PROGRAMACIÓN WEB
teoría/práctica

GUÍA BÁSICA DE JAVASCRIPT, parte II

Guía Básica de Javascript, parte II

Complejidad de esta currícula: alta
Tiempo aproximado de desarrollo: de (2-4) encuentros.

Qué necesitás saber?

- HTML Básico
- JAVASCRIPT Básico/intermedio (parte I)

Temario de esta currícula

1. Comentarios en Javascript.
2. Crear un archivo html con Javascript embebido.
3. Funciones de usuario. Parámetros.
4. Funciones de Javascript.
5. Incremento y decremento.

1er Paso: Crear archivo html

- 1. Vamos a crear un archivo html y allí vamos a "embeber" Javascript. Recordá que en la currícula de Javascript parte 1, utilizábamos el método de llamar a un archivo externo de Javascript.

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title>SUMA DE NUMEROS</title>
    </head>
    <body>
        <h1>Este es un programa que calcula la suma de dos números</h1>

        <script type="text/javascript">

        </script>

    </body>
</html>
```

Comentarios en Javascript

- Es muy útil cuando escribimos código, realizar "comentarios" no solo para que nosotras mismas entendamos que escribimos sino también para que otras programadoras cuando los lean, puedan entender.

- /* Los comentarios son como las notas al margen del papel que hacemos cuando realizamos algún tipo de tarea y en general son anotaciones importantes así no
- olvidamos ciertas cosas.*/

- Todos los lenguajes de programación nos dan la posibilidad de hacer esto. La forma en que hacemos comentarios puede variar de lenguaje en lenguaje.

- En Javascript lo hacemos así:
  - Para comentar en una sola línea:
  - Para comentarios de varias líneas:

```
// a continuación se muestra un mensaje alert("mensaje de prueba");
```

```
/* Los comentarios de varias líneas son muy útiles
cuando se necesita incluir bastante información en los comentarios */
```

Funciones del usuario

- Funciones es algo que vimos en matemáticas… pero...a no asustarse! En programación lo hacemos un poco diferente.

- En programación consideramos que una función es un pedacito de código que tiene un propósito definido. A esta función la podemos "llamar" desde cualquier parte de nuestro código para que nos arroje el resultado tan especial que esperamos.

- A una función también podemos enviarle datos para que dentro de la misma se realice un proceso. Por ejemplo: una función que calcule a partir de dos números que ingrese el usuario, la suma.

- Las ventajas de utilizar funciones es que solamente tendríamos que escribirla una sola vez y usarla las veces que sea necesario! De ésta forma escribimos código más prolijo y menos extenso.

Funciones

- Supongamos que nosotros somos el usuario, entonces decidimos "mandar" a la función los valores 3 y 6 para que los sume.
- La función tiene que devolvernos el resultado de esa suma.
- Como característica principal, las funciones tienen que tener nombre (así sabremos cómo llamarlas cuando las necesitamos)
- En este caso, nuestra función se llamará: **sumarDosNumeros**
- Y en Javascript lo podemos escribir así:

