

IFT1015 Programmation 1

Traitement de texte

Marc Feeley

Méthodes importantes sur les textes

- **`t.length`** *(méthode spéciale sans paramètres)*
 - retourne le nombre de caractères dans le texte *t*
- **`t.charAt(i)`**
 - retourne un texte de longueur 1, le caractère à l'index *i* du texte *t*
- **`t.charCodeAt(i)`**
 - retourne un entier qui est le **code Unicode** du caractère à l'index *i* du texte *t*
- **`t.slice(i, j)`**
 - retourne un texte qui contient les caractères de *t* entre l'index *i* (inclus) et *j* (exclus)

Extraction de sous-textes

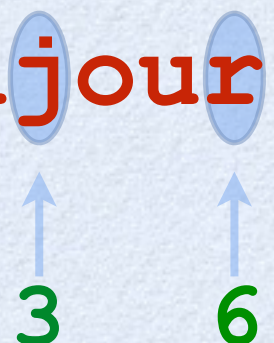
- Soit **t** une variable de type texte
- **t.charAt(i)** retourne le caractère à la position *i* (les index de caractères vont de 0 à longueur-1)

Exemple: **"Bonjour".charAt(3)** vaut **"j"**

"Bonjour".charAt(6) vaut **"r"**

- **t.slice(debut, fin)** retourne un sous-texte

Exemple : **"Bonjour".slice(3, 6)** vaut **"jou"**



Similarité des méthodes sur les textes et tableaux

```
> var t = "ABCD";  
  
> t.length  
4  
  
> t.charAt(1)  
"B"  
  
> t.slice(1,3);  
"BC"  
  
> t+"EF";  
"ABCDEF"  
  
> t.charCodeAt(0);  
65
```

```
> var t = ["A", "B", "C", "D"];  
  
> t.length  
4  
  
> t[1]  
"B"  
  
> t.slice(1,3);  
["B", "C"]  
  
> t.concat(["E", "F"]);  
["A", "B", "C", "D", "E", "F"]  
  
> t.concat(["EF"]);  
["A", "B", "C", "D", "EF"]
```

Exemple : encodagePositionnel

```
var encodagePositionnel = function (n, base) {  
  
    // Cette fonction retourne l'encodage du nombre n dans la  
    // base spécifiée, qui peut être entre 2 et 36.  
  
    var chiffres = "0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ";  
  
    var e = "";                                // pour accumuler l'encodage  
  
    do {  
        e = chiffres.charAt(n % base) + e; // accumuler un chiffre  
        n = Math.floor(n / base); // passer aux autres chiffres  
    } while (n > 0); // tant qu'il reste des chiffres  
  
    return e;  
};  
  
print( "0x" + encodagePositionnel(51966, 16) ); // imprime 0xCAFE
```


Exemple : encodagePositionnel

```
var encodagePositionnel = function (n, base) {  
  
    // Cette fonction retourne l'encodage du nombre n dans la  
    // base spécifiée, qui peut être entre 2 et 36.  
  
    var chiffres = "0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ";  
  
    var e = ""; // pour accumuler l'encodage  
  
    do {  
        var c = n % base; // le prochain chiffre  
        e = chiffres.slice(c, c+1) + e; // accumuler un chiffre  
        n = Math.floor(n / base); // passer aux autres chiffres  
    } while (n > 0); // tant qu'il reste des chiffres  
  
    return e;  
};  
  
print( "0x" + encodagePositionnel(51966, 16) ); // imprime 0xCAFE
```


Exemple : découper un texte

- Il est souvent nécessaire de découper un texte en parties (mots, lignes, etc)
- Par exemple :

"www.iro.umontreal.ca"



["www", "iro", "umontreal", "ca"]

texte contenant
des séparateurs

tableau de
textes

Exemple : découper un texte

```
var decouper = function (texte, separateur) {  
  var resultat = [];  
  var debut = 0;  
  var i = 0;  
  while (i < texte.length) {  
    if (texte.charAt(i) == separateur) {  
      resultat.push(texte.slice(debut, i));  
      debut = i + 1;  
    }  
    i++;  
  }  
  resultat.push(texte.slice(debut, i));  
  return resultat;  
};  
  
print( decouper("www.iro.umontreal.ca", ".") );
```


Exemple : découper en mots

- Un “mot” est une séquence de lettres
- Par exemple :

"oui... j'ai sept chats!"



