**PRO-C42**

Tema: Revisión de JavaScript

**Descripción:** Los alumnos revisan los conceptos cubiertos en las primeras 40 clases del curso: Variables, Números, Cadenas, Booleanos, Matrices, Funciones, Objetos, etc..

**Objetivos:** ● Repasar todos los conceptos cubiertos en el curso hasta ahora. ●Utilizar la consola para ejecutar comandos de Javascript.

¡Hola, <nombre del alumno>! Hemos recorrido un largo camino, desde la creación de un simple juego de Pong, hasta un juego de carreras de autos multijugador usando bases de datos en 40 clases, solo 40 horas. ¿Cómo ha sido tu viaje?

Las clases 41 a 48 serán muy especiales. Estarás creando tu propio proyecto dorado, tu propio juego, a través de estas clases. También lo publicarás en la tienda de aplicaciones.

*Además, si tu proyecto es innovador y le gusta a nuestro equipo (1% superior), te proporcionaremos un programa de mentores de Silicon Valley en EE. UU. Silicon Valley, es el lugar donde se han fundado las principales empresas de tecnología innovadora: Google, Facebook, Apple, Amazon, Microsoft, Tesla. En caso de que tu proyecto sea seleccionado, tendrás la oportunidad de visitar a los mejores emprendedores allí y obtener orientación y tutoría para comenzar con tu propia idea de emprendimiento.*

En la clase de hoy, intentaremos repasar y practicar todos los conceptos de Javascript que hemos cubierto en las primeras 40 clases.

Intentaremos volver a visitar los juegos que hicimos y tratar de recordar los conceptos que aprendimos a través de los juegos.

El alumno, abre el código para diferentes juegos e intenta recordar los diferentes conceptos de juego, y conceptos de javascript cubiertos en la clase hasta ahora en diferentes juegos:

Juego de Pong: variables, almacenamiento de números, cadenas, programación condicional (if-else), bucle for, funciones , estado del juego, etc.

T-Rex: Creación de suelo de desplazamiento infinito, matrices, declaraciones de conmutación, biblioteca,

escritura de biblioteca propia, DRY, etc.

Angry Birds: clases, objetos, motor físico, llamada a una API, legibilidad del código, escritura del código estructurado, etc.

Juego de carreras de autos: bases de datos, lectura y escritura en una base de datos, etc.

Estupendo. Es sorprendente, lo mucho que podemos hacer con estos conceptos combinados.

Hoy revisaremos todos los conceptos de Javascript escribiendo comandos en la consola JS.

Pídele al alumno que abra una ventana de consola en el navegador. ¿Sabías que puedes escribir comandos javascript en la consola?. Tan pronto como escribas un comando aquí y presiones enter - *entrar*, puedes ver el resultado de tu programa aquí.

Los desarrolladores de Javascript, usan la consola para probar rápidamente sus instrucciones. Usaremos la consola para probar y revisar los conceptos en Javascript.

*# JavaScript Review*

***## Numbers and Strings***

*JavaScript can understand different kinds of data like numbers and string.*

*Try entering a number in the console:*

*> 2*

*Similarly, try entering any string in the console:*

*> "Hello world"*

***## Variables***

*You can store data inside containers (variables) using "var". Can you store your name inside a variable?*

*> var name = "Rajeev"*

*Now call the variable to see what is inside it.*

*> name*

*You can see your name printed in the console. Strings are objects in JavaScript and they have some properties and functions defined. You can try some of them.*

*> name.length*

*> name.toUpperCase()*

*You will be able to see your name in all caps.You can also store any number inside the variable.*

*> var num = 5*

*You can also convert a string into a number:*

*> parseInt("123")*

***## Arithmetic Operations***

*You can use arithmetic operations on numbers using arithmetic operators (+ , - , /, \*, %).*

*You can try doing different arithmetic operations on the console.*

*> 2+3*

*Try using '+' arithmetic operator on strings and guess what will happen:*

*> "Hi" + "Friend"*

*Strings get joined together when you add them.*

*Now try using any other arithmetic operation on strings like division (/)*

*> "a" / "b"*

*You will get \*\*NaN\*\* which means not a number.*

*Now try dividing any number with 0:*

*> 1/0*

*You will get \*\*Infinity\*\* as an answer.*

***## Booleans***

*There is another boolean data type in JavaScript which can hold only \*\*true\*\* or \*\*false\*\**

