Programación Web

Clase 3 - Funciones

1. Funciones

¿Qué son?

Son bloques de código a los que les asignamos una tarea.

En general, estas tareas queremos repetirlas algunas veces.

Cada vez que queramos usar el código de la función, la llamamos con su nombre designado.

```
function saludar() {
    alert("Hola!");
}

saludar();
```

Definición

Para definir una función, utilizamos la palabra function, seguida de su nombre.

Sigue las mismas reglas de nombrado que las variables.

Entre paréntesis, podemos pasar info extra que requiramos, estos son los parámetros, separados por comas. Dentro de las llaves, nuestro código.

```
function saludar(saludo) {
    alert(saludo);
}

saludar("Hola!");
saludar("Adios!");
```

Asignar funciones a variables

Otra forma de crear funciones es asignando funciones sin nombre (las llamamos anónimas) a variables.

```
var saludo = function() {
    console.log("Hola!");
}
saludo();
var suma = function(numero1, numero2) {
    console.log(numero1 + numero2);
}
suma(1, 3);
```

return

Podemos hacer que las funciones “escupan” un valor al terminar, como **prompt()**.

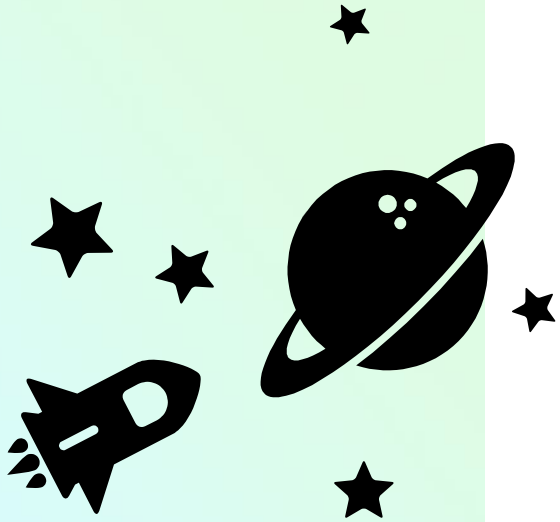
Ese valor podemos asignarlo a variables o pasarlo a otra función.

El **return** termina la función.

```
var num1 = parseInt(prompt("Ingrese un número"));
var num2 = parseInt(prompt("Ingrese otro número"));

function sumar(n, m) {
    return n + m;
}

var resultado = sumar(num1, num2);
console.log(resultado);
```

Actividad

Crear funciones para las cuatro operaciones aritméticas básicas, que tomen dos parámetros cada una y logueen el resultado en la consola.



Múltiples parámetros

Si no sabemos cuántos parámetros va a recibir una función (por ejemplo, si queremos que sume números ilimitados), podemos utilizar arguments.

Arguments es un listado de todos los parámetros que le pasamos a una función.

Podemos recorrerlo con un for y utilizar todos los valores en nuestro código.

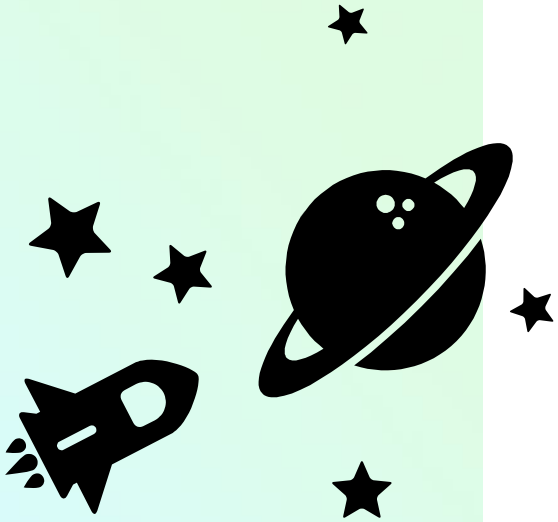
Todas las funciones tienen este listado escondido.

Ejemplo múltiples parámetros

```
function suma() {
    var resultado = 0;

    for (numero in arguments) {
        resultado += arguments[numero];
    }

    return resultado;
}

console.log(suma(5, 4)); // 9
console.log(suma(5, 4, 10, 400)); // 419
```

Actividad

Repetir actividad anterior, haciendo que nuestras funciones de suma, resta y multiplicación tomen múltiples parámetros.



