

# Javascript

2n DAW - DWECC



# Què és Javascript?

- Llenguatge de script per a la creació de webs dinàmiques.
  - Crear visualitzacions atractives i amb major interactivitat.
  - Permet gestionar diferents implementacions entre diversos navegadors.
- Llenguatge interpretat (No es compila)
  - El navegador s'encarrega d'interpretar i executar el codi.
- Javascript no és Java
  - Basa la seva sintaxi amb C
- Diversos Frameworks
  - jQuery, Mootools, ...
  - JSON per transmissió de dades
  - Ajax
  - ...
- S'executa en entorn client, però està canviant la tendència.
  - Node.js, Jaxer, EJScript, RingoJS, ...

# Funcionalitats i Característiques

1. Llenguatge interpretat en el navegador. Pot estar deshabilitat.
2. No pot escriure fitxers en el servidor.
3. Reacciona a la interacció amb l'usuari.
4. Controla diverses finestres, marcs, plugins, applets, ...
5. Pre-processa dades en el client.
6. Modifica estils i continguts dels navegadors.
7. Pot sol·licitar fitxers al servidor.

# Història

- LiveScript (Brendan Eich, 1995): llenguatge d'Script per Netscape Navigator 2.0
- JavaScript: Acord entre Netscape i Sun
- Estandardització:
  - ECMA-262 - ECMAScript Language Specification (1997)
  - Adoptat per ISO com ISO/IEC 16262
- Variants:
  - Microsoft: JScript
  - Firefox: JavaScript
  - Chrome: JavaScript
  - Opera: ECMAScript
  - Adobe Flash: ActionScript 3

# Compatibilitats

**És compatible Javascript en tots el dispositius?**

**Suportat per tots el navegadors?**

**Es pot habilitar i deshabilitar?**

# Compatibilitats

- Pràcticament tots els navegadors suporten Javascript, però ens hem d'assegurar.
- Hi pot haver algunes incompatibilitats entre navegadors.
- Alguns dispositius mòbils no poden executar JS.
- A tots els navegadors es pot deshabilitar l'execució del codi per l'usuari.

## Exercici:

Busca com es deshabilita JavaScript en 2 navegadors diferents.

Visualitza 3 pàgines webs i indica quines diferències i trobes.

# Eines i Utilitats

Editor de text:

- Edició de codi en diferents llenguatges.
- Sintaxi de colors.
- Verificació de la sintaxi.
- Diferencia comentaris del resta del codi.
- Genera parts de codi automàtiques.
- Utilitats addicionals.

Exemples:

Notepad++, Aptana Studio, Eclipse, Netbeans, SublimeText, KompoZer, Amaya, ...

# Eines i Utilitats

## Navegador Web

- No necessita Internet per provar el scripts de JS.
- Inspector d'elements HTML.
- Depurador.
- Editor d'estils.
- Altres funcions...

## Exemples

Firefox, Chrome, Explore, Opera, Safari, ...



# Integració codi JS - Mètode 1

Codi JavaScript dins un document.

- El codi JS s'escriu entre etiquetas `<script></script>` en qualsevol part del document.
- Es recomana incloure aquestes etiquetes dins el HEAD.
- És necessari incloure l'atribut **type** a l'etiqueta `<script type="text/javascript">`
- Aquest mètode s'utilitza quan es defineix in bloc de codi molt petit o quan es volen incloure instruccions específiques.
- Principal inconvenient, es que si es vol modificar, s'ha de modificar a totes les parts on hi hagi el codi.

# Mètode 1 - Exemple

Exemple:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <title>Ejemplo de código JavaScript en el propio documento</title>
    <script type="text/javascript">
        console.log("Un mensaje de prueba");
    </script>
</head>
<body>
    <p>Un párrafo de texto.</p>
</body>
</html>
```

# Integració codi JS - Mètode 2

Definir Javascript en un fitxer extern.

- Aquest document extern s'enllaçarà en les nostres pàgines XHTML mitjançant l'etiqueta `<script>`
- ```
<script type="text/javascript" src="/js/codigo.js"></script>
```
- Es poden crear tots els arxius JavaScript que es vulgui i enllaçar-los.
- Apart del atribut `type`, aquest mètode requereix l'atribut **src**, que indica la ruta de l'arxiu extern.
- Extensió `.js`
- Permet simplificar i ordenar el codi.

# Mètode 2 - Exemple

Archivo `codigo.js`

