capabilities
- Syntax, resembles to C
- Most commonly used in browser, client side JavaScript, but actually its general language
- JavaScrips <> Java
- Before the name : LiveScript
- One of WWW Recommendation (please be http://w3.org and http://w3school 's lover)

- Was designed to add interactivity to HTML pages
- Is a scripting language (a scripting language is a lightweight programming language)
- JavaScript code is usually embedded directly into HTML pages
- JavaScript is an interpreted language (means that scripts execute without preliminary compilation)

What can it does?

- Gives HTML designers a programming tool
- Put dynamic text into an HTML page
- React to events
- Read and write HTML elements
- Be used to validate input data
- Be used to detect the visitor's browser
- Be used to create cookies

In advanced of JS

- to add functionality on webpage, ~~validate forms, communicate with the server~~, etc
- JQuery. Lib of JS
- AJAX. Asynchronous JavaScript and XML
- JSON. JavaScript Object Notation → will be quite important

Some suggestions

- Just like the other languages, Js has built in function, along with practice it will be easier to understand and Look Up Again the Compl Ref
- U can put it anywhere in Ur page
- Lack in security, so U know U wanna use Js where and for what
- Not full featured in OO, dont use it for large development project

Place

- <script> … </script> in <head></head>

  or

- <body> … </body>
- Sama seperti css, dapat diambil dari remote file

```
<head>
<script language="JavaScript" type="text/javascript"
        src="http://site1/JavaScript1.js">
</script>
<script language="JavaScript"type="text/javascript"
        src="myJS/myJavaScript2.js">
</script>
```

```
<body>
<script>
var fact=1;
for (i=1;i<10;i++){
fact=fact*i;
document.write(i+"!= "+fact+"<br>");
}
</script>
</body>
```

```
1!= 1
2!= 2
3!= 6
4!= 24
5!= 120
6!= 720
7!= 5040
8!= 40320
9!= 362880
```

Variables

- is name associated with value
- Create a variable with or without the ”var” statement

  var strname = some value

  strname = some value
- Before use it, declar it.
  - var a; var msg;
  - var a=7; var msg=”hi”;

- Scope, the region of your program in which it is defined
  - Global, its defined everywhere in your Js code
  - Local, defined only in the body of function
- The lifetime of these variables starts when they are declared, and ends when the page is closed

```
var scope="its global";     //global
function check(){
     var scope ="its local";     //local
     function check2(){
          var scope="its local nested";
//nested local
          document.write(scope);       //print
nested local
     }
     Check2();
}
check();
```

Identifier

- Naming variables and function
  - Ex: a, $a, a_b, _a
- Case sensitif
- Also have a lot reserved words
  - Ex: break, case, continue, do, else, if, this, true, null, ...etc

Datatypes

- Numbers
  - Integer: 0, 20, 10000000
  - Hexadecimal: 0xff
  - Octal: 0377
  - Floating: 1.1, .7, 1.7e20, 1.7E-20
- String
  - Literal: ”this”, 'this'

- Boolean
  - True, false
- Function
  - A piece executable code that defined by Js or predefined by Js
  - Function circle()
- Object
  - Collection of name value, usually referred to properties of object

- var a = new Object()
- image.width
- document.write
- document.myform.

- Array
  - A collection of data values
  - var a = new array()
  - a[0] = 7;a[1]='seven';

Function

- a block in Js code that is defined once but may be invoked or execute any number of times
- Have parameter or arguments → local var, specified when the function is invoked, usually to compute a return value that become the value of function
- When invoked on an object, function is called method
- Familiar with concept procedure/ subroutine

```
function BoxVolume(l,w,h){
	var V;
	V=l*w*h;
	document.write(V);
}
BoxVolume(3,3,3);
```

```
function distance(x1,x2,y1,y2){
	var dx=x2-x1;
	var dy=y2-y1;
	return Math.sqrt((dx*dx)+(dy*dy));
}
```

```
function factorial(x){
if (x<=1)
return 1;
return x*factorial(x-1);
}
document.write(factorial(7));
```

Return, turn to exist parameter

- A JavaScript function contains some code that will be executed only by an event or by a call to that function
- You may call a function from anywhere within the page (or even from other pages if the function is embedded in an external .js file).
- Functions can be defined either <head> or <body> section. Umumnya di <head> section

- Nested

```
function hypo(a,b){
      function square(x){return x*x}
      return Math.sqrt(square(a)+square(b));
}
```

- Function literal → function definition

```
function square(x){return x*x} // f.statement
var square = function(x){return x*x;}; //f.literal
```

