Meza Vargas Brandon David 2CM16

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(), mathAll(), replace(), replaceAll(), search() y Split() métodos de String.

Para construir una expresión regular hay dos formas:
-Usando una expresión regular literal, que consiste en un patrón encerrado entre barras:
Let re = /ab+c/
-Llamando a la función constructora del objeto RegExp:
Let re = new RegExp('sb+c')

Arreglos asociativos

Un array asociativo es un array cuyos índices no son numéricos sino cadenas de texto (claves). En Javascript no existen como tal los arreglos asociativos, pero se pueden simular creando objetos y accediendo a sus propiedades.

Declarar un array asociativo

Se usan llaves {} para generar el array de elementos clave:valor. Podemos añadir elementos de dos maneras:

```
Let coche = new Array()
Coche["color"] = "rojo"
Coche["marca"] = "seat"
Coche["modelo"] = "leon"
```

O añadir elementos así:

```
Let coche =
{"coche":"rojo","marca":"Seat","modelo":"león"}
```

Aquí también se pueden guardar datos de distintos tipos.

Para recorrer un array asociativo se usa el bucle for in:

```
for(let clave in coche){
    document.write(clave+": "+coche[clave])
}
```

Notación JSON

Javascript Object Notation (JSON) es un formato basado en texto estándar para representar datos estructurados en la sintaxis de objetos de Javascript. Es comúnmente utilizado para transmitir datos en aplicaciones web.
Los JSON son cadenas útiles cuando se quiere transmitir datos a través de una red. Debe ser convertido a un objeto nativo de Javascript cuando se requiera accedes a sus datos.

Estructura

Un JSON es una cadena cuyo formato recuerda a los objetos literales Javascript. Es posible incluir en los mismos tipos de datos básicos dentro de un JSON que en un objeto estándar de Javascript – cadenas, números, arreglos, booleanos y otros literales de objeto.
Para escribir en JSON se usan comillas dobles para las cadenas y los nombres de las propiedades. Las comillas simples no son validas.

Un ejemplo de un JSON

```
{
  "squadName": "Super hero squad",
  "active": true,
  "members": [
    {
      "name": "Molecule Man",
      "age": 29,
      "powers": [
        "Radiation resistance",
        "Radiation blast"
      ]
  } }
```

Definición de clases con class

Las clases de javascript, introducidas en ECMAScript 2015, son una mejora sintáctica sobre la herencia basada en prototipos de JavaScript. La sintaxis de las clases no introduce un nuevo modelo de herencia orientada a objetos en JavaScript. Las clases de JavaScript proveen una sintaxis mucho más clara y simple para crear objetos y lidiar con la herencia.

Declaración

Para declarar clases se utiliza la palabra reservada class y un nombre para la clase.
Veamos un ejemplo:

```
class Rectángulo{
   constructor(alto, ancho){
      this.alto = alto
     this.ancho = ancho
   }
}
```

Es importante mencionar que siempre necesitas declarar tu clase primero antes de intentar acceder a ella.
La palabra clave extends es usada en declaraciones de clase o expresiones de clase para crear una clase hija. La palabra super es usada para llamar a funciones del objeto padre