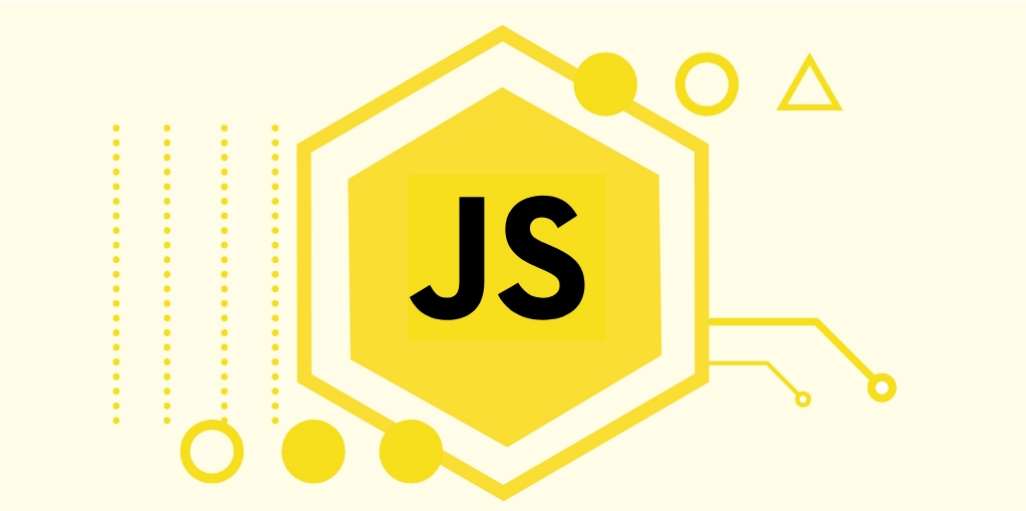
Universidad Autonoma de Aguascalientes

LICENCIATURA EN MATEMATICAS APLICADAS

MATERIA: Java Script

Docente: Bryan Ricardo Barbosa Olvera

FECHA DE CREACION: 18 de Junio de 2022



BRYAN RICARDO BARBOSA OLVERA

18 de junio de 2022

1. POO

1.1. Crear Objetos

```
const producto = {
nombre: "Monitor 20 pulgadas",
precio: 300,
}
```

1.2. Acceder a los valores

```
const producto = {
nombre: "Monitor 20 pulgadas",
precio: 300,
}
console.log(producto.nombre);
```

1.3. Agregar O Eliminar valores

```
const producto = {
nombre: "Monitor 20 pulgadas",
precio: 300,
}
//Agregar nuevas propiedades
producto.imagen = ïmagen.jpg";
//ELiminar propiedades del objeto
delete producto.precio
```

1.4. Destructuring

```
const producto = {
nombre: "Monitor 20 pulgadas",
precio: 300,
}
//Crea y asigna el valor a la variable
const nombre, precio = producto;
console.log(nombre,precio);
```

1.5. Destructuring de objetos anidados

```
const producto = {
nombre: "Monitor 20 pulgadas",
precio: 300,
informacion: {
fabricacion: {
pais: 'CHina'
}}}
const { nombre, informacion , informacion : { fabricacion: {pais }}} = producto;
console.log(pais);
```

1.6. Conjelar un objeto

```
”use strict ”
const producto = {
nombre: "Monitor 20 pulgadas",
precio: 300,
}
//Al conjelar el objeto no deja que se le agregen o eliminen valores del objeto
Object.freeze(producto);
//el siguiente comando indica con un true si esta conjelado el objeto
console.log(Object.isFrozen(producto));
```

1.7. Sellar un objeto

```
”use strict ”
const producto = {
nombre: "Monitor 20 pulgadas",
precio: 300,
}
//Al sellar un objeto es parecido a conjelarlo con la diferencia que le permite cambiar el valor de las llaves
Object.seal(producto);
producto.precio = 200;
//el siguiente comando indica con un true si esta sellado el objeto
console.log(Object.isSealed(producto));
```

1.8. Spread Operator o Rest Operator

```
const producto = {
nombre: "Monitor 20 pulgadas",
precio: 300,
}
const medidas = {
peso: ’1kg’,
medida: ’1m’
}
//Lo que realiza es unir dos objetos en uno solo
const resultado = { ...producto , ...medidas }
```

