Definición y creación de objetos

- C.F.G.S. DAW
- 6 horas semanales
- Mes aprox. de impartición: Dic
- iPasen - cjaedia071@g.educaand.es

Carmelo José Jaén Díaz

Índice

OBJETIVO

- Profundizar en el concepto de objeto.
- Conocer y manejar funciones relativas al lenguaje sobre arrays, strings, números.

GLOSARIO

Backtracking. Estrategia utilizada en algoritmos que resuelven problemas que tienen ciertas restricciones. Este término fue creado por primera vez por el matemático D. H. Lehmer en la década de los cincuenta.

BOM (Browser Object Model). Convención específica implementada por los navegadores para que JavaScript pudiese hacer uso de sus métodos y propiedades de forma uniforme.

Expresión regular. Secuencia de caracteres que forman un patrón determinado, generalmente un patrón de búsqueda.

NaN. Propiedad que indica Not a Number (valor no numérico).

Objeto window. Aquel que soportan todos los navegadores y que representa la ventana del navegador. Se estudiará en profundidad en capítulos posteriores.

URI (Uniform Resource Identifier). Cadena de caracteres que identifica un recurso en una red de forma unívoca. Una URI puede ser una URL, una URN o ambas.

GLOSARIO

URN. Localizador de recursos en la web que funciona de forma parecida a una URL, pero su principal diferencia es que no indica exactamente dónde se encuentra dicho objeto.

INTRODUCCIÓN

En esta primera lección aprenderemos los **tres modos de hacer una definición y creación de objetos en Javascript**: de manera simple usando un literal, con la palabra reservada `new` o como definición de objeto propio utilizando `function`. Además, aprenderemos para qué se utiliza la palabra `this`.

A partir de la versión de Javascript ES6 se introdujeron las clases, una mejora sobre la herencia basada en prototipos que provee una sintaxis más clara y simple para crear objetos.

INTRODUCCIÓN

En estas primeras cuatro lecciones:

UT4.1 DEFINICIÓN Y CREACIÓN DE OBJETOS
UT4.2 OBJETOS. PROPIEDADES
UT4.3 OBJETOS. MÉTODOS
UT4.4 OBJETOS. PROTOTIPOS

se presenta la sintaxis empleada en la versión ES5 de JavaScript, con la finalidad didáctica de comprender los conceptos de definición, propiedades, métodos y prototipos de un objeto.

A partir, de la UT4.5 OBJETOS. CLASES se estudiará la nomenclatura actualizada.

DEFINICIÓN Y CREACIÓN DE OBJETOS
Utilizando un literal

A continuación, se presenta la sintaxis de los tres modos comentados anteriormente:

- El primero, utilizando un **literal**.

  Entre llaves irán las propiedades y valores de estas propiedades utilizando el formato *clave:valor*

```
let nombreObjeto = {
  miembro1clave: miembro1Valor,
  miembro2clave: miembro2Valor,
  miembro3clave: miembro3Valor
};
```

```
let nombreObjeto = {
    nombre: "Carmelo",
    apellido: "Jaen",
    inicio: 2020
};
```

DEFINICIÓN Y CREACIÓN DE OBJETOS
Utilizando *new*

- El segundo, utilizando la palabra reservada **new**:

  Recuerda: *la palabra reservada* `Object` *hace referencia a un tipo de datos objeto.*

```
let nombreObjeto = new Object();
```

Una vez definido el objeto, podemos definir las propiedades de dicho objeto indicando el nombre del objeto (.) nombre de la variable:

```
nombreObjeto.nombre = "Carmelo";
nombreObjeto.apellido = "Jaen";
nombreObjeto.inicio = 2020;
```

DEFINICIÓN Y CREACIÓN DE OBJETOS
Limitaciones

LIMITACIÓN

En los casos anteriores, la definición de objetos **no es escalable**, es decir, si quisiéramos definir más objetos tendríamos que copiar y pegar las sentencias anteriores para cada uno de los objetos a crear.

A continuación se presenta la manera, más recomendable, en la que se crea un tipo de objeto y se le indican las propiedades que va a tener ese objeto para finalmente asignarle los valores correspondientes.

DEFINICIÓN Y CREACIÓN DE OBJETOS
Utilizando *function*

- Y por último, **la más recomendable**, definiendo un constructor de un objeto utilizando la palabra `function` y crear objetos del tipo construido. ***Importante:*** *No confundas los objetos con funciones.*

```
function NombreObjeto ([parametros_si_los_hubiese]){
        this.propiedad1 = parametro1;
        this.propiedad2 = parametro2;
        this.propiedad3 = parametro3;
    }
```

DEFINICIÓN Y CREACIÓN DE OBJETOS
Utilizando *function*

EJEMPLO

Se crea un objeto de nombre `persona` que tendrá tres propiedades `nombre`, `apellido` e `inicio`.

Para asignarle los argumentos pasados como valores a esas propiedades utilizamos la palabra reservada `this`.

```
function Persona (nom, ape, ini){
        this.nombre = nom;
        this.apellido = ape;
        this.inicio = ini;
    }
```

DEFINICIÓN Y CREACIÓN DE OBJETOS Utilizando *function*

PALABRA RESERVADA `this`

En el caso concreto de JavaScript, la palabra reservada **`this`**:

- Hace referencia al propietario de la función que está invocándose.
- Hace referencia al objeto donde la función es un método.
- Por lo que depende de dónde se encuentre esta palabra reservada hará una u otra función.

En el caso que nos ocupa (constructor de objetos) **`this.nombre`**, **`this`** está haciendo referencia al objeto que se encuentra en la función **`NombreObjeto`** y **`nombre`** está refiriéndose a la propiedad nombre de **`NombreObjeto`**.

DEFINICIÓN Y CREACIÓN DE OBJETOS
Utilizando *function*

EJEMPLO

Es indiferente si los nombres de los parámetros y propiedades coinciden. En este caso, se ha optado por no hacerlo por una finalidad didáctica.

```
function NombreObjeto (nombre, apellido, inicio){
        this.nombre = nombre;
        this.apellido = apellido;
        this.inicio = inicio;
    }
```

Podemos decir que lo que estamos definiendo es una plantilla de objetos, con la finalidad de llamar a la función `NombreObjeto`, pasarle como argumento los valores que queremos asignarle a sus propiedades y así tendremos un nuevo objeto creado, es por ello, por lo que a esta función se le conoce como **constructor** (de objetos).

DEFINICIÓN Y CREACIÓN DE OBJETOS
Utilizando *function*

EJEMPLO

Si queremos crear un nuevo objeto de tipo `NombreObjeto`, haremos:

- Declarar una nueva variable para acceder posteriormente a las propiedades del objeto.
- En la definición utilizaremos la palabra reservada new seguido del nombre del objeto.
- Entre paréntesis, pasaremos los valores que queramos asignar a las propiedades del objeto.

```
let docente = new NombreObjeto ("Carmelo", "Jaén", 2020);
```

DEFINICIÓN Y CREACIÓN DE OBJETOS Utilizando *function*

PALABRA RESERVADA `this`

En el contexto de la variable **`docente`**, cuando invocamos al constructor (**`NombreObjeto`**) lo que hacemos es referirnos a la variable **`docente`** y asignarle mediante la palabra reservada **`this`**, los valores para sus propiedades **`nombre`**, **`apellido`** e **`inicio`**.