



CFGS DAW

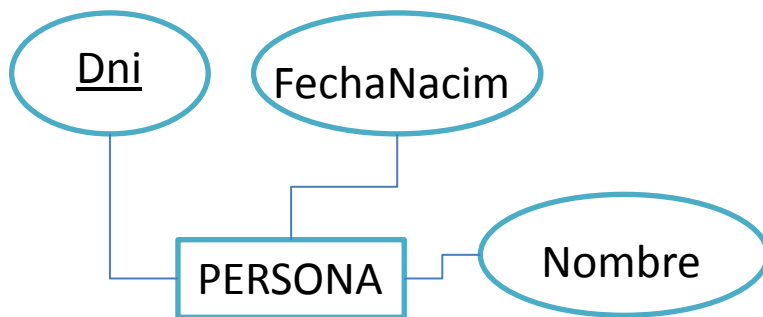
Mòdul 6: Desenvolupament web en entorn
client

INS Joan d'Àustria

Objetos predefinidos



- Javascript permite trabajar con objetos y proporciona unos cuantos objetos predefinidos
- Un objeto es un tipo estructurado que tiene propiedades y métodos
 - Las propiedades son valores asociados con el objeto
 - Los métodos son acciones que pueden ser llamadas sobre los objetos



+ Programación

PERSONA
Dni
Nombre
FechaNacim
Edad():Entero
...

Objetos predefinidos



- Podemos crear nuestros propios objetos

```
var punto= new Object( );  
punto.x = 2.3;  
punto.y = -1.2;
```

```
var punto= {  
    x: 2.3,  
    y: 1.2  
};
```

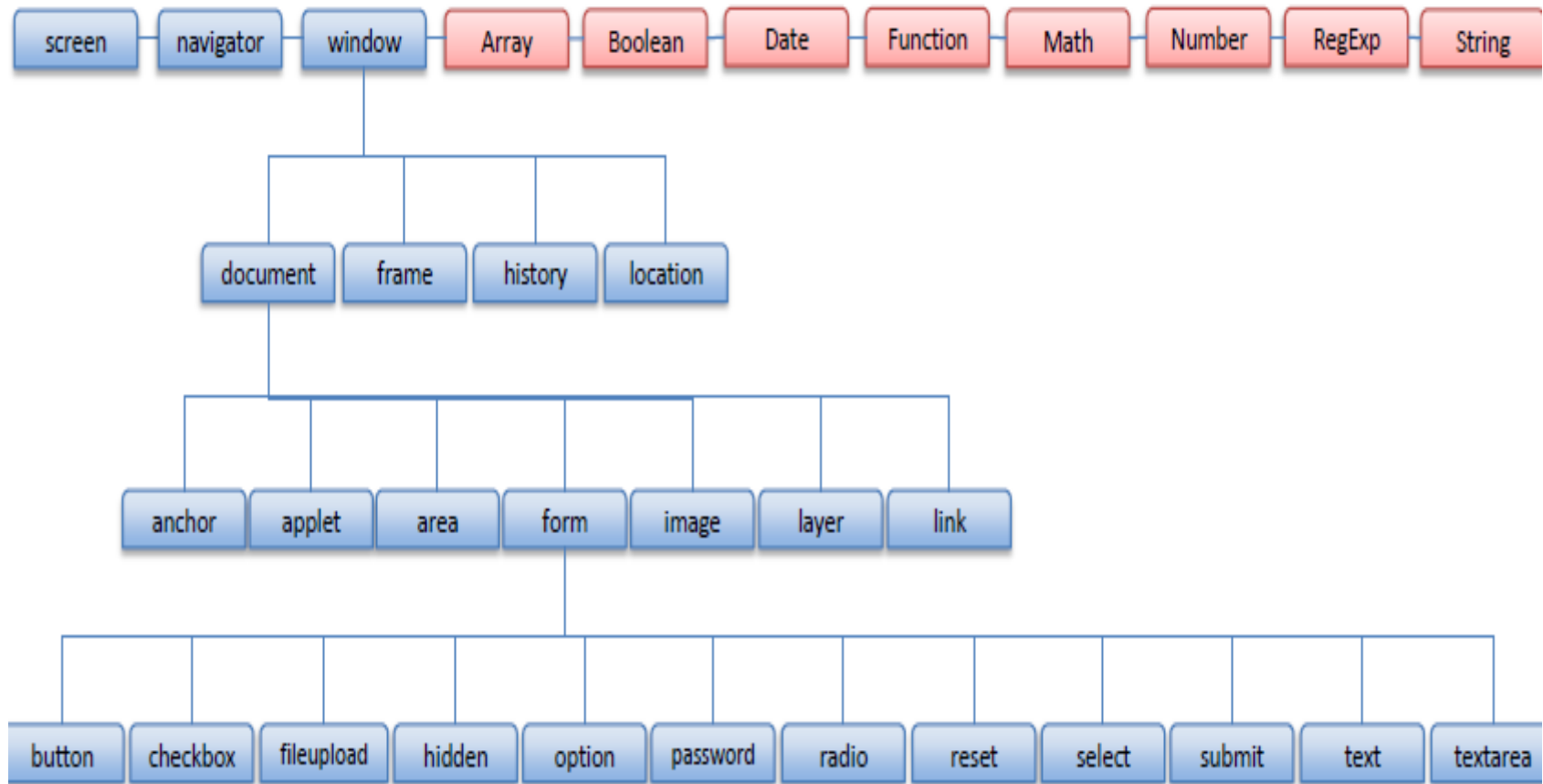
```
var punto= {};  
punto.x= 2.3;  
punto.y= 1.2;
```