1.9. La palabra reservada this

```
//Te permite no perder la referencia de la varable que se esta llamando y no tomar variables fuera del objeto
const producto = {
nombre: "Monitor 20 pulgadas",
precio: 300,
,mostrarInfo: function () {
console.log(‘EL producto tiene como nombre: ${ this.nombre }‘) }
}
```

1.10. .keys .values .entries

```
const producto = {
nombre: "Monitor 20 pulgadas",
precio: 300,
}
//.keys te retorna las llaves del objeto en un objeto
console.log(Object.keys(producto));
//.values te retorna los valores del objeto en un objeto
console.log(Object.values(producto));
//.entries te retorna las llaves y los valores del objeto en pares en un objeto
console.log(Object.entries(producto));
```

2. ARRAYS

2.1. Crear un arreglo

const meses = ['Enero' , 'Febrero' , 'Marzo' , 'Abril' , 'Mayo' , 'Junio' , 'Julio'];

2.2. Acceder a los valores de un arreglo

const meses = ['Enero' , 'Febrero' , 'Marzo' , 'Abril' , 'Mayo' , 'Junio' , 'Julio'];
console.log(meses[0]); //Lo siguiente es para acceder al valor de un arreglo dentro de otro arreglo
const todo = [1,[1,2]];
console.log(todo[1][1]);
//Observece que imprimira el valor 2

2.3. Longitud de un arreglo

const meses = ['Enero' , 'Febrero' , 'Marzo' , 'Abril' , 'Mayo' , 'Junio' , 'Julio'];
console.log(meses.length);
//Observese que retornara como valor 7

2.4. Agregar un valor nuevo en un arreglo

const meses = ['Enero' , 'Febrero' , 'Marzo' , 'Abril' , 'Mayo' , 'Junio' , 'Julio'];
//Observamos que el arreglo solo tiene 7 valores, en el cual se puede acceder en la posicion6
//Para agregar uno nuevo es de la siguiente manera:
meses[7] = 'Nuevo mes';

2.5. Agregar elementos con el spreed operator al inicio o final

const meses = ['Enero' , 'Febrero' , 'Marzo' , 'Abril' , 'Mayo' , 'Junio' , 'Julio'];
let mesesNuevo = [...meses, 'Junio'];
mesesNuevo = ['Diciembre' , ...mesesNuevo];

2.6. Eliminar un valor al principio del arreglo

const meses = ['Enero' , 'Febrero' , 'Marzo' , 'Abril' , 'Mayo' , 'Junio' , 'Julio'];
meses.shift();

2.7. Eliminar un valor al final del arreglo

const meses = ['Enero' , 'Febrero' , 'Marzo' , 'Abril' , 'Mayo' , 'Junio' , 'Julio'];
meses.pop();

2.8. Eliminar un valor en cualquier posicion del arreglo

const meses = ['Enero' , 'Febrero' , 'Marzo' , 'Abril' , 'Mayo' , 'Junio' , 'Julio'];
meses.splice(1,1);
//Elimina febrero
meses.splice(posicion donde empesara a borrar valores , cantidad de valores que se quieren eliminar);

2.9. destructuring con Arreglos

```
const meses = ['Enero' , 'Febrero' , 'Marzo' , 'Abril' , 'Mayo' , 'Junio' , 'Julio'];
const [vEnero, vFebrero] = meses;
console.log(vFebrero);
//Observemos que crea la variable vFebrereo con el valor febrero no necesariamente con el mismo nombre
const [, vFebrero] = meses;
console.log(vFebrero);
//Observemos que crea la variable vFebrereo pero dejamos un espacio para indicar que no queremos crear una variable inecesaria
const [, vFebrero,...mesesFaltantes] = meses;
console.log(mesesFaltantes);
//meses Faltantes obtendra un arreglo con todos los valores que no se asignaron
```

2.10. Iterarcon Arreglos con .forEach

```
const meses = ['Enero' , 'Febrero' , 'Marzo' , 'Abril' , 'Mayo' , 'Junio' , 'Julio'];
meses.forEach((producto) {
console.log(producto);
} )
```

2.11. Iterarcon Arreglos con .map

```
const meses = ['Enero' , 'Febrero' , 'Marzo' , 'Abril' , 'Mayo' , 'Junio' , 'Julio'];
meses.map((producto) {
console.log(producto);
} )
//Realiza lo mismo que el forEach pero el map crea un nuevo arreglo
```

3. FUNCIONES