```
function sumarDosNumeros(3,6) {
  resultado = 3 + 6;
  return resultado;
}
```

- Esta función lo único que podría sumar es SIEMPRE 3 y 6

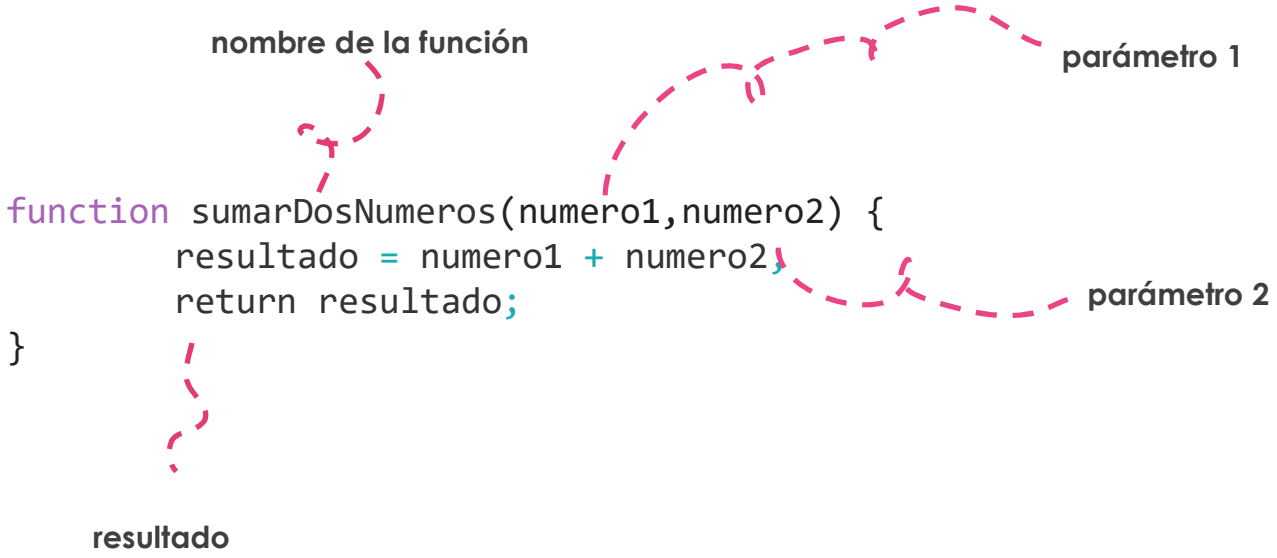
Funciones

- Qué pasaría si pudiéramos utilizar la función no solamente para sumar el 3 y el 6, sino cualquier número?
- Entonces tendríamos que usar PARÁMETROS.
- Parámetro: es una variable o un dato que se utiliza como "entrada" a las funciones.

- Cuando la función nos devuelva un resultado, ese resultado será un valor. Ese valor puede ser almacenado en una variable!

Funciones

- Una función que reciba cualquier número, podríamos plantearla así:



Funciones

- Para utilizar este tipo de funciones los pasos serían los siguientes:
  - Declarar variables con los números que nos gustaría sumar (u obtenerlos desde otro lado)
  - Llamar a la función, pasándole los parámetros
  - Almacenar el resultado de la función en una variable para después poder mostrarla
- Como se guarda el valor de la función? Simplemente:

```
var resultado = sumarDosNumeros(numero1,numero2);
alert( resultado );
```

2do

- 1. Y … a escribir Jav … ngan dos núr …

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title>SUMA DE NUMEROS</title>
    </head>
    <body>
        <h1>Este es un programa que calcula la suma de dos números</h1>

        <script type="text/javascript">
            // declaramos las dos variables con los dos números a sumar
            var numero1 = 3;
            var numero2 = 6;

        </script>

    </body>
</html>
```

2do Paso: Programamos

- 2. Ahora que ya sabemos qué es una función y como se usa, la agregamos al código? Y recordá almacenar el valor de la función en la variable resultado

```
<script type="text/javascript">
      // declaramos las dos variables con los dos números a sumar
      var numero1 = 3;
      var numero2 = 6;

      var resultado = sumarDosNumeros(numero1,numero2);
      alert( resultado );

      function sumarDosNumeros(numero1,numero2) {
               resultado = numero1 + numero2;
               return resultado;
      }

 </script>
```

Funciones

- Algo muy importante: no importa en qué lugar esté ubicada la función sumarDosNumeros. Lo importante es desde donde se la quiera llamar a dicha función.

```
<script type="text/javascript">
      function sumarDosNumeros(numero1,numero2) {
               resultado = numero1 + numero2;
               return resultado;
      }

      // declaramos las dos variables con los dos números a sumar
      var numero1 = 3;
      var numero2 = 6;

      var resultado = sumarDosNumeros(numero1,numero2);
      alert( resultado );

 </script>
```

Funciones

- Y si quisiéramos sumar otros números y no solamente el 3 y el 6? Bueno una opción podría ser esta...

```
<script type="text/javascript">
	function sumarDosNumeros(numero1,numero2) {
		resultado = numero1 + numero2;
		return resultado;
	}

	// declaramos las dos variables con los dos números a sumar
	var numero1 = 3;
	var numero2 = 6;

	var resultado = sumarDosNumeros(numero1,numero2);
	alert( resultado );

	numero1 = 2;
	numero2 = 3;

	var resultado = sumarDosNumeros(numero1,numero2);
	alert( resultado );

 </script>
```

Funciones de Javascript

- Recién veíamos cómo era posible que nosotras podamos crear nuestras propias funciones.
- Una de las ventajas que tiene Javascript es que ya trae por defecto, funciones que están hechas y que solo tenemos que utilizarlas llamándolas por su nombre. Por ejemplo: alert("hola"), prompt("Ingrese su edad") y otras más.
- Para llamar a una función, sea de Javascript o creada por nosotras, siempre tenemos que utilizar los paréntesis luego del nombre: alert();
- Las funciones de Javascript pueden usarse para diferentes propósitos, por ejemplo: saber cuántos caracteres tiene una palabra, cuántos elementos tiene un arreglo (array), para unir palabras, para pasar de un tipo de dato a otro, etc…
- Veamos las más frecuentes:

