

Tota la documentació ha de tenir portada, cos del treball i bibliografia (si passa d'una pàgina). El document es lliurà sempre en **format pdf**.

Tarea2. Programació Avançada en Java

Entrega

El lliurament serà per mitjà d' un document pdf o en format zip a la plataforma moodle del centre, segons especifiqui el professor. També és possible compartir-ho a través de Google Drive, en aquest cas, el document ha de tenir l'opció de "Compartir amb qualsevol que tingui l'enllaç". NO CAL QUE COMPARTEIXI EL DOCUMENT AMB L' ADREÇA DE CORREU DEL PROFESSOR. S'ha de lliurar el document/s amb la següent nomenclatura: Apellido1_Nombre_Task2. En cas d' especificar-se una altra nomenclatura s' ha de prioritzar la indicada en l' activitat.

1. Qualsevol pràctica que estigui copiada d' un company o d' Internet **automàticament en tindrà un 0**.
2. El lliurament s' ha de lliurar dins del límit temporal fixat, si no és així i sense tenir una deguda justificació **tindrà automàticament un 0** en la nota de l' esmentada pràctica.
3. Qui no lliuri el document amb el format que es demana (nom i estil) **tindrà una penalització del 25%** de la nota final que obtingui.

Valoració

Es valorarà a nivell de presentació:

- El format
- L' ortografia
- La informació aportada

Es valoraran les activitats de la següent manera:

1. Activitat 1: 10%
2. Activitat 2: 10%
3. Activitat 3: 10%
4. Activitat 4: 10%
5. Activitat 5: 15%
6. Activitat 6: 15%
7. Activitat 7: 20%
8. Activitat 8: 10%

Es valorarà a nivell de preguntes:

- Resultats
- Explicació

Activitats

S'han d' adjuntar tots els elements en un arxiu amb la nomenclatura:

Apellido1_Apellido2_Nombre_T2Practica.zip.

Internament s' han de distingir el codic segons l'activitat, per exemple: a l' activitat 1 → T2Activitat1.* (definim l'extensió segons correspongue).

1. Introduir el preu de varios articles, mínim deu sol·licitar per el input un total de 5 articles i fixarà com a punt de parada del input per introduir valors el 0 o el "". Calcular l'IVA i el preu total (amb l'import inclòs) que ha de sortir arrodonit a dos decimals.

PD: Haureu d'utilitzar una funció.

2. Introduir un mes i dir a que pertany:

Ex: Dime un mes: gener

Som a Hivern

(Nota: No cal cap comprovació. Els mesos desembre, gener, febrer seran hivern, març, abril i maig seran primavera, juny, juliol i agost seran estiu i setembre, octubre i novembre serà tardor). Utilitzar l' estructura switch per resoldre d' aquest exercici.

3. Calcula el sumatori i el productori de dos números a y b. Primer, l'usuari deu introduir a i b. Després, si $a > b$ mostrar alerta "Error $a > b$ ". En cas contrari, calcula el sumatori $a + (a+1) + \dots + b$. Mostra el sumatori amb un alert. Després calcula el productori $a * (a+1) * \dots * b$. Si el productori parcial, quan anem calculant, excedeix el valor 100, deixa de calcular mostra eixe productori parcial i indica amb un alert que eixe valor (posseeu el valor amb un número) es major que 100. En cas contrario, mostra amb un alert el productori.

4. Crear un programa que ens permeti redimensionar la pantalla amb el mètode **RESIZE** a 500x500 en cas que sigui Google Chrome, si és un altre navegador que surti el missatge "Operació no permesa amb aquest navegador".

5. Crear una calculadora: Crea un objecte calculator amb tres mètodes:

- read() demana dos valors dos valors per pantalla i els emmagatzema com a propietats d'objecte amb noms a i b.

- `sum()` retorna la suma dels valors emmagatzemats.
- `mul()` multiplica els valors emmagatzemats i retorna el resultat.
- `rest()` resta els valor emmagatzemats i retorna el resultat.
- `divi()` divideix els valor emmagatzemats i retorna el resultat.
- `Modu()` el mòdul dels valor emmagatzemats i retorna el resultat.