["oui", "j", "ai", "sept", "chats"]

texte contenant
des mots

tableau des
mots

Exemple : découper en mots

```
var lettre = function (car) {  
    return (car >= "a" && car <= "z") ||  
           (car >= "A" && car <= "Z");  
};  
  
var decouperEnMots = function (texte) {  
    var resultat = [];  
    var debut = 0;  
    while (debut < texte.length) {  
        if (lettre(texte.charAt(debut))) {  
            var i = debut+1;  
            while (i < texte.length && lettre(texte.charAt(i))) {  
                i++;  
            }  
            resultat.push(texte.slice(debut, i));  
            debut = i+1;  
        } else {  
            debut++;  
        }  
    }  
    return resultat;  
};  
  
print( decouperEnMots("oui... j'ai sept chats!") );
```


Exemple : chercher un mot

- Trouver la position où se trouve un mot
- Par exemple :

```
positionMot("Je connais Java et JavaScript", "Java") => 11
```


Exemple : chercher un mot

version avec
charAt

```
var positionMot = function (texte, mot) {  
  prochainePos:  
  for (var i=0; i<=texte.length-mot.length; i++) {  
    for (var j=0; j<mot.length; j++) {  
      if (texte.charAt(i+j) !== mot.charAt(j)) {  
        continue prochainePos;  
      }  
    }  
    return i; // mot trouvé!  
  }  
  return -1; // indiquer échec  
};  
  
print( positionMot("Je connais Java et JavaScript", "Java") );
```


Exemple : chercher un mot

version avec
slice

```
var positionMot = function (texte, mot) {  
    for (var i=0; i<=texte.length-mot.length; i++) {  
        if (texte.slice(i, i+mot.length) == mot) {  
            return i; // mot trouvé!  
        }  
    }  
    return -1; // indiquer échec  
};  
  
print( positionMot("Je connais Java et JavaScript", "Java") );
```


La méthode indexOf

- Pour rechercher la position d'une valeur dans un tableau on peut se servir des méthodes **indexOf** et **lastIndexOf**
- *tableau* . **indexOf** (*valeur*)
 - retourne l'index (le plus **bas**) où se trouve *valeur* dans *tableau* (-1 si la valeur n'existe pas dans *tableau*)
- *tableau* . **lastIndexOf** (*valeur*)
 - retourne l'index (le plus **haut**) où se trouve *valeur* dans *tableau* (-1 si la valeur n'existe pas dans *tableau*)

La méthode indexOf

- La méthode **indexOf** fait donc le même travail que la fonction **position** :

```
var position = function (t, val) {  
    for (var i=0; i<t.length; i++) {  
        if (t[i] == val) {  
            return i; // on a trouvé l'index!  
        }  
    }  
  
    return -1; // code indiquant échec  
};
```

tableau.**indexOf**(*valeur*) ≡ **position**(*tableau*, *valeur*)

La méthode `lastIndexOf`

- La méthode `lastIndexOf` fait une recherche descendante comme la fonction `lastPosition`:

```
var lastPosition = function (t, val) {  
    for (var i=t.length-1; i>=0; i--) {  
        if (t[i] == val) {  
            return i; // on a trouvé l'index!  
        }  
    }  
  
    return -1; // code indiquant échec  
};
```

tableau.`lastIndexOf`(*valeur*) ≡ `lastPosition`(*tableau*, *valeur*)

La méthode indexOf

- On peut indiquer le départ de la recherche aux méthodes **indexOf** et **lastIndexOf**
- *tableau* . **indexOf** (*valeur* , *départ*)
 - retourne l'index le plus **bas** et \geq *départ* où se trouve *valeur* dans *tableau* (-1 si la valeur n'est pas trouvée)
- *tableau* . **lastIndexOf** (*valeur* , *départ*)
 - retourne l'index le plus **haut** et \leq *départ* où se trouve *valeur* dans *tableau* (-1 si la valeur n'est pas trouvée)