Funciones de Javascript

Función	Uso	
length	var mensaje = "hola" mensaje.length	Obtiene la cantidad de caracteres de "hola". Se usa para arrays también
+	"hola"+"chau"	Concatena (une) dos cadenas
toUpperCase()	mensaje.toUpperCase();	Pasa el mensaje a mayúsculas
toLowerCase()	mensaje.toLowerCase();	Pasa el mensaje a minúsculas
charAt()	mensaje.charAt(0)	Obtiene el caracter indicado por el número.
substring(inicio,final)	var mensaje = "hola mundo"; mensaje.substring(2);	Extrae una porción del texto. En este caso devuelve "la Mundo". El parámetro final es opcional
split()	var mensaje = "hola mundo como estan"; mensaje.split(" ");	Convierte una cadena de texto en un array. En este caso devuelve un array ["Hola", "mundo"]
parseInt()	parseInt("10");	Convierte la cadena "10" en un número entero

Funciones de Javascript

- Consideremos la función que venimos trabajando. Qué pasaría si en vez de precargar los números a sumar, preguntamos al usuario y que los ingrese por pantalla?

- Podríamos utilizar la función de Javascript Prompt.

```
// utilizamos la función prompt para pedir los datos de entrada al usuario
var numero1 = prompt("Ingresá el primer número para la suma");
var numero2 = prompt("Ingresá el segundo número para la suma");
```

- Si intentamos sumar ambos números ingresados, notaremos que en lugar de sumar los números, los concatena...por ejemplo, si ingresamos 2 y 3, el resultado de numero1+numero2 sería 23. Esto ocurre porque la función prompt devuelve el número como si fuera una cadena. Para poder operar matemáticamente con estos datos ingresados por el usuario, es necesario convertirlos a números enteros.

Funciones de Javascript

- Para seguir haciendo más interesante esta cuestión, veremos que las funciones de Javascript también se pueden anidar.
- Esto nos sirve para no utilizar tantas variables.
- Tomando nuestro ejemplo:

```
// utilizamos la función prompt para pedir los datos de entrada al usuario
var numero1 = parseInt(prompt("Ingresá el primer número para la suma"));
var numero2 = parseInt(prompt("Ingresá el segundo número para la suma"));
```

- Lo cual nos devuelve el número entero que tanto esperamos.

3er Paso: ParseInt()

- 1. Completemos nuestro programita agregando la función de Javascript ParseInt() cuando el usuario ingresa un dato por teclado para la suma. El resultado final tendría que ser algo similar:

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title>SUMA DE NUMEROS</title>
    </head>
    <body>
        <h1>Este es un programa que calcula la suma de dos números</h1>

        <script type="text/javascript">
            // utilizamos la función prompt para pedir los datos de entrada al usuario
            var numero1 = parseInt(prompt("Ingresá el primer número para la suma"));
            var numero2 = parseInt(prompt("Ingresá el segundo número para la suma"));

            var resultado = sumarDosNumeros(numero1,numero2);
            alert( resultado );

            function sumarDosNumeros(numero1,numero2) {
                resultado = numero1 + numero2;
                return resultado;
            }
        </script>

    </body>
</html>
```

Incremento y decremento

- Vimos anteriormente que podemos tener variables numéricas y que con estas variables podemos hacer operaciones.

- Incrementar significa aumentar en una o más unidades un número, por ejemplo, al número 5 incremento 1 unidad y el resultado es 6. Al número 2 incremento 3 unidades y el resultado es 5. En Javascript se lo puede utilizar así:

```
var numero = 5;
++numero;
alert(numero); // numero = 6
```

- Decrementar es lo contrario a incrementar. Significa descontar unidades: Si al número 4 decremento en 1 unidad el resultado es 3.

```
var numero = 5;
--numero;
alert(numero); // numero = 4
```

¿Preguntas?

Desafío

- Probá agregar más funciones a tu programa, más operaciones… por qué no sumas y restas? Multiplicaciones? Lo que quieras!
- Solicitá a tu mentora el archivo de práctica de esta currícula.
- Buena Suerte!

MUCHAS GRACIAS