Para acceder a las propiedades y métodos de un objeto, utilizaremos un punto:

Objeto.propiedad

Objeto.método(parámetros)

Objetos predefinidos





El objeto Date

- Constructor: `new Date()`
- Si no se le pasan parámetros, el constructor toma la fecha del sistema
- Se le puede pasar como parámetros la fecha deseada
 - Con base a `Date.parse`
 - `month/date/year` (como: 6/13/2004)
 - `month_name date, year` (como: January 12, 2004)
 - `day_of_week month_name date year hours:minutes:seconds time_zone` (como: Tue May 25 2004 00:00:00 GMT-0700)
 - ISO 8601 `YYYY-MM-DDTHH:mm:ss.sssZ` (como: 2004-05-25T00:00:00).
 - Con base a `Date.UTC` (base 0 y horas de 0 a 23)
 - `YYYY,MM,DD,HH,MM,SS`

El objeto Date



```
var fecha;  
fecha=new Date ("10/25/2012");  
fecha = new Date("25 Nov 1971");  
fecha = new Date("Nov 25 1971");  
fecha = new Date("25 Nov 1971 00:30");  
fecha = new Date("25 Nov 1971 00:30:24");
```

```
fecha= new Date(2012, 0);  
fecha= new Date(2005, 4, 5, 17, 55, 55);
```

```
var fecha;  
  
fecha=new Date ("13/10/2009");  
fecha = new Date("25 Ago 1975");
```

```
var fecha;  
  
fecha=new Date(2012, 13);  
fecha= new Date(2005, 12, 5, 17, 65, 55);
```





El objeto Date

- Existe otro constructor en ecmascript 5:

```
var fecha = Date.now();
```

Este constructor retorna la representación de la fecha en milisegundos

```
var start=Date.now();  
alert(start);  
var stop=Date.now();  
alert(stop-start);
```

Sólo esta implementado en IE9+, Firefox 3+, opera 10.5+ y Chrome. Para navegadores anteriores podríamos utilizar el operador + para convertir la fecha a número:

```
start = +new Date();  
alert(start);  
var stop = +new Date();  
alert(stop-start);
```



El objeto Date

- http://www.w3schools.com/jsref/jsref_obj_date.asp

Métodos

getDate()	getTimezoneOffset()	getFullYear()	setTime()
getDay()	getUTCDate()	parse()	setUTCDate()
getFullYear()	getUTCDay()	setDate()	setUTCFullYear()
getHours()	getUTCFullYear()	setFullYear()	setUTCHours()
getMilliseconds()	getUTCHours()	setHours()	setUTCMilliseconds()
getMinutes()	getUTCMilliseconds()	setMilliseconds()	setUTCMinutes()
getMonth()	getUTCMinutes()	setMinutes()	setUTCMonth()
getSeconds()	getUTCMonth()	setMonth()	setUTCSeconds()
getTime()	getUTCSeconds()	setSeconds()	setYear()



El objeto Number

- Constructor: `new Number(valor)`
- Al asignar un valor numérico a una variable ya establece el constructor?

```
var x=new Number(3);
```

¿≡?

```
var x=3;
```

- Con el constructor `new`, si el valor pasado no se puede convertir a número, retorna `NaN`

El objeto Number



- http://www.w3schools.com/jsref/jsref_obj_number.asp

Propiedades

[MAX_VALUE](#)

[MIN_VALUE](#)

[NEGATIVE_INFINITY](#)

[NaN](#)

[POSITIVE_INFINITY](#)

Métodos

[toExponential\(x\)](#)

[toFixed\(x\)](#)

[toPrecision\(x\)](#)

[toString\(\)](#)

[valueOf\(\)](#)

El objeto Number



```
var num = 10;  
alert(num.toString());  
alert(num.toString(2));  
alert(num.toString(8));  
alert(num.toString(10));  
alert(num.toString(16));
```



```
var x = new Number(3);  
var y = 3;  
alert(typeof x);  
alert(typeof y);
```





El objeto Math

- Se proporciona como una localización común para trabajo matemático que facilita el trabajo de forma mucho más eficiente que programando directamente en javascript.
- No tiene constructor, sino que las propiedades y métodos pueden ser llamadas directamente usando el objeto Math.



