

# EXAMEN: Sistemes i Tecnologies Web 26 Juny 2017

Niu:

Nom:

Permutació A

## JEE: Tokenizer

Si entregueu via moixero.uab.cat, ompliu el següent requadre. Altrament, ompliu la resta de l'examen.

### Entrega Electrònica

The SHA1 checksum of the received file is:

.....

Time stamp is:

.....

Guardeu-vos una copia de l'arxiu que entregueu.

>>>> L'entrega no electrònica puntua sobre 7 / 10. <<<<

# Introducció

S'ha d'implementar un codi javascript que s'executarà a NodeJS. Per poder fer-lo cal dominar els següents tres conceptes: callbacks, clausures, i classes.

L'examen consisteix en implementar una classe anomenada `Tokenizer` i una funció de proves. La classe `Tokenizer` escanejarà una llista de caràcters i per cadascun, si li hem dit abans, cridarà una funció.

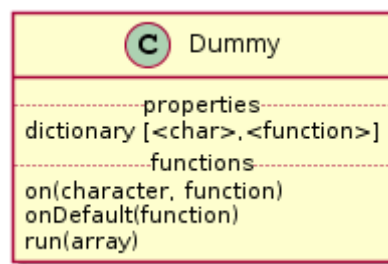
Per exemple:

```
t = new Tokenizer();
t.on('1', callback);           // crida callback() quan vegis un '1'
t.run(['1','2','3','1']);      // mira't aquesta llista.
```

Aquí estem escanejant la llista `['1','2','3','1']` amb el `Tokenizer`, i li hem dit que quan vegi un `'1'` que cridi la funció `callback()`. Així doncs primer veurà el `'1'` i cridarà la funció `callback`. Després veurà el `'2'` i no farà res. Tampoc farà res amb el `'3'`. Després veurà el `'1'` del final de la llista i tornarà a cridar la funció `callback`.

## Especificacions

El diagrama de classes de la classe `Tokenizer` és el següent:



A continuació s'indica què han de fer les funcions de la classe i el funcionament de la propietat `dictionary`. Heu d'afegir les propietats i funcions que us calguin.

### Propietat `dictionary`

La classe `Tokenizer` ha de contenir un diccionari anomenat `dictionary`. Els index d'aquest diccionari seran caràcters. A cada index del diccionari hi guardarem una funció. O dit d'una altra forma registrarem una funció per cada caràcter.

Visualment el diccionari de la classe `Tokenizer` podria ser el següent:

```
// [ index => value ]

[
  'a' => function(){console.log('found a')};},
  'b' => function(){console.log('found b')};},
  'z' => function(){console.log('found z')};}
]
```

### Detalls d'implementació del diccionari

1. El podeu implementar com un objecte, o com una llista (és la mateixa sintaxi).
2. Si intentem accedir a un índex del diccionari que no existeix obtenim `undefined`.

```
var a = { 'x' : function() { console.log('Hello World') } };  
  
if (a['y'] == undefined) {  
    // ...  
}
```

## Funcions

La classe `Tokenizer` ha de tenir tres funcions: `on`, `onDefault`, i `run`.

### Funció `run`

La funció `run` rebrà com a paràmetre una llista de caràcters on hi poden haver repeticions (i.e., `['a', 'b', 'c']`). La funció `run` examinarà els caràcters de la llista, un a un, començant pel primer caràcter i acabant a l'últim (per exemple, amb `llista.forEach(...)`).

Quan la funció `run` trobi un caràcter que estigui al diccionari, s'executarà la funció associada a aquest caràcter.

Quan la funció `run` trobi un caràcter que *no* estigui al diccionari, s'executarà la funció que amb anterioritat se li hagi passat com a argument a la funció `onDefault`.

Si no s'ha cridat la funció `onDefault` amb anterioritat, no es cridarà a cap funció pel caràcter que estem examinant i no s'ha de produir cap error.

