Introduction a javascript

Dr Bakary Diarra
Cité collégiale

Plan de la presentation

- Introduction
- Les variables et leur portée
- Les instructions conditionnelles
- Les boucles
- Fonctions mathématiques

Introduction

- Javascript est un langage très populaire de nos jours
- Il sert à programmer à la fois le frontend et le backend
- Son environnement d’exécution est node.js et les navigateurs
- Node est parmi les technologies les plus populaires dans le domaine du web (voir ici)

Variables et leur portée en javascript

- Type var
  - ✓ Portée se limite à la fonction sinon globale
  - ✓ Peut-être redéclaré et peut être utilisé avant déclaration et sans initialisation retourne undefined
  - ✓ Eviter son utilisation si possible
- Type let
  - ✓ Portée se limite au bloc ou fonction
  - ✓ Ne peut pas être redéclaré mais peut être réassigné
- Type const
  - ✓ Mêmes propriétés que let
  - ✓ Utilisé pour les constantes et ne peut pas être réassigné

	var	let	const
Portée globale	oui	non	non
Portée fonction	oui	oui	oui
Portée bloc	non	oui	oui
Redéclarable	oui	non	non
Rassignable	oui	oui	non

Variables et leur portée en javascript

Quelques exemples

```
//Remontée ou hissage (hoisting)
console.log("premier texte", salutation)
var salutation = "Un simple bonjour"

//Portée
function saluer(){
  var salut = "Un autre salut"
}
console.log(salut)

//Re-déclaration
var message = "premier contenu"
var nombre=6
if(nombre>4){
   var message = "second message"
}
console.log(message)
```

```
//Remontée ou hissage
console.log("premier texte", salutation)
let salutation = "Un simple bonjour"

//Portée
function saluer(){
  let salut = "Un autre salut"
}
console.log(salut)

//Re-assignation
let message = "premier contenu"
let nombre=6
if(nombre>4){
   let message = "second message" //sans let
}
console.log(message)
```

Les instructions conditionnelles

Elles sont de la forme

```
//forme générale

  if (condition1) {
     instruction_1;

  } else if (condition2) {
     instruction_2;

  } else {
     instruction_3;
  }
```

Les conditions sont des expressions booléennes de la forme (pour a et b deux variables)

- $a === b$, $a\ !== b$ pour tenir compte à la fois du type et des valeurs de a et b
- $a == b$, $a\ != b$ pour tenir compte seulement des valeurs de a et b
- $a > b$, $a < b$, $a >= b$, $a <= b$, $a\ ||\ b$, $a\ \&\&\ b$
- $!a$ la négation et $!!a$ conversion en booléen
- Pour les objets obj?.x pour l'existence de x

Les instructions conditionnelles

Operateur ternaire ? :

- Version courte et convenable pour des expressions
- Quelques exemples

```
//forme générale
const output = condition ? valeur1 : valeur2

//forme générale pour plusieurs conditions
const output = condition1 ? valeur1 : condition2 ? valeur2 : valeur3
```

```
const age = 18

const output1 = age < 18 ? "un mineur" : "un majeur"

const output2 = age < 18 ? "un mineur" : age > 18 ? "un majeur" : "pile au milieu"
```

Les boucles

Il existe différents types de boucles en javascript

- Boucle do…while
- Boucle while
- Boucles for
  - ✓for avec les itérations
  - ✓for …in
  - ✓for …of

Les boucles

Les boucles while et do …while

- Boucle while

```
//forme générale

  while (condition) {
     instruction;

  }
```

```
//exemple avec itération

 let i = 0
 while (i < 5) {
     alert(`La valeur est ${i}!`)
     i++
   }
```

```
//exemple avec entrée de l'utilisateur

 let nombre = prompt('Donner une valeur')

 while (nombre > 15) {

     alert(`${nombre} est trop grand!`)
     nombre = prompt('Donner une autre valeur')
   }

 alert(`${nombre} est correct!`)
```

Les boucles

Les boucles while et do …while

- Boucle do …while

```
//forme générale

  do {
     instruction;

  } while (condition)
```

```
//exemple avec itération
 let i = 0

 do (i < 5) {
     alert(`La valeur est ${i}!`)
     i++
  } while (i < 5)
```

```
//exemple avec entrée de l'utilisateur

 let nombre = prompt('Donner une valeur')

 do {

     alert(`${nombre} est trop grand!`)
     nombre = prompt('Donner une autre valeur')
  } while (nombre > 15)

 alert(`${nombre} est correct!`)
```