2. Scope

Alcance de variables

- Es el contexto desde el cual podemos acceder a una variable.
- Las variables sólo pueden ser utilizadas dentro del bloque de código en el que han sido definidas.
- Diferenciamos entre variables locales (pueden ser accedidas en un bloque pequeño).
- Y globales (pueden ser accedidas en todo el documento).

Ejemplos de scope

- Local

```
function myFunction() {
    var a = 4;
    return a * a;
}
console.log(a); //undefined
```

- Global

```
var a = 4;
function myFunction() {
    return a * a;
}
```

3.
Funciones útiles

confirm()

- Al igual que el prompt(), nos deja recibir información del usuario. En este caso, una respuesta **true** o **false** que nos devuelve para que podamos utilizar en una variable.

```
var respuesta = confirm("¿Ya tiene cuenta?");

if (respuesta) {
    alert("Por favor ingrese su cuenta");
} else {
    alert("Por favor cree una nueva cuenta");
}
```

Math.floor()

Toma un número con coma (los llamamos floating point number, o **floats**) y lo redondea hacia abajo hasta el entero más cercano.

A pesar de que hacemos esta diferencia al hablar entre floats o números enteros, tenemos que recordas que Javascript sólo tiene un tipo de número.

```
Math.floor(3.2); // 3
Math.floor(4.8); // 4
Math.floor(10.5); // 10
```

Math.ceil()

- Al igual que **Math.floor()**, redondea un número, pero hacia arriba.

```
Math.ceil(3.2); // 4
Math.ceil(4.8); // 5
Math.ceil(10.5); // 11
```

Math.random()

- Nos sirve para generar números aleatorios.
- Devuelve un número entre 0 y 1.
- Para obtener enteros entre cierto rango, podemos realizar operaciones básicas con el resultado y usar Math.floor() para redondear

```
Math.random(); // 0 - 1
Math.floor(Math.random() * 20); // 0 - 19
```

Math.round()

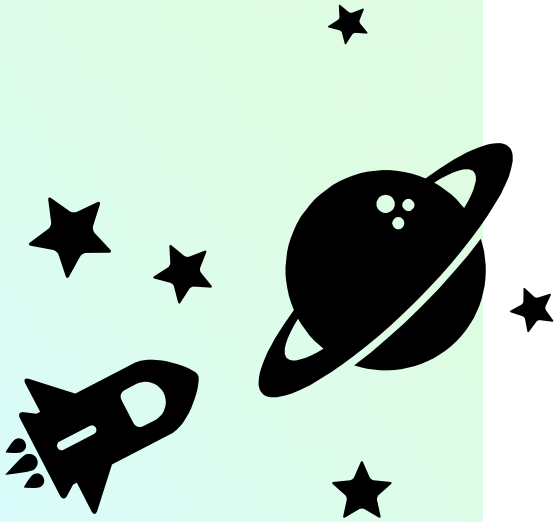
- Al igual que **Math.floor()**, redondea un número, pero hacia el número entero más cercano

```
Math.round(3.2); // 3
Math.round(4.5); // 5
Math.round(10.7); // 11
```

Actividad

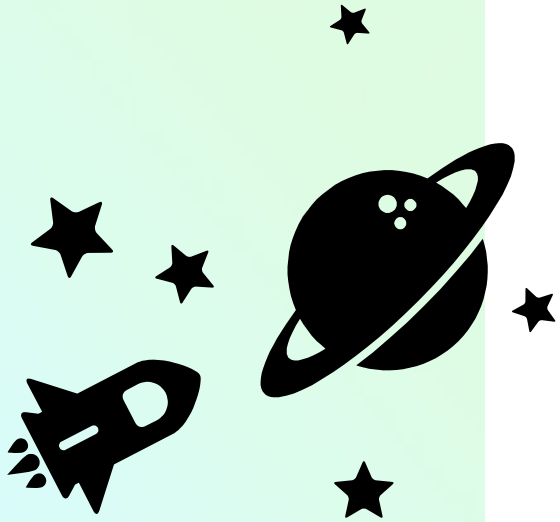
Hacer una función que tome un precio y devuelva un alert informando el precio con y sin iva.