```
console.log( "Un mensaje de prueba" );
```

Documento XHTML

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" >
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <title>Ejemplo de código JavaScript en el propio documento< /title>
    <script type="text/javascript" src="/js/codigo.js"></script>
</head>
<body>
    <p>Un párrafo de texto.< /p>
</body>
</html>
```

# Integració codi JS - Mètode 3

Incloure JavaScripts en els elements XHTML

- És el menys utilitzat ja que consisteix en incloure instruccions JS dins del codi XHTML.
- Inconvenient:
  - Codi XHTML brut.
  - Complica el manteniment del codi JavaScript
- En general, aquest mètode només s'utilitza per definir certs events i en alguns casos especials.

# Mètode 3 - Exemple

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <title>Ejemplo de código JavaScript en el propio documento</title>
</head>
<body>
    <p onclick="console.log('Un mensaje de prueba')">Un párrafo de texto.</p>
</body>
</html>
```

# Navegador no permet JS

Etiqueta <noscript>

- Permet definir quin text proporcionar al usuari quan el navegador no suporta o no té activat JavaScript.
- Té que anar dins del BODY.

```
<head> ... </head>
```

```
<body>
```

```
  <noscript>
```

```
    <p>Bienvenido a Mi Sitio</p>
```

```
    <p>La página que estás viendo requiere para su funcionamiento el  
    uso de JavaScript. Si lo has deshabilitado intencionadamente,  
    por favor vuelve a activarlo.</p>
```