PD: El codi demanarà als usuaris dos valors a i b. Després demana operació a fer ("sum", "mul", "resta", "divi", "modul"). Si l'operació introduïda no es cap d'elles, mostra alert amb error. Si no, calcula l'operació i mostra resultat amb un alert.

6. Suma de números ingressats: Escriu una funció `sumInput()` que:

- Demano a l'usuari valors usant `prompt` i els emmagatzemo a l'array.
- Acaba de demanar-los quan l'usuari ingressi un valor no numèric, una cadena buida, o pressioni "Escape".
- Calculi i retorni la suma dels ítems de l'array.

P.D. Un zero 0 és un nombre vàlid, per favor no atures els ingressos amb el zero.

7. Crea una calculadora extensible: Crea una funció `Calculator` que creï objectes calculadors "extensibles". L'activitat consisteix de dues parts.

- a) Primer, implementar el mètode d'actuació que pren un string com a "1 + 2" en el format "NUMERO operador NUMERO" (delimitat per espais) i retorna el resultat. Ha d'entendre més + i menys - Exemple d'ús:

```
let calc = new Calculator;
alert( calc.calculate("3 + 7") ); 10
```

- b) Després agrega el mètode **`addMethod(name, func)`** que ensenyi a la calculadora una nova operació. Pren l'operador `name` i la funció amb dos arguments `func(a,b)` que l'implementa. Per exemple, vam afegir la multiplicació *, division / i potència **:

```
let powerCalc = new Calculator;
powerCalc.addMethod("*", (a, b) => a * b);
powerCalc.addMethod("/", (a, b) => a / b);
powerCalc.addMethod("**", (a, b) => a ** b);

let result = powerCalc.calculate("2 ** 3");
alert( result ); 8
```

Sense parèntesi ni expressions complexes en aquesta tasca. Els números i l'operador han d'estar delimitats per exactament un espai. Hi pot haver maneig d'errors si volguéssim afegir-ho.

8. Barallar un arrelament: S'han d'introduir números per pantalla que s'emmagatzemaran en un arrelament, s'introduiran números fins que s'entri el número 0 que aturarà la sol·licitud de números.

Escriu la funció **`shuffle(array)`** que baratge (reordene de forma aleatòria) els elements de l'array. Múltiples execucions de `shuffle` pot conduir a diferents ordres d'elements. Per exemple:

```
let arr = [1, 2, 3];
shuffle(arr); arr = [3, 2, 1]
```

```
shuffle(arr); arr = [2, 1, 3]
```

```
shuffle(arr); arr = [3, 1, 2]
```

Tots els reordenaments d' elements han de tenir la mateixa probabilitat. Per exemple, [1,2,3] pot ser reordenat com [1,2,3] o [1,3,2] o [3,1,2] etc, amb igual probabilitat en cada cas.

Una vegada reordenat en un nou array es deu mostrar aquest

Anexo

En aquest tema es veuen les funcions, es parla de l'àmbit de les variables, de les sentències break i d'altres estructures de control (while,do... while,switch). Us incloc un nou manual que a més de servir-vos d'ampliació per al vist fins ara, us servirà de referència per veure alguns mètodes i propietats d'aquests objectes. També veiem amb més profunditat l'objecte Date , ja que és un dels més utilitzats en les aplicacions de JavaScript.

Programació Avançada

Com he comentat anteriorment, les funcions , les sentències break i continue i les estructures de control: while, do... while i switch. Aquestes estructures són senzilles d'utilitzar i estan ben explicades en el tema. Pel que fa a les funcions, sabem com s'utilitzen, a veure com anomenar-les. Existeixen diferents maneres de cridar a una funció:

- **Des de Javascript:** El més correcte és crear una funció a la capçalera del programa, així ens assegurem que quan carreguem la pàgina aquest ja accessible. La crida a la funció estarà en el cos del programa dins de les etiquetes <script> i </script>. Entre les etiquetes hem de posar el nom de la funció seguit del parèntesi i dins d'aquests, si calen, els arguments per executar-la.

```
<html>
<head>
<title>ejemplo llamada función </title>
<script type="text/javascript">
function nom_fun([argumentos]) {
// instrucciones a realizar
}
</script>
</head>
<body>
<script type="text/javascript">
nom_fun([argumentos])
</script>
</body>
</html>
```

