JavaScript et HTML

Cours 3

Jean-Jacques Lévy

jean-jacques.levy@inria.fr

http://jeanjacqueslevy.net/lp-js

Plan

- solutions des exercices
- itération sur les tableaux et chaînes de caractères
- création d’objets
- classes en Javascript ES6
- alias

Itération sur un tableau

- itération sur un tableau avec une fonction

```
let r = 0
    a.forEach (m => {
        r = r + m
    })

console.log (r)
```

itération sur tous les éléments de a

```
let r = 0
    a.forEach (function (m) {
        r = r + m
    })

console.log (r)
```

- itération sur un tableau avec une boucle `for`

```
let r= 0
for (let m of a)
  r = r + m
console.log (r)
```

itération sur tous les éléments de a

- itération sur un tableau avec une boucle `for`

```
let r= 0
for (let i in a)
  r = r + a[i]

console.log (r)
```

itération sur tous les indices de a

```
let r = 0
for (let i = 0; i < a.length; i++)
  r = r +  a[i]

console.log (r)
```

Itération sur une chaîne de caractères

- itération sur une chaîne avec une fonction (en la transformant en tableau)

```
let s = ‘abbacadeaa'

let r = 0
Array.from(s).forEach (c => {
  if(c == 'a')
    ++r
})
```

itération sur tous les caractères de s

- itération sur un tableau avec une boucle `for`

```
function number_of_a (s) {
    let r = 0
    for (let c of s)
      if (c == 'a')
        ++r
    return r
  }
```

itération sur tous les caractères de s

- itération sur un tableau avec une boucle `for`

```
let r= 0
for (let i in s)
  if(s[i] == 'a')
    ++r
```

itération sur tous les caractères de s

```
let r = 0
for (let i = 0; i < a.length; i++)
  if (s[i] == 'a')
    ++r
```

Solutions des exercices

Exercice Écrire la fonction qui trouve l’indice de la plus grande valeur

Exercice Écrire la fonction qui teste le nombre de ‘a’ dans une chaîne de caractères

Exercice Écrire la fonction qui retourne l’indice du premier ‘abc’ dans une chaîne de caractères

Exercice Écrire la fonction qui teste si une chaîne de caractères est un palindrome

Exercice Écrire la fonction qui copie un tableau

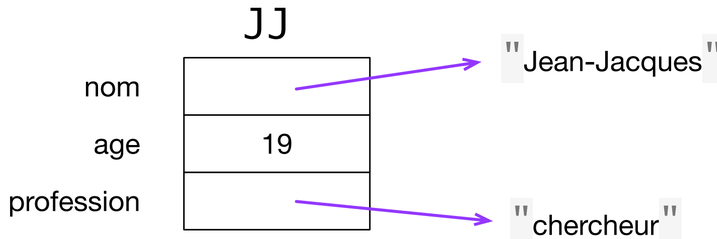
Objets

- en Javascript, on peut déclare les objets seuls (pas de classes)

```
const jj = {
  nom : 'Jean-Jacques',
  age : 19,
  profession : 'chercheur'
}
```

```
const wei = {
  nom : 'Wei',
  age : 28,
  profession : 'enseignant'
}
```

```
let iteki = {
    nom : 'Iteki',
    age : 36
  }
```



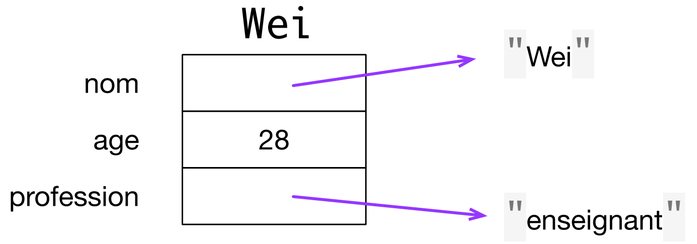
- on accède aux champs

```
console.log (jj)
console.log (`${jj.nom} a ${jj.age} ans`)
console.log (`${wei.nom} a ${wei.age} ans`)
console.log (`${iteki.nom} a ${iteki.age} ans`)
```

avec point

```
console.log (jj['nom'])
```

avec crochets



- attention aux alias

```
wei = iteki
```

interdit

```
iteki = wei
iteki.nom = 'albert'
console.log (iteki)
console.log (wei)
```

iteki et wei sont maintenant des alias

Objets et méthodes

- les objets sont des paires (clé, valeur), aussi appelées propriétés (*properties*)