```
  </noscript>
```

```
</body>
```

# Glossari Bàsic

## Script

- Cada un dels programes, aplicacions o blocs de codi del llenguatge JS, siguin de 2 línies de codi o de mil.

## Sentencia

- Cada una de les instruccions que formen un script.

## Paraules reservades

- Són les paraules reservades que s'utilitzen per construir les sentències en JS.

`break, case, catch, continue, default, delete, do, else, finally, for, function, if, in, instanceof, new, return, switch, this, throw, try, typeof, var, void, while, with.`



# Sintàxis

## Normes bàsiques:

- No es tenen en compte els espais en blanc i les noves línies.
  - L'interpret ignora qualsevol espai en blanc, per tant es poden utilitzar per ordenar el codi.
- Distinció entre Majúscules i minúscules.
- No es defineix el tipus de variable.
  - Quan cream una variable no es necessari indicar el tipus de dades que s'emmagatzameran.
- No és necessari acabar cada sentència en ;
  - No és obligatori però és convenient.
- Es poden incloure comentaris.

# Exercici - Primer JavaScript

Es demana crear un HolaMundo.js que saludi als logs de la consola.

1. Mètode 1 d'integració de codi JS.
2. Mètode 2 d'integració de codi JS.
3. Mètode 3 d'integració de codi JS.
4. Refer el codi dels dos exercicis anteriors perquè saludi per finestra emergent.
5. Refer el codi, i què els scripts anteriors saludin modificant el fitxer XHTML

Deshabilita JS del teu navegador.

6. Afegeix comentaris al teu codi i missatges `<noscript>`

# Sintaxi del Llenguatge

## Comentaris

- De una línia
  - **// Això és un comentari de una línia**
  - **//**
  - **// Deixem una línia en blanc**
- De varies línies
  - **/\* Això és un**
  - **comentari de varies**
  - **línies \*/**
- Utilització de comentaris:
  - Codi més clar i informatiu de l'ús del codi.
  - Desactivar codi.
  - Informació de contacte per possibles incidències.

# Variables

- Crear variables emprant la paraula reservada 'var':
  - **var edad;**
  - **var edad1,edad2,edad3;**
- Assigna valor a una variable ja creada:
  - **edad = 15;**
- Crear variables i assignar valor:
  - **var edad = 15;**
- Utilitzacio de variables:
  - Formades per caràcters alfanumèrics i `_`. No es poden emprar signes, %, ...
  - No podem començar per un número i no solen començar en M.
  - No tenen tipus associat, podem canviar de tipus.

# Tipus de dades

## Números:

- Enter: 726
- Decimal: 3.75
- Científic: 3e7 (30000000)
- Octal: Va precedit per un 0. 0327
- Hexadecimal: Va precedit per 0x. 0xA3F2

# Exercici u1e1\_numeros

Crea un programa en el que creis 5 variables numèriques (enter, decimal, científic, octal i hexadecimal)

Mostra el seu valor en un document HTML. (Escriu en el document).

Comenta el codi, indicant autor i nom de l'exercici.

# Cadenes

Cadenes:

- “Text entre cometes”
- “645464”
- “Cadenes”+”Concatenades”

Utilització de cadenes:

- Salt de línia: \n
- Tabulador: \t
- Cometes simples: \'
- Cometes dobles: \”

# Exercici u1e2\_cadenes

Crea un programa que crei 4 variables de tipus cadena amb els següents valors, “Hola”, “7”, “13” i “Adeu”.

Mostra en un Alert una frase que contengui cometes simples.

Mostra a un Alert que ocupi una línia de codi les variables 1 i 4 separades per un salt de línia.

Mostra en un Alert la suma de les variables 2 i 3.

Mostra a un Alert la suma de totes les variables.

Comenta el codi indicant autor i nom de l'exercici.



# Booleans

Booleans:

- True
- False

Objectes:

- String
- Array
- Date
- ...

# Conversions entre tipus de dades

Conversions:

- Enter+Float = Float
- Número+Cadena = Cadena

Conversió de cadenes a números:

- `parseInt("32");`
- `parseFloat("32.5");`

Conversió de números a cadenes:

- `" "+número`
- `toString();`

# Operadores

Sintaxis	Nombre	Tipos de operandos	Resultados