La méthode indexOf

- Les méthodes **indexOf** et **lastIndexOf** existent aussi sur les textes :
- *texte*.**indexOf** (*texte2*)
 - retourne l'index (le plus **bas**) où se trouve *texte2* dans *texte* (-1 si la valeur n'existe pas dans *texte*)
- *texte*.**lastIndexOf** (*texte2*)
 - retourne l'index (le plus **haut**) où se trouve *texte2* dans *texte* (-1 si la valeur n'existe pas dans *texte*)

La méthode indexOf

- La méthode **indexOf** sur les textes fait donc le même travail que la fonction **positionMot** :

```
var positionMot = function (texte, texte2) {  
    for (var i=0; i<=texte.length-texte2.length; i++) {  
        if (texte.slice(i, i+texte2.length) == texte2) {  
            return i; // on a trouvé l'index!  
        }  
    }  
  
    return -1; // code indiquant échec  
};
```

texte.**indexOf**(*texte2*) ≡ **positionMot**(*texte*, *texte2*)

La méthode indexOf

- On peut indiquer le départ de la recherche aux méthodes **indexOf** et **lastIndexOf** sur les textes

"abracadabra".indexOf("bra", 2) vaut 8

Exemple : positions d'un mot

- Trouver les **positions** où se trouve un mot
- Utiliser un tableau de positions, par exemple :

```
positionsMot("Je connais Java et JavaScript", "Java")
```



```
[11, 19]
```


Exemple : positions d'un mot

version avec
slice

```
var positionsMot = function (texte, mot) {  
    var resultat = [];  
    for (var i=0; i<=texte.length-mot.length; i++) {  
        if (texte.slice(i, i+mot.length) == mot) {  
            resultat.push(i); // mot trouvé!  
        }  
    }  
    return resultat;  
};  
  
print( positionsMot("Je connais Java et JavaScript", "Java") );
```


Exemple : positions d'un mot

version 2 avec
slice

```
var positionsMot = function (texte, mot) {  
  var resultat = [];  
  for (var i=0; i<=texte.length-mot.length; i++) {  
    if (texte.slice(i, i+mot.length) == mot) {  
      resultat.push(i); // mot trouvé!  
      i += mot.length-1;  
    }  
  }  
  return resultat;  
};  
  
print( positionsMot("La CAA et AAA sont partenaires", "AA") );
```

évite de trouver
un autre AA ici




Exemple : positions d'un mot

version avec
indexOf

```
var positionsMot = function (texte, mot) {  
  var resultat = [];  
  var i = 0;  
  while (true) {  
    i = texte.indexOf(mot, i);  
    if (i == -1) break;  
    resultat.push(i); // mot trouvé!  
    i += mot.length;  
  }  
  return resultat;  
};
```

point de départ
de la prochaine
recherche du mot



```
print( positionsMot("La CAA et AAA sont partenaires", "AA") );
```


Exemple : remplacer un mot

- Dans un texte, remplacer un mot par un autre, par exemple :

```
remplacerMot("il a deux chats et deux chiens", "deux", "dix")
```



```
"il a dix chats et dix chiens"
```


Exemple : remplacer un mot

version avec
positionsMot

```
var remplacerMot = function (texte, mot, autre) {  
  var positions = positionsMot(texte, mot);  
  var resultat = "";  
  var debut = 0;  
  for (var i=0; i<positions.length; i++) {  
    resultat += texte.slice(debut, positions[i]) + autre;  
    debut = positions[i] + mot.length;  
  }  
  resultat += texte.slice(debut, texte.length);  
  return resultat;  
};  
  
print( remplacerMot("il a deux chats et deux chiens", "deux", "dix") );
```


Les méthodes `split` et `join`

- Les méthodes `split` et `join` permettent de découper et former un texte avec un séparateur
- `texte.split(texte2)`
 - retourne un tableau contenant les textes issus d'un découpage de `texte` au séparateur `texte2`
`"abXYcdXYef".split("XY")` vaut `["ab", "cd", "ef"]`
- `tableau.join(texte)`
 - retourne un texte formé de la concaténation des textes dans `tableau` séparés par `texte`
`["ab", "cd", "ef"].join("XY")` vaut `"abXYcdXYef"`

Exemple : remplacer un mot

version avec
`split` et `join`

```
var remplacerMot = function (texte, mot, autre) {  
    return texte.split(mot).join(autre);  
};  
  
print( remplacerMot("j'ai deux chats et deux chiens", "deux", "dix") );
```