- on peut rajouter ou modifier des propriétés

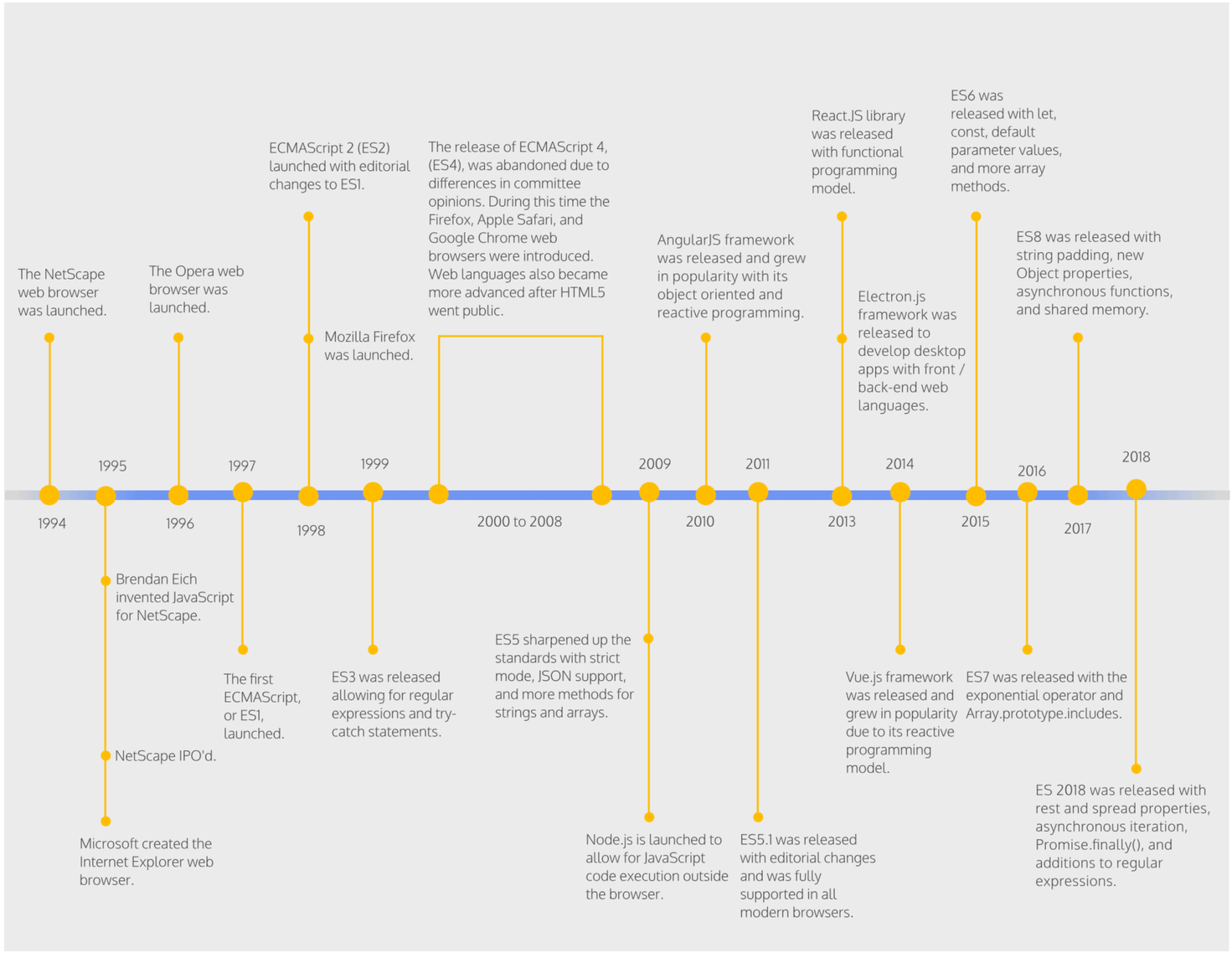
- certains champs peuvent être aussi des objets

```
const jj21 = {
    nom : 'Jean-Jacques',
    age : 19,
    profession : 'chercheur',
    vaccination : {
        date : '3 mars 2021',
        type: 'Moderna'
    }
  }

console.log (jj21.vaccination.date)
```