==	Igualdad	Todos	Boolean
!=	Distinto	Todos	Boolean
===	Igualdad estricta	Todos	Boolean
!==	Desigualdad estricta	Todos	Boolean
>	Mayor que	Todos	Boolean
>=	Mayor o igual que	Todos	Boolean
<	Menor que	Todos	Boolean
<=	Menor o igual que	Todos	Boolean

Operadores de comparación

# Exercici u1e3\_comparació

Crea un programa en el que mostris el resultat de diverses operacions mitjançant Alert, mostrant el text exacte de la operació realitzada i el seu resultat.

**Ex:**

**var operacion1 = 10==10,**

**alert(“La operacion 10==10 es ”+operacion1);**

Les operacions a realitzar són:

- 10 == 10
- “Laura” == “laura”
- “123” == 123
- 10 === 10
- “Laura” > “laura”
- “123” === 123
- 10 === 10.0
- “Laura” < “laura” - parseInt(“123”) === 123

Comenta el codi indicant autor i nom de l'exercici.

# Operadors aritmètics

Sintaxis	Nombre	Tipos de operandos	Resultados
+	Más	Entero, real, cadena	Entero, real, cadena
-	Menos	Entero, real	Entero, real
*	Multiplicación	Entero, real	Entero, real
/	División	Entero, real	Entero, real
%	Módulo	Entero, real	Entero, real
++	Incremento	Entero, real	Entero, real
--	Decremento	Entero, real	Entero, real
+valor	Positivo	Entero, real, cadena	Entero, real
-valor	Negativo	Entero, real, cadena	Entero, real

Operadores aritméticos

# Operadors d'assignació

Sintaxis	Nombre	Ejemplo	Significado
=	Asignación	x=y	x=y
+=, -=, *=, /=, %=	Operación y asignación	x+=y	x=x+y
<<=	Desplazar bits a la izquierda	x<<=y	x=x<<y
>=, >>=, >>>=	Desplazar bits a la derecha	x>=y	x=x>y
&=	Operación AND bit a bit	x&=y	x=x&y
=	Operación OR bit a bit	x =y	x=x y
^=	Operación XOR bit a bit	x^=y	x=x^y
[]=	Desestructurar asignaciones	[a,b]=[c,d ]	a=c, b=d

Operadores de asignación

# Operador Booleans

Sintaxis	Nombre	Operandos	Resultados
&&	And	Boolean	Boolean
	Or	Boolean	Boolean
!	Not	Boolean	Boolean

La operació AND només és TRUE quan tots els operands són TRUE.

La operació OR és TRUE sempre que hi hagi un operand que sigui TRUE.

La operació NOT canvia el valor del boolean resultant.

# Operadors d'Objecte

Punt:

- Objecte.propietat
- Objecte.mètode

“Corchetes”:

- Crear un array: `var a = [“Mao”, “Ciutadella”, “Alaior”];`
- Enumerar un element d'un array: `a[1] = “Ciutadella”;`
- Enumerar propietat d'un objecte: `a[“color”] = “blau”;`



# Operadors d'Objecte

Delete:

- `Delete a[2];` //Borraria l'element "Alaio" i el substituiria per Undefined.

In:

- Retorna TRUE si l'objecte conté la propietat o mètode
- Ex: `"write" in document`

InstanceOf:

- Retorna TRUE si és una instància d'un objecte natiu de JS.
- Ex: `a = new Array (1,2,3);`
- `a instanceof Array;` //Retorna TRUE

# Operadors d'Objecte

Coma:

- Operacions que s'executen de esquerra a dreta. `var nom,adreça,cognoms;`
- Operació (loop): `for(var i=0,j=0;i<125M i++;j+10)`

Interrogació (Operador Condicional)

- Forma reduïda de `if...else`
- `Condicció ? Expressió si es certa : Expressió si es falsa;`
- Ex: `var a= 3, b= 5;`
- `var r = a > b ? a : b;`

Typeof:

- Retorna el tipus de valor d'una variable o expressió
- Els tipus són `number`, `string`, `boolean`, `object`, `function`, `undefined`.

# Regles de precedència d'operadors

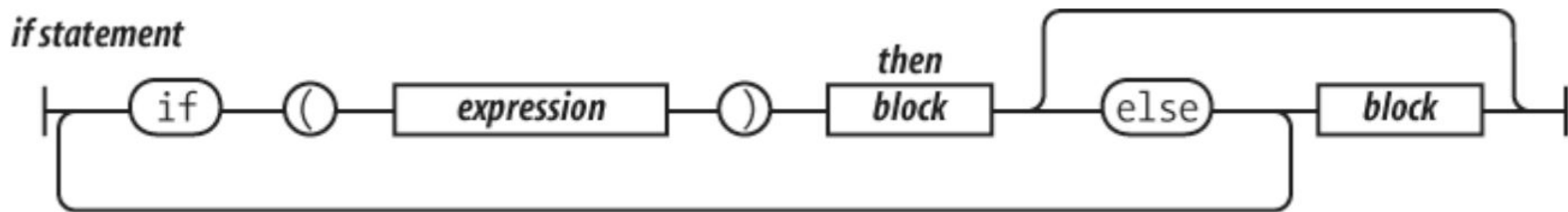
## Reglas de precedencia de operadores

- `() [] .` (el operador punto sirve para los objetos)
- `! - ++ --`
- `* / %`
- `+-`
- `<< >> >>>` (desplazamientos a nivel de bit)
- `< <= > >=`
- `== !=`
- `& ^ |` (lógicos a nivel de bit)
- `&& ||` (lógicos booleanos)
- `= += -= *= /= %= <<= >>= >>>= &= ^= !=` (asignación)

# Estructuras de control

- IF
- IF...ELSE
- SWITCH
- WHILE
- DO...WHILE
- FOR
- FOR...IN
- TRY

# Estructuras de control - IF & IF...ELSE



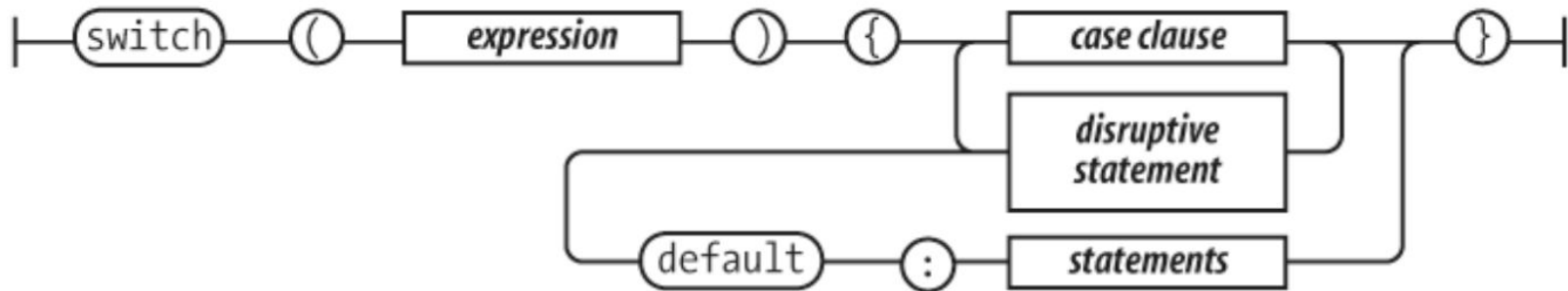
```
if(condicion) {  
    ...  
}
```

