

# Ingeniería de Sistemas

## Desarrollo y Servicios Web

### Sesión 7

*Fernando Barraza A.  
fbarraza@puj.edu.co*

# Sesión 7



- Objetivo: Conocer el lenguaje de programación JavaScript, su estructura y aplicación en el desarrollo de aplicaciones Web.
- Temas:
  - Historia y características
  - Uso del lenguaje
  - Sintaxis general
  - Estructuras, funciones y clases
  - Eventos

# Que es JavaScript?



- Es un lenguaje de programación muy utilizado para desarrollo de páginas y aplicaciones Web
- Corre en el lado cliente de la aplicación, es decir en el entorno del navegador de internet mediante un “motor” que lo interpreta
- Está presente en otros lenguajes como ActionScript de Macromedia, que acompaña al sistema Flash.

# Historia



- Fué inventado por Brendan Eich y apareció en el navegador Netscape 2.0 de la empresa del mismo nombre.
- En junio de 1997 fue adoptado como un estándar ECMA, con el nombre de ECMAScript. Poco después también lo fue como un estándar ISO.
- A través del modelo DOM promulgado por W3C se hace compatible entre los diversos navegadores del mercado.
- Su versión actual es la 1.9. La versión 2.0 tiene como base la especificación ECMA-262.

# Que se puede hacer con JS?



- Formularios de entrada de datos
- Acceder los elementos del HTML (Modelo DOM) para consultar y manipular sus valores y propiedades
- Construir páginas dinámicas (DHTML)
- Intercambiar información entre distintas ventanas de la aplicación
- Manipulación de elementos multimedia (graficos, video, texto, etc.)
- Comunicación con plug-ins (Flash, Java, Shockwave, etc.)

# Como escribir código en JS



- Se introduce mediante un tag `<script>` de HTML, en cualquier parte de la página web
- Otra forma es con el tag  
`<script language="JavaScript">`
- W3C recomienda utilizar el tag  
`<script type="text/javascript">`
- También es posible separando el código de de HTML y JS en archivos diferentes

# Ejemplo:



```
<html>
<head>
<title>Esta es una pagina web</title>
<script type="text/javascript">
    var mi_numero = 1;

    function calcula(numero) {
        return numero + mi_numero;
    }
</script>
</head>
<body>
<script>
    document.write(calcula(1));
</script>
</body>
</html>
```

# Ejemplo con código separado



- Código en JavaScript en archivo funciones.js:

```
function saludo(nombre) {  
    alert('Hola, ' + nombre);  
}
```

- Código HTML en archivo saludo.html:

```
<html>  
<head>  
<title>Esta es una pagina web</title>  
<script language="Javascript" src="funciones.js"></script>  
</head>  
<body>  
<script>  
    saludo('Ana');  
</script>  
</body>  
</html>
```



# Tipos de datos



- Números (enteros, decimales, etc...)
- Letras y números (cadenas de caracteres)
- Valores lógicos (True y False)
- Objetos (una ventana, un texto, un formulario, etc.)

# Variables



- Se definen en tiempo de ejecución y toman el tipo dependiendo del tipo de dato asignado. Ej:
  - `var myString = "Hello Word!";`
  - `var myNumber = 123;`
  - `var myBoolean = true;`

# Arreglos



- Los arreglos son variables que contienen un objeto de tipo `Array()`.
- Se pueden definir de varias formas:
  - `var myArray = new Array();` / ilimitado numero de elementos
  - `var myArray = new Array(15);` / quince elementos

# Arreglos(2)



- La asignación de un elemento de un arreglo es igual a la de una variable escalar, excepto porque se identifica el elemento por su número encerrado en “corchetes”:
  - `myArray[1] = “Hello”;`
  - `myArray[15] = 123;`
- Los elementos de un mismo arreglo pueden ser de tipos de datos diferentes.