- certains champs peuvent aussi être des fonctions

```
const jj = {
    nom : 'Jean-Jacques',
    age : 19,
    profession : 'chercheur',
    hello : function () {console.log ("hello, c'est JJ")}
  }

console.log (jj.hello())
```

Javascript dans le temps



Classes

- en Javascript moderne (ES6, 2015), on utilise la notion usuelle de ‘classe’

```
class Personne {
    constructor (nom, age) {
        this.nom = nom
        this.age = age
    }
}

jj = new Personne ('Jean-Jacques', 19)
wei = new Personne ('Wei', 28)

console.log (jj)
```

- et on définir des méthodes tout aussi classiquement

```
class Personne {
    constructor (nom, age) {
        this.nom = nom
        this.age = age
    }

    hello () {console.log (`hello, c'est ${this.nom}`)}
}

jj = new Personne ('Jean-Jacques', 19)
wei = new Personne ('Wei', 28)
console.log (wei.hello())
```

Classes

- on peut utiliser toutes les notations des fonctions

```
class Personne {
    constructor (nom, age) {
        this.nom = nom
        this.age = age
    }

    hello = function () {console.log (`hello, c'est ${this.nom}`)}
}
```

```
class Personne {
    constructor (nom, age) {
        this.nom = nom
        this.age = age
    }

    hello = () => {console.log (`hello, c'est ${this.nom}`)}
}
```

Sous-classes

- on peut étendre des classes

```
class Etudiant extends Personne {
    constructor (nom, age, matiere) {
        super (nom, age)
        this.matiere = matiere
    }
}

let iteki = new Etudiant ('Iteki', 36, 'chinois')
console.log (iteki)
```

Tableaux et méthodes

- les tableaux sont des objets particuliers (indice, valeur)

```
const table1 = ['manger', 'dormir']
const table2 = new Array ('manger', 'dormir')
```

concat()	joins two or more arrays and returns a result
indexOf()	searches an element of an array and returns its position
find()	returns the first value of an array element that passes a test
findIndex()	returns the first index of an array element that passes a test
forEach()	calls a function for each element
includes()	checks if an array contains a specified element

push()	aads a new element to the end of an array and returns the new length of an array
unshift()	adds a new element to the beginning of an array and returns the new length of an array
pop()	removes the last element of an array and returns the removed element
shift()	removes the first element of an array and returns the removed element
sort()	sorts the elements alphabetically in strings and in ascending order
slice()	selects the part of an array and returns the new array
splice()	removes or replaces existing elements and/or adds new elements

Chaînes de caractères et méthodes

- les tableaux sont des objets particuliers (indice, valeur)

```
const s1 = "abcdefg"
const s2 = new String ("abcdefg")

console.log (s1, " ", s2)
console.log (s1[0], " ", s2[0])
console.log (typeof (s1), " ", typeof (s2))
```

moins efficace — ne pas utiliser

charAt(index)	returns the character at the specified index
concat()	joins two or more strings
replace()	replaces a string with another string
split()	converts the string to an array of strings
substr(start, length)	returns a part of a string
substring(start,end)	returns a part of a string

slice(start, end)	returns a part of a string
toLowerCase()	returns the passed string in lower case
toUpperCase()	returns the passed string in upper case
trim()	removes whitespace from the strings
includes()	searches for a string and returns a boolean value
search()	searches for a string and returns a position of a match

Prochain cours

- un bon tutorial JavaScript: `http://www.programiz.com/javascript`
- un autre tutorial JavaScript: `http://www.w3schools.com/js`
- documents structurés en HTML