*> var bool = true*

*Comparison operators (>,<,>=,<=,===, !=) also evaluate to a boolean value. Try to use any of the comparison operators with numbers on the console*

*> 2===3*

*This will print false on the console.*

***## Other data types***

*There are two other kinds of data in JavaScript - \*\*null\*\* and \*\*undefined\*\**

*\*\*null\*\* is used when you want a variable to hold nothing.*

*\*\*undefined\*\* is the value inside a variable when you have forgot to assign anything to it.*

*> var test*

*Now try to see what is inside 'test'*

*> test*

***## Conditional programming***

*We can use if-else block to create conditional programming.*

*Try writing a simple if-else statement:*

*> if(3>2){console.log("Happy")}else{console.log("Not Happy")}*

*If the condition evaluates to \*\*true\*\*, if block is executed; otherwise else block is executed.*

*Switch statement is another way in which we can do conditional programming.*

*> var name = "My name"; switch(name) { case "My name": console.log("Condition 1"); break; case "my Name": console.log("Condition 2"); break; default: console.log("None of the conditions are true");}*

***## Loops***

*Computers do not like to repeat themselves. We use loops to perform repeated function. There are two kinds of loops - \*\*for\*\* and \*\*while\*\* loop.*

*Write a simple \*\*for\*\* loop:*

*> for(var i=0; i<=5; ++){ console.log(i) }*

*Write a simple \*\*while\*\* loop:*

*> var i =0 while (i<=5){ console.log(i) i=i+1 }*

***## Arrays***

*Arrays are a type of data structure where you can use single variable to store a list of items.*

*Write an array which stores a list of items.*

*> var friends = ["friend1","friend2","friend3"]*

*You can access any item in the list using indexes.*

*> friends[0]*

*You can also loop over all the items using \*\*for-each\*\* loop*

*> for(var index in friends){ console.log(friends[index]) }*

*Arrays are also objects in javaScript. They have some properties and function defined.*

*For example, you can get the length of the array using length property:*

*> friends.length*

*You can also push new elements into the array using push():*

*> friends.push("friend4")*

*You can see the elements in the friends array now:*

*> friends*

*You can also pop out the last element from the array using pop():*

*> friends.pop();*

***## Functions***

*Javascript has certain in-built functions which you can use.*

*You can also write your own functions. Write a function to calculate the circumference of a circle. It should take radius as the argument.*

*> function circumference(radius){ var circumference = 2 \* 3.14 \* radius*

*return circumference }*

*Now you can use the circumference functions to calculate the radius of any circle.*

*> circumference(5)*

***## Objects***

*We used classes to design blue prints of objects in javascript and then used \*\*new\*\* to create new object using the class. Internally JavaScript creates a new object using \*new Object()\**

*> var paddle = new Object();*

*You can assign new properties and functions to test object. Ask the student to assign a new property and function to a test object*

*> paddle.length = 60*

*> paddle.showLength = function(){ console.log(paddle.length); }*

*Call paddle.showLength() to see the length of the paddle.*

*> paddle.showLength()*

*Call paddle object to see what's stored inside the paddle.*

*> paddle*

**Conclusión:** Intentamos cubrir los conceptos más importantes en Javascript. Juntos, hemos combinado estos conceptos, para crear desde juegos muy simples hasta complejos. Esta clase fue una revisión de todos los conceptos técnicos cubiertos en nuestras lecciones hasta ahora.

¿Esta revisión te ayudó?

En la próxima clase, intentaremos aprender los aspectos de diseño de los juegos:

- ¿Cómo se diseñan los juegos?

- ¿Qué hace que los juegos sean

divertidos?

- ¿Cómo los juegos cuentan una

historia?

¡A través de esto, estarás diseñando tu propio juego!

**Desafío:** El mejor hombre asciende

**Objetivo del proyecto:** En la clase 42, hemos revisado todos los conceptos técnicos que se aprendieron usando JS, y lo practicamos en la ventana de la consola.

En este proyecto, implementarás algunos de esos conceptos para crear un efecto de día lluvioso.

**Historia:** El joven Brad caminaba por las calles. Mientras caminaba pensaba en Batman. Era una noche oscura y lluviosa cuando de repente vio un murciélago volando.

Recordó su miedo a los murciélagos y se le ocurrió una idea. Decidió que se disfrazará y se llamará El Mejor

Hombre. Brad quiere que retrates su noche oscura y lluviosa, ¡para que pueda quedársela como un recordatorio del día en que decidió convertirse en El Mejor Hombre!