- **Des d'HTML:** La crida a la funció la realitzem com si fos un atribut d'HTML. És habitual utilitzar aquest sistema quan es produeixen els esdeveniments onload, onclick, etc.

```
<html>
<head>
<title>ejemplo llamada función </title>
<script type="text/javascript">
function nom_fun([argumentos]) {
// instrucciones a realizar
}
</script>
</head>
<body onload="nom_fun([argumentos])">
</body>
</html>
```

L'àmbit de les variables, el llegiu detingudament doncs sol ser font d'alguns errors quan es treballa amb funcions ja que si no està accessible la variable de la funció, tal com està explicat en els exemples, no es podrà executar correctament. Com a ampliació al tema us adjunto un altre exemple de l'estructura

switch que pot ser útil en el cas que diversos valors hagin d'executar les mateixes instruccions.

```
<html>
<head>
<title>ejemplo de javascript </title>
<script type="text/javascript">
  var dia=prompt("dia de la semana:", "");
  switch(parseInt(dia)){
    case 1:
    case 2:
    case 3:
    case 4:
    case 5:
      document.write("Es dia laboral");
      break;
    case 6:
    case 7:
      document.write("Es fin de semana");
      break;
    default:
      document.write("No has introducido un número de dia correcto");
      break;
  }
</script>
</head>
<body>
  <p>La página que estás viendo es un ejemplo de la estructura Switch
</body>
</html>
```

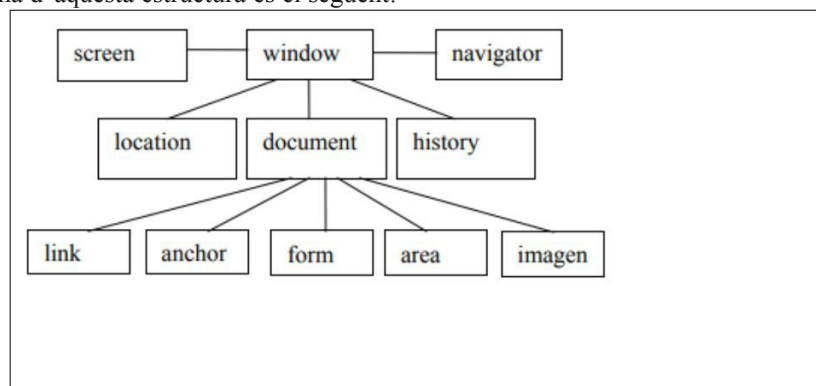
L'expressió entre parèntesis que avalua ha de ser sencera. Si per contra el que avaluem és una cadena, recordant posar-la entre cometes:

```
<html>
<head>
<title>ejemplo de javascript </title>
<script type="text/javascript">
  var dia=prompt("dia de la semana:", "");
  switch(dia){
    case "lunes":
    case "martes":
    case "miércoles":
    case "jueves":
    case "viernes":
      document.write("Es dia laboral");
      break;
    case "sábado":
    case "domingo":
      document.write("Es fin de semana");
      break;
    default:
      document.write("No has introducido un número de dia correcto");
      break;
  }
</script>
</head>
```

```
<body>
  <p>La p gina que est s viendo es un ejemplo de la estructura Switch
</body>
</html>
```

JavaScript encara que no sigui un llenguatge orientat a objectes com pot ser Java, si que treballa amb objectes. Realment ja treballem amb objectes en la pr ctica anterior, la classe String, la classe Array, en realitat s n classes de JavaScript. Tenen m todes, propietats, etc. Aquests s n objectes predefinits de JavaScript per  a banda d'aquests objectes tenim els objectes predefinits de navegador: window, document, history, navigator i screen, amb els seus m todes i propietats, que s n els que ens han de permetre realitzar aplicacions m s completes en aquest llenguatge. Com a refer ncia per a l'estudi d'aquests objectes deveu llegir el punt 15 (Objectes predefinits del navegador) del nou tutorial que us heu penjat, veureu els m todes i propietats de cadascun d'ells, no obstant aix  a Internet podeu obtenir molta m s informaci  sobre cadascun d'ells.