```
if(condicion) {  
    ...  
}  
else if{  
    ...  
}  
else {  
    ...  
}
```

```
if(edad < 12) {  
    console.log("...");  
} else if(edad < 19) {  
    console.log("xxx");  
} else if(edad < 35) {  
    console.log("yyyy");  
} else {  
    console.log("zzzz");  
}
```

# Estructuras de control - SWITCH

*switch statement*



```
switch(dia) {  
  case 1: console.log("Hoy es lunes."); break;  
  case 2: console.log("Hoy es martes."); break;  
  case 3: console.log("Hoy es miércoles."); break;  
  case 4: console.log("Hoy es jueves."); break;  
  case 5: console.log("Hoy es viernes."); break;  
  case 6: console.log("Hoy es sábado."); break;  
  case 0: console.log("Hoy es domingo."); break;  
}
```

# Estructuras de control - WHILE & DO...WHILE

*while statement*

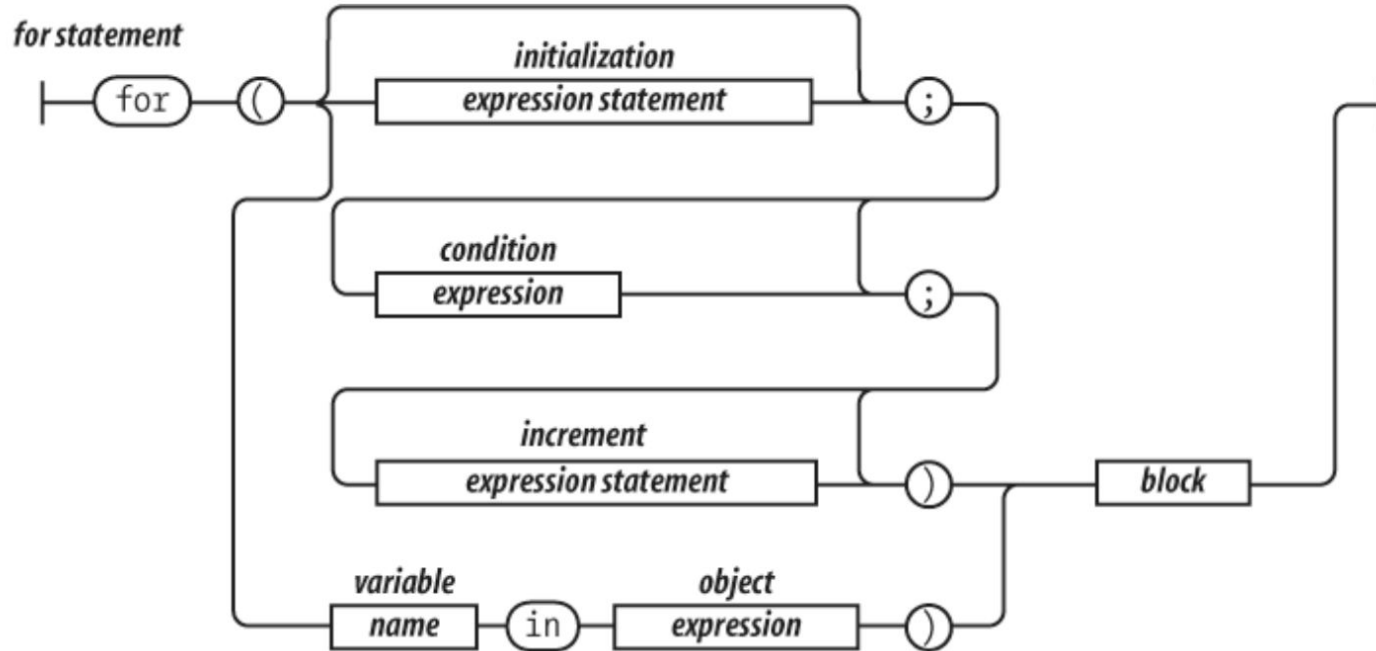