### Funció `on`

La funció `on(c, f)` associa el caràcter `c` amb la funció `f` al diccionari `dictionary`, de forma que quan `run` escanegi una llista de caràcters, `run` cridi la funció `f` quan trobi el caràcter `c`.

### Funció `onDefault`

La funció `onDefault(f)` registra (guarda) la funció `f` al `Tokenizer`. Quan la funció `run` del `Tokenizer` escanegi una llista de caràcters i trobi que un caràcter que no estigui en el diccionari, haurà de cridar la funció `f`.

## Banc de Proves

A l'esquelet que se us proporciona teniu la funció `testTokenizer()` que també heu de completar. Aquesta funció testeja l'objecte `Tokenizer`.

Cal que:

1. Creeu un objecte de tipus `Tokenizer`.
2. Registreu al `Tokenizer` una funció que mantingui un comptador del nombre de caràcters 'a' que ha vist.
3. Registreu al `Tokenizer` una funció que mantingui un comptador del nombre de caràcters 'c' que ha vist.
4. Registreu al `Tokenizer` una funció que mantingui un comptador del nombre de caràcters diferents de 'a' i 'c' que ha vist.
5. Feu que el `Tokenizer` examini la llista a la variable `testString` de l'esquelet.

```
var testString = ['H','o','l','a',' ',' ','c','o','m',' ',' ','a','n','e','u','?'];
```

6. Imprimiu el nombre d'ocurrències del caràcter 'a' que s'han trobat a la cadena de test.
7. Imprimiu el nombre d'ocurrències del caràcter 'c' que s'han trobat a la cadena de test.
8. Imprimiu el nombre de caràcters de la cadena de test que no estaven al diccionari.

La sortida per consola d'executar la funció `testTokenizer()` ha de ser:

```
numero de a's: 2  
numero de c's: 1  
numero d'altres caracters: 11
```

# Com executar, testar i entregar l'exercici.

## Executar i Testar

1. Posicioneu-vos dins del directori tokenizer.
2. Executeu l'aplicació de NodeJS: `nodejs app.js`
3. Per consola heu de veure:

```
numero de a's: 2
numero de c's: 1
numero d'altres caracters: 11
```

## Entregar

Entregueu l'arxiu `app.js` via moixero si us ha funcionat tot, o copieu el codi que tingueu als requadres disponibles al final d'aquest document.

## Indicacions

(No és imprescindible llegir-se aquesta secció per fer l'exercici)

### Si l'exercici no us funciona assegureu-vos que heu entès aquest script

```
// Exemple 2: Que apareix a la consola?
var myObject = {
  foo: "bar",
  func: function() {
    var self = this;
    console.log("outer func: this.foo = "+ this.foo);
    console.log("outer func: self.foo = "+ self.foo);
    (function() {
      console.log("inner func: this.foo = " + this.foo);
      console.log("inner func: self.foo = " + self.foo);
    }());
  }
};

myObject.func();

// El this fa referencia al que hi ha abans del '.' quan s'ha cridat una funció.
// L'exemple és de https://www.toptal.com/javascript/interview-questions
```

## Resposta

El deliverable d'aquest exercici és un únic arxiu `app.js` que defineixi la classe `Tokenizer` i que executi la funció `testTokenizer`.

Ompliu els següents espais. Recordeu que la mida no dona cap indicació del nombre de línies de codi que us calen.

```
// TODO: implement class Tokenizer here.
```

```
function Tokenizer(){  
  this.dictionary = []; // {}  
  this.run =
```

```
    this.on =
```

```
    this.onDefault =
```

```
}
```

```

// TODO: complete the function testTokenizer()
function testTokenizer(){

    var t =

    var countA = 0;
    var testString = ['H','o','l','a',' ','c','o','m',' ','a','n','e','u','?'];

    t.on('a',

    );

    t.onDefault(

    );

    // Write down the code to run the test over testString

    console.log("numero de a's: " + countA);
    console.log("numero de c's: " +

    );

    console.log("numero d'altres caracters: " +

    );

}

```

testTokenizer();