TECNOLOGICO NACIONAL DE MEXICO

CAMPUS MILPA ALTA II

ING. EN SISTEMAS COMPUTACIONALES

MEAN STACK FRONT END

TEMA:

INVESTIGACION DE FUNCIONES

PROFESOR:
ROLDAN AQUINO SEGURA

ALUMNO:

BOLLAS PAREDES DIEGO ALBERTO

No. Control: 151190087

Prototype

No importa cómo declares la función, una función en JavaScript siempre tendrá una propiedad prototype predeterminada.

```
function hacerAlgo(){}

console.log( hacerAlgo.prototype );

var hacerAlgo = function(){};

console.log( hacerAlgo.prototype );
```

Arguments

Arguments es un objeto similar a Array accesible dentro de funciones que contiene los valores de los argumentos pasados a esa función.

El objeto arguments es una variable local disponible en todas las funciones que no son funciones flecha. Puedes hacer referencia a los argumentos de una función dentro de esa función utilizando su objeto arguments. Tiene entradas para cada argumento con el que se llamó a la función, con el índice de la primera entrada en 0.

```
function func1(a, b, c) {
    console.log(arguments[0]);
    console.log(arguments[1]);
    console.log(arguments[2]);
}

func1(1, 2, 3);
```

Objeto Number

Number es un objeto primitivo envolvente que permite representar y manipular valores numéricos cómo 37 o -9.25. El constructor Number contiene constantes y métodos para trabajar con números. Valores de otro tipo pueden ser convertidos a números usando la función Number().

```
new Number(value);
var a = new Number('123'); // a === 123 es false
var b = Number('123'); // b === 123 es true
a instanceof Number; // es true
b instanceof Number; // es false
```

Objeto String

El objeto String se utiliza para representar y manipular una secuencia de caracteres.

Las cadenas son útiles para almacenar datos que se pueden representar en forma de texto. Algunas de las operaciones más utilizadas en cadenas son verificar su length, para construirlas y concatenarlas usando operadores de cadena + y +=, verificando la existencia o ubicación de subcadenas con indexOf() o extraer subcadenas con el método substring().

```
const string1 = "Una cadena primitiva";
const string2 = 'También una cadena primitiva';
const string3 = `Otra cadena primitiva más`;

const string4 = new String("Un objeto String");
```

Objeto Date

Permite trabajar con fechas y horas.

```
new Date()
new Date(milisegundos)
new Date(cadenaFecha)
new Date(año_num,mes_num,dia_num
        [,hor_num,min_num,seg_num,mils_num])
```

Objeto Match

El método match() se usa para obtener todas las ocurrencias de una expresión regular dentro de una cadena.

```
cadena.match(regexp)

cadena = "Para más información, vea Capítulo 3.4.5.1";
expresion = /(capítulo \d+(\.\d)*)/i;
hallado = cadena.match(expresion);
console.log(hallado);
```

Expresiones Regulares

Las expresiones regulares son patrones que se utilizan para hacer coincidir combinaciones de caracteres en cadenas. En JavaScript, las expresiones regulares también son objetos. Estos patrones se utilizan con los métodos exec() y test() de RegExp, y con match(), matchAll(), replace(), replaceAll() (en-US), search() y split() métodos de String. Este capítulo describe las expresiones regulares de JavaScript.

```
function escapeRegExp(string) {
    return string.replace(/[.*+\-?^${}()|[\]\\]/g,'\\$&'); // $& significa toda la cadena coincidente
}
```