- Function as Data, mean can be assigned to variables

```
function square(x){return x*x;}
var a = square(5); // w'll have 16
var b = square; // b refers to the same function that square does
var c = b(7); // w'll have 49
```

- As property of object (method)

```
var o = new Object;
o.square = function(x){return x*x;} //as object property, f.literal
y = o.square(7); // 49
```

Object

- Is collection of named values (property)

```
var x = new Object();
var d = new Date();
```

- Object literal → object definition (make new obj and its properties)

```
var point = {x=1,y=-1};
```

- All objects in Js inherit from class object
- Every object has constructor property that refers to the constructor function

```
var dt = new Date();
dt.constructor == Date;
```

Class, Constructor

- Class of object, individual object as instance of class. Class in Js just like pseudo class, not like in Java or C++. Its constructor or prototype
- A function designed to be used with new operator is called a constructor.
- Constructur use to initialize a newly created object, setting any properties that need to be set before the object is used
- Constructor usually dont have return value

```
//initiate the constructor
function Rectangle(w,h){
this.width = w;
this.height = h;
//note, no return value
}
```

```
//new objects
var rect1 = new Rectangle(1,7);
var rect2 = new Rectangle(7,17);
```

Array

- is an ordered collection of values
- Value contains index, element

```
var x = [] // var x = new Array();
var d = [”d”,”w”,”w”];
dar d = [”d”,”w”,”0”];
```

- A lot keyword function to manage array, ex: join(), reverse(), concat(), slice(), etc
- Tidak support multidimensional array, but dapat dibuat array of arrays → alike objects

Operator

- Arithmatic: +,-,*,/,%,++,--
- Equality: ==, ===, !=, !==
- Comparison: <,>,<=,>=, ”in”
- String : +
- Logical: and/&&, or/||, not/!

Statement

- End with (;)
  - a=7;
- Compound of statements

```
{
a=7;b=13; c=a*b;
}
```

Control

- If <expression>

  (statement)

  else

  (statement)

-
```
var username= ”jono”;
if(username !=null)
        alert("Hai "+username+" Welcome ");
else{
        username=prompt("Welcome, whats Ur name");
        alert("Thanks, hello "+username);
}
```

- switch<expression>{ (statement) }
- While <exp> (statement)

```
switch(i){
case 1:
break;
case 2:
break;
case 3:
break;
default: // if all fails
break;
}
```

```
var count=0;
while (count<7){
        document.write(count+"  ");
        count++;
}
```

- do (statement) while<exp>

```
var c=0;
do
        document.write(c+"-");
while(c++<10);
```

- for (initialize;control;increment) (statement)

```
for(i=0,j=1,k=0,f=0;i<10;i++,f=j+k,j=k,k=f){
document.write("Fibonacci("+i+ ")= "+f);
document.write("<br>");
}
```

```
var book = {};
book.title ="Data Mining"; //set property of object book
book.chapter1 = new Object();
book.Chapter1.title = "Introduction of Data Mining";
book.Chapter1.pages =17;
book.Chapter2 = {title:"Associate Rule Algorithm",pages:27};

alert("Outline: "+book.title+" "+
"Chapter1 "+book.Chapter1.title+" "+
"Chapter2 "+book.Chapter2.title);
```

Client side Js

The browser environment

- Window object that serves as global object and global execution context for client-side Js code
- Hierarchy and DOM-Document Object Model that forms a part of it
- Event-driven programming model