```
var veces = 0;
```

```
while (veces < 7) {  
    console.log("Mensaje " + veces);  
    veces++;  
}
```

```
do {  
    instrucciones  
} while (condicion)
```

# Estructures de control - FOR



```
for(inicializacion; condicion; actualizacion) {  
    ...  
}
```



# Estructuras de control - FOR... IN

```
for(propiedad in object) {  
    if (object.hasOwnProperty(propiedad)) {  
        ...  
    }  
}
```

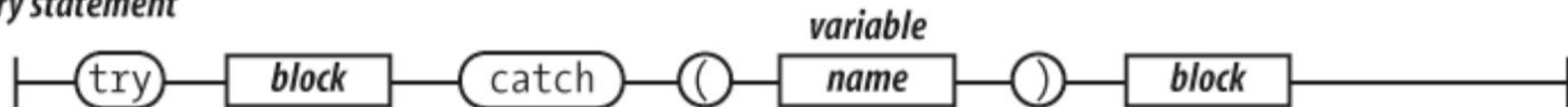
```
for(indice in array) {  
    ...  
}
```

```
var dias = ["Lunes", "Martes", "Miércoles", "Jueves", "Viernes", "Sábado", "Domingo"];
```

```
for(i in dias) {  
    console.log(dias[i]);  
}
```

# Estructuras de control - TRY

*try statement*



```
try {  
    funcion_que_no_existe();  
} catch(ex) {  
    console.log("Error detectado: " + ex.description);  
}
```

```
abrirFichero()  
try {  
    escribirFichero(datos);  
} catch(ex) {  
    // Tratar la excepción  
} finally {  
    cerrarFichero(); // siempre se cierra el recurso  
}
```

# Exercici u1e4\_ifelse

Crea un programa que demani al usuari que introdueixi una edad i mostri el següent missatge en funció del número introduït.

- 0-12 Nen
- 13-26 Jove
- 26-60 Adult
- >60 Jubilat

Per guardar el valor del usuari podem utilitzar:

**var resposta = prompt(“Mensaje”);**

També cal comprovar que no s'introdueixin números fora de rang. Comenta el codi amb els comentaris que trobis oportuns, nom i títol del exercici.

# Exercici u1e5\_for

Crea un programa que generi un llista d'hores que vagin des de les 9 fins les 21:30 de 30 min en 30 min.

Enlloc de emprar alert, utilitzeu la instrucció `document.write("Missatge")` que mostra les dades en el HTML.

Tindràs que emprar un for per les hores i una variable booleana que anirà canviant a TRUE o FALSE si es hora en punt o i mitja.

Comenta el codi amb els comentaris oportuns, autor i nom de l'exercici.

# Exercici u1e6\_for2

Crea un programa que crei un llista d'hores que vagi de les 9 fins les 21 de 5 min en 5 min.

Utilitza `document.write`

Comenta el codi amb els comentaris oportuns, autor i nom de l'exercici

# Exercici u1e7\_while

Crea un control parental per que no es pugui entrar a una pàgina.

Es demanarà a l'usuari que escrigui el cognom d'algun personatge (Pregunta el que vulguis).

Mentre l'usuari no introdueixi el valor correcte, el missatge seguirà sortint.

Comenta el codi amb els comentaris oportuns, autor i nom de l'exercici.

# Exercici u1e8\_while2

Modifica l'exercici anterior de la següent manera:

- Es carrega la mateixa pregunta que abans.
- En cas que l'usuari introdueixi el nom i el cognom malament, mostrarem “ERROR, intenta un altre cop” i una altre vegada la pregunta.
- Si introdueix el nom correctament, mostrarem el missatge “Et falta el cognom” i una altre vegada la pregunta.
- Si introdueix el cognom correctament, mostrarem el missatge “Et falta el nom” i una altre vegada la pregunta.

Comenta el codi com cregueis oportú i l'autor i el nom de l'exercici.

# Exercici u1e9\_switch

Modifica el programa ifelse per què funcioni amb SWITCH

Comenta el programa, autor i nom de l'exercici.



# Exercici final

Realitza una pàgina amb un script que calculi el valor de la lletra del NIF.

Algoritme:

- Comprovar que el num està entre 0 i 999999999.
- Calcula el “resto” de la divisió entra el num de DNI i el num 23.
- Selecciona la lletra dins l’array corresponent:
  - var letras = ['T', 'R', 'W', 'A', 'G', 'M', 'Y', 'F', 'P', 'D', 'X', 'B', 'N', 'J', 'Z', 'S', 'Q', 'V', 'H', 'L', 'C', 'K', 'E', 'T'];
- L’script ha de demanar el num al usuari.
  - Si el num no es correcte, indica un misstage d’error.
  - Si es correcte, calcular i mostra el NIF.

**Quan acabis, realitza el mateix amb botons per introduir els num 1 a 1. (IG)**

# Objectes predefinitos pel llenguatge

Es diu que Javascript es un llenguatge basat en objectes.

- A Javascript no es defineixen classes, només objectes.
- És un llenguatge basat en prototips (no basat en classes).
  - Es poden crear objectes copiant prototips d'altres objectes.

Un objecte Javascript es un conjunt de variables amb un nom.

- Les variables del objecte es denominen **Propietats**
  - Les propietats poden ser valor de qualsevol tipus de dades: arrays, funcions i altres objectes.
- Les propietats que son funcions s'anomenen **Mètodes**.

# Objectes

Es poden crear objectes directament indicant les seves propietats:

```
persona=new Object();  
persona.nom="Joan";  
persona.id= 12389;
```

O amb una sola instrucció, indicant les propietats entre claus:

```
var persona = {nom: "Joan", id: 12389}
```

O definir un constructor

```
fucntion persona(nom, id){  
  this.nom = nom;  
  this.id = id;  
}
```

# Objectes

Es poden definir mètodes per un objecte dins del constructor

```
function persona (nom, id){  
  this.nom = nom;  
  this.id = id;  
  function renombra(nom){  
    this.nom = nom;  
  }  
}
```

I s'invoca sobre l'objecte:

```
var juan = new persona("Juan",12345);  
juan.renombra("Juanjo");
```

# Objectes

Amb els constructors es poden anar creant diversos objectes:

```
var juan = new persona("Juan", 12345);
```

```
var lluis = new persona("Lluis", 12135);
```

Accedim a les seves propietats:

```
nom = persona.nom;
```

```
nom = persona["nom"];
```

# Tipus d'objectes

## Objectes del llenguatge:

- Object, Boolean, Number, Math, Date, String, Array, RegExp

## Objectes del navegador:

- Window, Navigator, Screen, Location, History, Timing, Cookies

## Objectes del DOM:

- Core DOM:
  - Node, NodeList, NameNodeMap, Document, Element, Attr
- HTML DOM:
  - Document, Event, Elements
  - Anchor, Area, Base, Body, Button, Form, Frame, Frameset, Image, Input, Link, Meta, Object, Option, Select, Style, Table, Textarea

## Objectes definits per l'usuari

# Objectes del llenguatge

Donen suport per manejar tipus bàsics.

Tots els objectes del llenguatge tenen les propietats:

- **Constructor**
- **Prototype**
  - És un constructor que permet afegir propietats i mètodes al objecte.
  - S'aplicarà a tots els objectes del tipus.
    - `Boolean.prototype.noavFuncio=function(){`
    - `// codi`
    - `}`

I els mètodes:

- **toString():** Retorna una representació com string del objecte.
- **valueOf:** Retorna el valor primitiu (true/false, un numero, etc.) del objecte.

# Exercici d'investigació

Explica en un PDF quina funció tenen, propietats i mètodes de cada un dels següents objectes nadius del llenguatge JavaScript.

- Boolean
- Number
- Math
- String
- Date
- Arrays
- RegExp
- Propietats i Mètodes Globals



# Exercicis

# Programació amb arrays, funcions i objectes definits per l'usuari

- Funcions predefinides pel llenguatge
- Funcions definides per l'usuari
- Arrays
- Objectes

# Funcions predefinides pel llenguatge