**Pràctica 13**

**L’objecte String**

Les cadenes en JavaScript són objectes de la classe String, que contenen diversos mètodes que ens serveixen per gestionar cadenes de caràcters. Estudiarem les seves propietats i els seus mètodes més importants.  
  
En JavasSript les variables de tipus text són objectes de la classe String. Això vol dir que cadascuna de les variables que creem de tipus text té una sèrie de propietats i mètodes. Recordem que les propietats són les característiques, com per exemple longitud en caràcters de l'string i els mètodes són funcionalitats, com poden ser extreure un substring o posar el text en majúscules.  
  
Per crear un objecte de la classe String l'únic que cal fer és assignar un text a una variable:

var cadena = 'Institut Marianao';

El text va entre cometes, com ja hem vist en els capítols de sintaxi.

També es pot crear un objecte String amb l'operador new com s’indica a continuació:

var cadena = new String('Institut Marianao');

L'única diferència és que en versions de Javascript 1.0 no funcionarà new per crear els Strings.

**Propietats de l’objecte String**

* **length**

La classe String només té una propietat: length, que guarda el nombre de caràcters de l’String. Podem veure com funciona al següent exemple:

var cadena = 'Marianao';

console.log(cadena.length); // Retorna 8

**Mètodes de l’objecte String**  
  
Els objectes de la classe String tenen una bona quantitat de mètodes per a realitzar moltes coses interessants. A continuació detallarem els més importants:

* **charAt(índex)**  
    
  Retorna el caràcter que hi ha a la posició indicada com a índex. Les posicions d'un string comencen a 0.
* **indexOf(caràcter, des de)**  
    
  Retorna la posició de la primera vegada que apareix el caràcter indicat per paràmetre en un string. Si no troba el caràcter en el string retorna -1. El segon paràmetre és opcional i serveix per indicar a partir de quina posició es desitja que comenci la recerca.
* **lastIndexOf(caràcter, des de)**  
    
  Cerca la posició d'un caràcter exactament igual a com ho fa la funció indexOf() però des del final en lloc del principi. El segon paràmetre indica el nombre de caràcters des d'on es busca, igual que a indexOf().
* **replace(substring\_a\_buscar, nouStr)**  
    
  Serveix per reemplaçar porcions del text d'un string per un altre text, per exemple, podríem utilitzar-lo per a substituir totes les aparicions del substring "xxx" per "yyy". El mètode no reemplaça en el string, sinó que retorna el resultat de fer aquest reemplaçament. Accepta expressions regulars com substring a buscar.
* **split(separador)**  
    
  Serveix per crear un array a partir d'un String en què cada element és la part del String que està separada pel separador indicat per paràmetre.
* **substring(inici, fi)**  
    
  Retorna el substring que comença en el caràcter d'inici i acaba en el caràcter de final. Si intercanviem els paràmetres d'inici i fi també funciona. Simplement ens dóna el substring que hi ha entre el caràcter menor i el major.
* **toLowerCase()**  
    
  Posa totes els caràcters d'un string en minúscules.
* **toUpperCase()**  
    
  Posa totes els caràcters d'un string en majúscules.
* **toString()**  
    
  Aquest mètode el tenen tots els objectes i s'utilitza per convertir-los en cadenes.
* **search(substring\_a\_buscar)**

Busca una cadena amb el valor indicat i retorna la posició de la coincidència.

El valor de la cerca pot ser una cadena o una expressió regular.

Aquest mètode retorna -1 si no es troba cap coincidència.

**PROBLEMES**

1. Proba totes les propietats i mètodes d’aquest document en una pàgina HTML. (Per mostrar el valor que retornen els mètodes pots utilitzar console.log())