Windows as global execution

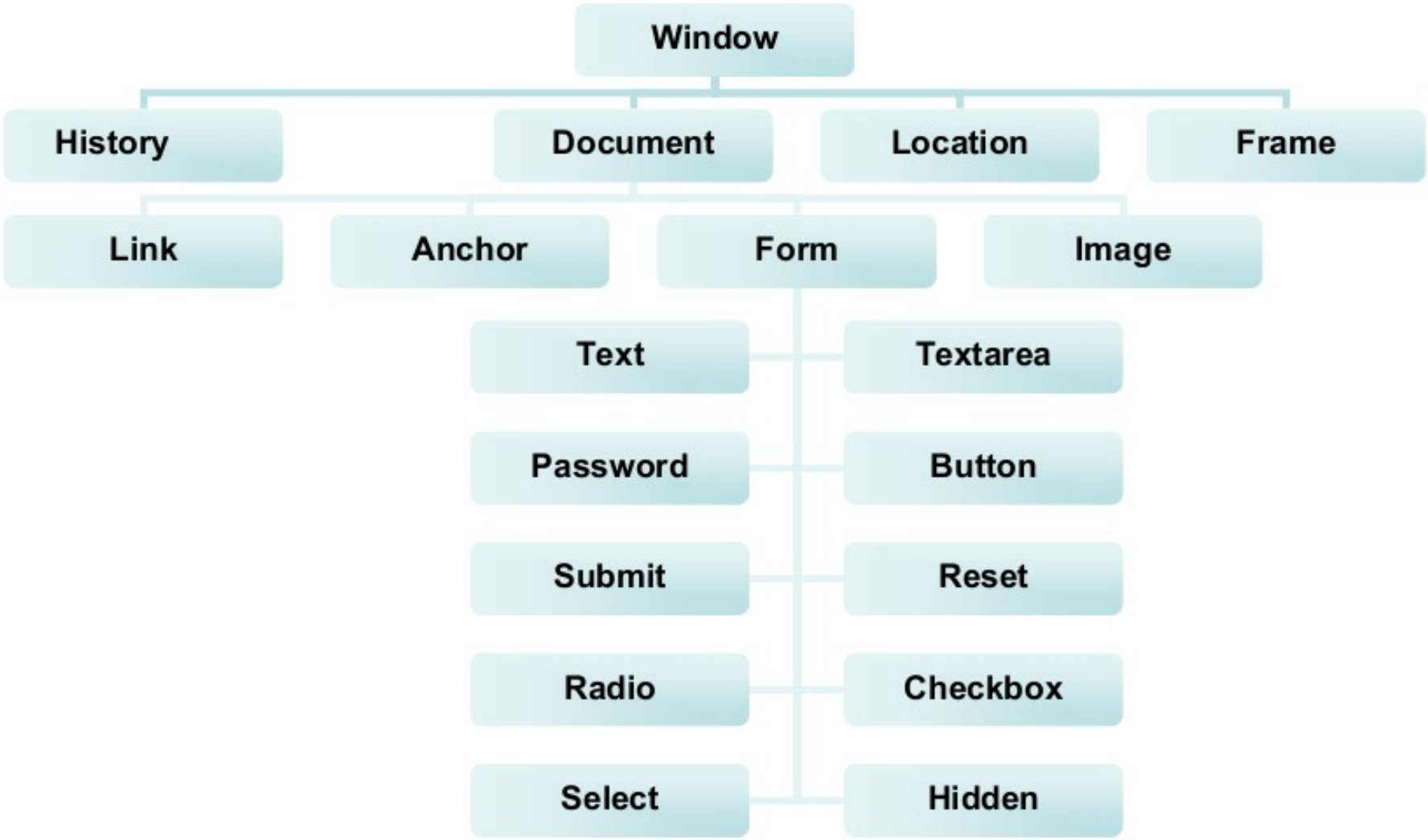
- Web browser = display HTML
- In Js, document object represent HTML doc
- Window object defines a number of properties and method that allow to manipulale window browser
- Has 2 self_ref: window and self
- As global object, all global var are defined as properties of the window

```
var a = 7;
window.a = 7;//same
```

Hierarchy (DOM Lv 0)

- Window → self/window/parent/top(various window object), navigator, frames[](array of window object), location, history, document, screen
- Document → forms[], anchors[], links[], images[], applets[]
- Forms → elements[]
- Element → options[]

```
window.document.forms[0];
parent.frames[0].document.forms[0].element[3].options[2].text;
```


Window
History
Document
Location
Frame
Link
Anchor
Form
Image
Text
Textarea
Password
Button
Submit
Reset
Radio
Checkbox
Select
Hidden

- The HTML DOM defines a standard set of objects for HTML, and a standard way to access and manipulate HTML documents
- All HTML elements, along with their containing text and attributes, can be accessed through the DOM.
  - The contents can be modified or deleted, and new elementscan be created.
- The HTML DOM is platform and language independent
  - It can be used by any programming language like Java, JavaScript, and VBScript

DOM Object

- Anchor object
- Document object
- Event object
- Form and Form Input object
- Frame, Frameset, and IFrame objects
- Image object
- Location object
- Navigator object
- Option and Select objects
- Screen object
- Table, TableHeader, TableRow, TableData objects
- Window object

```
<html>
<head>
      <title>About DOM</title>
</head>
<body>
<h2>DOM DOcument Object Model</h2>
<p>DOM is ...</p>
<p>HTML DOM is ...</p>
</body>
</html>
```

```
<html> //1
<head> //2
      <title>About DOM</title>//3
</head>
<body>//2
<h2>DOM DOcument Object Model</h2>//3
<p>DOM is ...</p>//3
<p>HTML DOM is ...</p>//3
</body>
</html>
```