Les boucles

Il existe trois types de boucles for

- En utilisant les indices/itérations

```
//forme générale

   for(let i=0; i<condition; i++){
      instruction
    }
```

```
//exemple avec itération

  for(let i=0; i<10; i++){

       alert(`Iteration ${i}`)
  }
```

```
//Exemple sur un tableau (array)

const notreArray=["habits","cahiers","livres","chaussures"]

//Affichage des éléments du tableau

 for (let i=0; i<notreArray.length;i++){

        if(i>=2) break  //Arrêter une boucle

        if(i==0) continue  //Sauter une itération

        alert(`Element no ${i} = ${notreArray[i]}`)
  }
```

Les boucles

Il existe trois types de boucles for

➢En utilisant for …of

- ✓Principalement avec les tableaux
- ✓Possible d'avoir l'indice

```
//forme générale

   for(let element of elements){
      instruction
    }
```

```
//Exemple sur un tableau (array)

const notreArray=["habits","cahiers","livres","chaussures"]

//Affichage des éléments du tableau

 for (let elem of notreArray){

        let indice = notreArray.indexOf(elem)

        alert(`Element no ${indice} = ${elem}`)
  }
```

Les boucles

Il existe trois types de boucles for

➢En utilisant for …in

- ✓Principalement avec les objets
- ✓Clefs retournées pour les objets
- ✓Indices retournés pour les tableaux

```
//forme générale

   for(let champ in object){
      instruction
    }
```

```
//Exemple sur un objet

  const person={
            nom:"john",
            age:30,
            simple:false
    }
//Parcourir les clefs de l'objet

   for (let field in person) {
         alert(`person[${field}]  = ${person[field]}`)

    }
```

Fonctions mathématiques

Les fonctions mathématiques sont dans l'objet Math

➢Quelques exemples

```
//Exemple de fonction
  Math.abs()

  Math.sin()

  Math.random()

//Exemple de constante

  const PI = Math.PI
  alert(`PI vaut ${Math.PI}`)
```

Chaines de caractères

Les chaines des caractères sont de la forme

```
//Exemples
 const text='Bonjour et bienvenu’

 const autreText="L'enfant s'est endormi!"

// Cas rare
  const unText = new String('Texte avec String')

//string literals (sur plusieurs lignes ou avec des
variables dynamiques
const nbr = 3

const unText = `Ce texte est
                sur ${nbr} lignes
                et continue encore`
alert(unText)
```

Quelques méthodes très utiles

```
//Quelques méthodes utiles

 const text='Bonjour et bienvenu’

  text.charAt(0)
  text.startsWith('B')
  text.endsWith('u')
  text.toLowerCase()
  text.toUpperCase()
  text.slice(3,8)
  text.split(' ')
  text.includes('et')
  text.trim()
```

Expressions régulières ou Regex

- Beaucoup utilisées pour les validations de formulaires
- Présentes dans plusieurs langages
- Les règles de formations sont données a ce lien

```
//Exemple de regex

const nomRegex=/^[a-zA-Z]$/  // Seulement des
caractères alpha

const nomRegex=/^[a-zA-Z]{5,}$/  // Au moins 5
caractères alpha

//Exemple de constante
```

Quelques méthodes très utiles

```
//Quelques méthodes utiles

 const nomRegex=/^[a-zA-Z]$/

  nomRegex.test('hello')
  text.match('B')
  text.search('u')
```

Conclusion

- Comprendre les bases de javascript
- Les variables et leur portée
- Savoir écrire de petits programmes et les exécuter
- Continuer à apprendre ici

Exercices

Petit jeu pour deviner un nombre

- Un nombre fixe est donné initialement (entre 0 et 10)
- L’utilisateur doit entrer un nombre au hasard (popup interactif)
- Le nombre entré doit être un vrai nombre (conditions)
- Le nombre entré est comparé au nombre fixe et l’utilisateur continue à deviner le nombre tant qu’il ne le trouve pas (boucle)
- A chaque fois, on dira a l’utilisateur si le nombre choisi est plus grand ou plus petit que le nombre fixe (popup)

Exercices

Connexion à une page web

- Les noms des personnes autorisées sont dans un tableau prédéfini
- Chaque personne appartient à une équipe numérotée de 1 à 5 à deviner
- Dès l’ouverture de la page, l’utilisateur doit entrer son numéro d’équipe
  - si le numéro est incorrect ou non valide, il voit un message « accès refusé »
- Une fois l’équipe, il doit entrer son nom,
  - si le nom est présent, un message de bienvenu lui est adresse avec son nom complet
  - Sinon un message disant « accès refusé» lui est affiché