Extra: un segundo parámetro booleano para ver si tiene o no descuento del 15%.



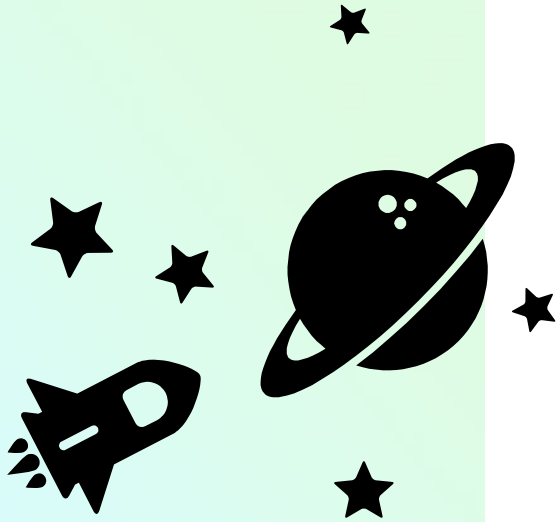
Actividad

Función calcula sueldo. Ingresar sueldo base y horas extra. Devolver categoria (1 numero cada 1000 de sueldo, ejemplo, $4000 -> cat. 4, hasta cat. 5). -10% de aportes. Hora extra se paga 1.5 de hora normal para todas las cat. Menos 4 y 5 que pagan 2. Imprimir a consola sueldo base, aportes, horas extras, categoría y total. Devolver sueldo total.



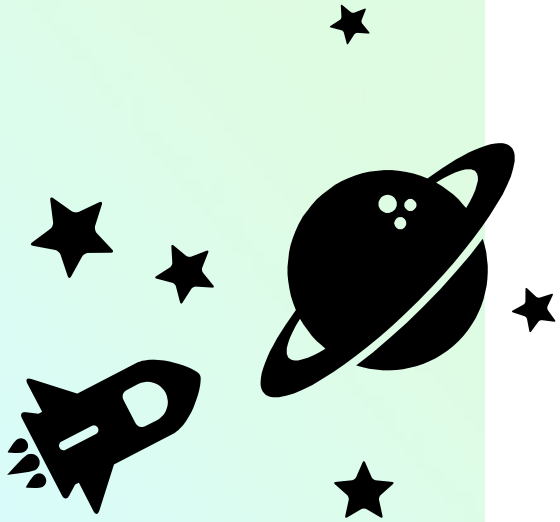
Actividad

Rehacer el juego de dados de la clase anterior, guardando la puntuación del usuario. Debería durar 5 turnos o hasta que alguien gane.



Actividad

Rehacer el Piedra, papel o tijera! de la clase anterior, guardando la puntuación del usuario. Debería durar 10 turnos y después anunciar el ganador. Extra: chequear empate.

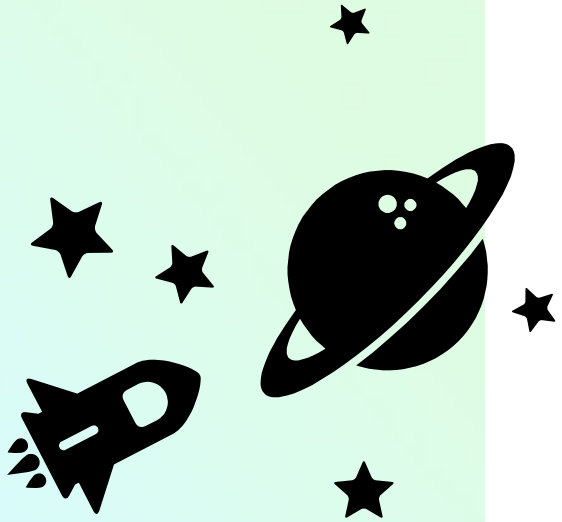


T.P. Nº 3

Mejorar nuestra calculadora.

Deberá poder ingresar 2 valores y qué operación quiere realizar.

Imprimir resultado en la consola.



Hasta la próxima clase!

¿Preguntas?

Enviar a

namorcamila@gmail.com