# Condicional *if*



```
if (expresión logica) {  
    / código ejecuta si expresion logica es 'true'  
}  
else {  
    / código ejecutado si expresion logica es 'false'  
}
```

# Condicional múltiple



```
if (expresionlogica1) {  
    / código se ejecuta si condicionlogica1 es 'true'  
}  
else if (expresionlogica2) {  
    / código se ejecuta si condicionlogica1 es 'false'  
    / y si condicionlogica2 es 'true'  
}  
else {  
    / código se ejecuta si condicionlogica1 es 'false'  
    / código se ejecuta si condicionlogica2 es 'false'  
}
```

# Operadores



## LÓGICOS:

| Operador | Significado |
|----------|-------------|
| &&       | Y           |
|          | O           |
| !        | NO          |

## RELACIONALES:

| Operador | Significado       |
|----------|-------------------|
| ==       | Equivalencia      |
| <        | Menor que         |
| >        | Mayor que         |
| <=       | Menor o igual que |
| >=       | Mayor o igual que |
| !=       | Distinto de       |

# Operadores aritméticos



| OPERADOR | OPERACION        |
|----------|------------------|
| +        | SUMA             |
| -        | RESTA            |
| *        | MULTIPLICACION   |
| /        | DIVISION         |
| %        | MODULO O RESIDUO |



# Switch



```
switch (variable) {  
    case valor1:  
        // código para valor1  
        break;  
    case valor2:  
        // código para valor2  
        break;  
    case valor3:  
    case valor4:  
    case valor5:  
        // código para valor3, valor4 y valor5  
        break;  
}
```

# Ciclo *for*



```
for (expresioninicial; expresionlogica; expresiondeciclo) {  
    /Codigo se ejecuta hasta que expresionlogica es 'false'  
}
```

Ej:

```
for (x=1;x<=3; x++) {  
    document.write("Ciclo " + x);  
}
```

# Ciclo while



```
while (expresionlogica) {  
    /Codigo se ejecuta mientras expresionlogica es 'true'  
}
```

Ej:

```
var x = 1;  
while (x < 5) {  
    document.write("Ciclo: " + x);  
    x++;  
}
```

# Ciclo *do-while*



```
do {  
    / Código se ejecuta hasta que expresionlogica sea 'false'  
} while (expresionlogica);
```

Ej:

```
var x = 2;  
do {  
    document.write("Ciclo " + x);  
    x++;  
} while (x < 6);
```

# Funciones



```
function functionname(param1, param2, ...)  
    / function code  
    return value;  
}
```

Ej:

```
function mostrarsaludo(nombre) {  
    return "Hola " + nombre + "!";  
}
```

```
document.write(mostrarsaludo('Maria'));
```

# Algunas funciones comunes JS



- `accion=confirm("Pulsa un botón");`
- `document.write("Hola");`
- `Window.open("http://www.google.com");`
- `Window.print();`

# Clases



- Se definen por medio de funciones:

```
function className(param1) {  
    this.varString = param1;  
    this.varString2 = 'string value';  
    this.varNumber = 1;  
}
```

- Para instanciar un objeto de la clase *className* y referencia un atributo del objeto:

```
var = myObject = new ClassName ('String Value');  
myObject.varString2 = 'new string value';
```

# Clase predefinidas en JS y algunos de sus métodos



- Array: *length, join, reverse, push, pop, shift.*
- Date: *getDate, getMonth, getFullYear.*
- Math: *round, random, max, min.*
- String: *concat, match, replace, search, substr, toLowerCase, toString, toUpperCase, valueOf.*



# Métodos de clases



- Son funciones que están relacionadas en el constructor de la clase como otro miembro de la misma:

```
function className(param1) {  
    this.varString = param1;  
    this.varString2 = 'string value';  
    this.varNumber = 1;  
    this.myFunction = myFunction;  
}
```

```
function myFunction () {  
    / Function code  
}
```

# Créditos



- Programación en JavaScript. Licencia GNU.
- JavaScript Bible, second edition.