**Reporte de prácticas 2**

Fernando Julián Verján Vargas 2E

**Introducción**

En la práctica numero 2 empezamos sincronizando nuestra computadora personal con el equipo del laboratorio de telemática mediante Google Drive, para tener disponibles todos los archivos que realicemos en clase. También reflexionamos algunos conceptos básicos de matemáticas, como el neutro aditivo, que es el 0, y el neutro multiplicativo, que es el 1. Realizamos algunos programas usando while, do, for, y empezamos a usar las funciones en otros programas.

**Desarrollo**

Ya que realizamos varios programas usando ciclos, es importante definir las diferencias entre cada uno de ellos:

While: Primero se evalúa la condición, si esta se cumple, se ejecutan las instrucciones, una vez hecho esto se vuelve a verificar la condición. Este proceso se repite hasta que la condición sea falsa y no se ejecutará el bucle, y después continuará con la ejecución del resto de código.

Do while: Primero se ejecutan las instrucciones y luego se verifica la condición (Por lo tanto siempre se ejecuta al menos una vez), en caso sea verdadera, se ejecuta nuevamente el cuerpo. Este proceso se sigue llevando a cabo siempre y cuando la condición sea verdadera.

For: La principal diferencia entre este ciclo y los dos primeros es que este se usa cuando ya se conocen las veces que se va a repetir y en los anteriores no se conoce el número de repeticiones.

Una vez entendido esto, realizamos programas que nos imprimieran los números del 1 al 10, hicimos programas distintos para usar cada uno de los ciclos. A manera de ejercicios, también hicimos programas similares, pero hicimos uno que imprimiera del 1 al 5 (Con cada uno de los ciclos), y después otro que mostrara del 5 al 9.

Posteriormente se evaluaron algunas de las características de los REPL y Runner

|  |  |
| --- | --- |
| REPL | Runner |
| -No usa document.write. | -No ocupa <HTML> |
| -Utiliza Print. | -No ocupa Script |
| No usa console.log. | -Si puede usar alert |
| No usa alert. | -Usa print |
|  | -No utiliza document.write |

En seguida, revisamos contenido bibliográfico que el maestro nos presentó, hicimos uso de algunos para obtener códigos fuente. Creamos el runner “Louse.js”, sacado del libro Wilton, capítulo 2.

También utilizamos el libro PowSch02, Capítulo 2, sección 4. De aquí obtuvimos algunos programas:

-typeof (Para conocer el tipo de una variable, ya que JS es de tipado dinámico y no se especifica al declarar variables)

-typeof3 (Similar al anterior, obteniendo diferentes tipos de variable)

-boolean (Similar al primero, introducimos “false” para obtener una variable tipo boolean)

-navigator (Imprimimos la versión del navegador, utilizando navigator.appName)

Después de desarollar estos programas, nos enseñaron otra manera de escribir nuestro código, en este caso utilizamos aquamacs, una cosa importante a resaltar es que al seleccionar la pestaña File/Save buffer as y poner un nombre seguido de js (Ejemplo: Aquamacs.js) El programa nos reconocerá el texto como código y le dará colores y un mejor acomodo.

Estudiamos un poco de teoría para entender el funcionamiento de las funciones, y poder implementarlas más tarde en programas:

* Una función es un componente de software que a su vez puede convocar otra función. Cuando terminan su trabajo se termina la última función, después la anterior y así hasta regresar al sistema operativo.
* Una función regresa un valor, un procedimiento no.
* Scope se refiere al ámbito (Variables declaradas en funciones más recientes no serán conocidas en funciones declaras antes. Sin embargo, variables declaradas las primeras funciones si serán conocidas en funciones mas recientes)
* Exceptions: Cuando ocurre una excepción termina el programa (Por ejemplo un error)

Usando las funciones, desarrollamos programas simples, como suma, resta, multiplicación y división. También hicimos una función para validar que los datos insertados en el programa sean números, y la implementamos en los programas simples anteriormente mencionados.

Por último realizamos un programa que explica mejor el Scope (o ámbito), con variables declaradas en diferentes niveles de funciones.

**Conclusiones**

Aprendimos a usar los ciclos, según sea necesario el caso, entendiendo que en el ciclo while podría ser que no se ejecute nunca, ya que primero se evalúa la condición y luego entra a las instrucciones, mientras que do while siempre se va a ejecutar, por lo menos una vez. Las funciones son utilizadas como pequeñas partes de código que se pueden invocar después según sean requeridas. Es importante resaltar que un procedimiento no regresa un valor, una función si, y por lo tanto en las funciones se usa “return”. Aprendimos más características de los REPL y Runner y ahora ya tenemos bibliografía para estudiar más y de esta manera poder sacarle provecho.

<https://www.quierotec.com/diferencia-entre-los-ciclos-while-do-while-y-for/>