Els navegadors accedeixen als objectes seguint una estructura jer rquica en la qual existeixen nivells. L' esquema d' aquesta estructura  s el seg ent:



L' objecte principal de la jerarquia  s l' objecte window que representa una finestra del navegador. Es considera un objecte impl cit i no cal nomenar-lo per accedir als objectes que es troben en un nivell inferior. Per exemple, si volem accedir a la propietat que canvia el color de fons d'una p gina podem fer-ho d'aquesta manera: window.document.- per  tamb  podem fer-ho escrivint document.bgcolor.

L' objecte navigator identifica les caracter stiques de la plataforma sobre la qual est  executant-se l' aplicaci  i l' objecte screen correspon a la pantalla que est  utilitzant l' usuari.

L' objecte document es refereix als documents que es carreguen a la finestra del navegador. Possiblement sigui un dels objectes m s utilitzats en la programaci  d' aquest llenguatge, ja que en depenen alhora altres objectes molt importants com s n els vincles, punts d' ancoratge, els formularis o les imatges.

L'objecte history emmagatzema les refer ncies de tots els llocs web visitats i l'objecte location correspon a la URL de la p gina en  s. screen window navigator Location location document history link anchor form area imagen.

Algunes propietats i m todes de l' objecte location, no incl s en el manual, s n els seg ents:

Propietats:

- hash: Mostra la cadena que representa l' ancoratge de la URL.
- host: Mostra la cadena que representa el nom del domini del servidor i el n mero del port dins la URL.
- hostname:Mostra la cadena que representa el nom del domini del servidor dins de la URL.
- href:Mostra la URL completa.

- pathname:Mostra la cadena que segueix al nom del servidor.
- port:Mostra el número del port de la URL.
- protocol:Mostra el protocol utilitzat per la pàgina
- search:Mostra la cadena de recerca que es mostra després del signe d'interrogació a la URL

Mètodes

- assign: Carrega un nou document.
- reload(): Càrrega de nou el document actual.
- replace(): Canvia la URL del document actual per una altra URL

Vegem un exemple amb alguns dels objectes que hem vist:

```
<head>
<title>ejemplo de javascript </title>
<script type="text/javascript">
Document.write("Tu Navegador es: "+navigator.appName + "<br>");
if (navigator.javaEnabled()){
document.write("Tu navegador soporta los Applets de Java"+"<br>");
}
else {
document.write("Tu navegador no soporta los Applets de Java"+"<br>");
}
document.write("La altura de tu pantalla es: "+screen.height+"<br>");
document.write("La anchura de tu pantalla es: "+screen.width+"<br>");
document.write("La URL completa es:"+location.href+"<br>");
</script>
</head>
<body>
<noscript>
<p>La página que estás viendo requiere para su funcionamiento el uso de JavaScript
</noscript>
<form>
<input type="button" value="WIKIPEDIA" onclick="location.assign('http://Wikipedia.es');">
</form>
</body>
</html>
```

El resultat de l'execució d'aquest exemple seria el següent:

- El teu Navegador és: Chrome
- El teu navegador no suporta els Applets de Java
- L'alçada de la teva pantalla és: 600
- L'amplada de la teva pantalla és: 800
- La URL completa és:
file:///C:/Documents%20and%20Settings/Mati/Escritorio/JAVASCRIPT/EJERCICIOS/pruebasobjetosdelnavegador.html

WIKIPEDIA → (En fer click sobre el botó es carrega la pàgina de la wikipedia).

Per acabar comentar que el mètode document.write no es limita només a escriure un text tal com hem vist fins ara. Podem utilitzar-lo per crear i manipular tota mena d'objectes. Això és una pràctica habitual a les pàgines web que utilitzen finestres emergents.

Ejemplo:

```
<script type="text/javascript">
var nuevaVentana;
function crearnueva() {
nuevaVentana=window.open("", "pagina2",
"toolbar=yes,location=no,menubar=yes"+
"width=300,height=300" );
nuevaVentana.document.write("<HTML><HEAD><TITLE>" +
"Sin Título</TITLE></HEAD>\n");
nuevaVentana.document.write("<BODY><form>\n");
nuevaVentana.document.write("<input type='button' " +
"value='Cerrar' onClick='window.close();'>\n");
nuevaVentana.document.write("</form>\n");
nuevaVentana.document.write("</BODY></HTML>\n");
}
</script>
```