El objeto Math

- http://www.w3schools.com/jsref/jsref_obj_math.asp

Propiedades			
<u>E</u>	<u>LN10</u>	<u>LOG10E</u>	<u>SQRT1_2</u>
<u>LN2</u>	<u>LOG2E</u>	<u>PI</u>	<u>SQRT2</u>

Métodos		
<u>abs(x)</u>	<u>cos(x)</u>	<u>pow(x,y)</u>
<u>acos(x)</u>	<u>exp(x)</u>	<u>random()</u>
<u>asin(x)</u>	<u>floor(x)</u>	<u>round(x)</u>
<u>atan(x)</u>	<u>log(x)</u>	<u>sin(x)</u>
<u>atan2(y,x)</u>	<u>max(x,y,z,...,n)</u>	<u>sqrt(x)</u>
<u>ceil(x)</u>	<u>min(x,y,z,...,n)</u>	<u>tan(x)</u>



El objeto Math

```
alert(Math.ceil(25.9)); //26  
alert(Math.ceil(25.5)); //26  
alert(Math.ceil(25.1)); //26
```

```
alert(Math.round(25.9)); //26  
alert(Math.round(25.5)); //26  
alert(Math.round(25.1)); //25
```

```
alert(Math.floor(25.9)); //25  
alert(Math.floor(25.5)); //25  
alert(Math.floor(25.1)); //25
```

```
alert(Math.SQRT2);  
alert(Math.SQRT1_2);
```

```
alert(Math.sqrt(9));  
alert(Math.pow(9,1/2));
```

Math.random() retorna un número entre 0 y 1 sin incluir los propios 0 y 1.
¿Como podemos obtener un número aleatorio entre X y Y?

```
number = Math.floor(Math.random() * rango) + valor_inicial
```

```
var num = Math.floor(Math.random() * 10) + 1; //Número entre 1 y 10
```





El objeto String

- Constructor: new String(valor)
- Al asignar un valor numérico a una variable ya establece el constructor?

```
var x=new String("Hola");
```

¿≡?

```
var x="Hola";
```

- Ejercicio: Diferencia entre tipos de referencia y tipos primitivos (boolean, number, string)

```
var s1 = "Hola don Pepito";  
s1.color = "azul";  
alert(s1.color);  
var s2 = new String("Hola don Jose");  
s2.color = "verde";  
alert(s2.color);
```



El objeto String



- http://www.w3schools.com/jsref/jsref_obj_string.asp

Propiedades

[length](#)

Métodos

<u>charAt()</u>	<u>indexOf()</u>	<u>search()</u>	<u>substring()</u>
<u>charCodeAt()</u>	<u>lastIndexOf()</u>	<u>slice()</u>	<u>toLowerCase()</u>
<u>concat()</u>	<u>match()</u>	<u>split()</u>	<u>toUpperCase()</u>
<u>fromCharCode()</u>	<u>replace()</u>	<u>substr()</u>	<u>valueOf()</u>

El objeto String

```
var s1 = "Hola don Pepito";
alert(s1[1]);
alert(s1.charAt(1));
alert(s1.charCodeAt(1));
```



```
var s1 = "Hola don Pepito";
alert(s1.slice(3));
alert(s1.substring(3));
alert(s1.substr(3));
alert(s1.slice(3, 7));
alert(s1.substring(3,7));
alert(s1.substr(3, 7));
alert(stringValue.slice(-3));
alert(stringValue.substring(-3));
alert(stringValue.substr(-3));
alert(stringValue.slice(3, -4));
alert(stringValue.substring(3, -4));
alert(stringValue.substr(3, -4));
```



El objeto String



¿Existe un método trim? ¿Y ltrim o rtrim? ¿En qué navegadores? Crea un bloque de código que pida un string y simule esta función en todos los navegadores

¿Qué diferencia existe entre toLowerCase(), toLocaleLowerCase(), toUpperCase(), y toLocaleUpperCase()? Ejemplos.

