Javascript
Partie 4

Java Script

Les événements

`click`	Cliquer (appuyer puis relâcher) sur l'élément
`dblclick`	Double-cliquer sur l'élément
`mouseover`	Faire entrer le curseur sur l'élément
`mouseout`	Faire sortir le curseur de l'élément
`mousedown`	Appuyer (sans relâcher) sur le bouton gauche de la souris sur l'élément
`mouseup`	Relâcher le bouton gauche de la souris sur l'élément
`mousemove`	Faire déplacer le curseur sur l'élément

Java Script

Les événements

`keydown`	Appuyer (sans relâcher) sur une touche de clavier sur l'élément
`keyup`	Relâcher une touche de clavier sur l'élément
`keypress`	Frapper (appuyer puis relâcher) une touche de clavier sur l'élément
`focus`	« Cibler » l'élément
`blur`	Annuler le « ciblage » de l'élément
`change`	Changer la valeur d'un élément spécifique aux formulaires (`input`, `checkbox`, etc.)
`select`	Sélectionner le contenu d'un champ de texte (`input`, `textarea`, etc.)

Les événements

Spécifique à <form>

`submit`	Envoyer le formulaire
`reset`	Réinitialiser le formulaire

Java Script

Les événements (exemple)

```
<span onclick="alert('Hello !');">Cliquez-moi !</span>
```

```
<span onclick="alert('Voici le contenu de l\'élément que vous avez
cliqué :\n\n' + this.innerHTML);">Cliquez-moi !</span>
```

```
   <input id="input" type="text" size="50" value="Cliquez ici !"
onfocus="this.value='Appuyez maintenant sur votre touche de tabulation.';"
onblur="this.value='Cliquez ici !';"/>
   <br /><br/>
   <a href="#" onfocus="document.getElementById('input').value = 'Vous avez
maintenant le focus sur le lien, bravo !';">Un lien bidon</a>
```

Java Script

Les événements (exemple)

```
    <a href="http://blog.crdp-versailles.fr/rimbaud/" onclick="alert('Vous avez
cliqué!');">Cliquez-moi !</a>
```

```
    <a href="http://blog.crdp-versailles.fr/rimbaud/" onclick="alert('Vous avez
cliqué !'); return false;">Cliquez-moi !</a>
```

```
<a href="#" onclick="alert('Vous avez cliqué !'); return false;">
Cliquez-moi !
</a>
```

Java Script

Les Formulaires

Dans , on utilise la propriété value

```
<input id="text" type="text" size="60" value="Vous n'avez pas le
focus !" />
<script>
   var text = document.getElementById('text');
   text.addEventListener('focus', function(e) {
         e.target.value = "Vous avez le focus !";
   }, true);
   text.addEventListener('blur', function(e) {
         e.target.value = "Vous n'avez pas le focus !";
   }, true);
</script>
```

Les Formulaires

Pour désactiver un champ de texte :

```
<input id="text" type="text" />
<script>
   var text = document.getElementById('text');
   text.disabled = true;
</script>
```

Java Script

Les Formulaires

On peut utiliser checked pour des boutons radio :

```
<label><input type="radio" name="check" value="1" />Case n°1</label><br />
<label><input type="radio" name="check" value="2" />Case n°2</label><br />
<label><input type="radio" name="check" value="3" />Case n°3</label><br />
<label><input type="radio" name="check" value="4" />Case n°4</label>
<br /><br />
<input type="button" value="Afficher la case cochée" onclick="check();" />
<script>
  function check() {
  var inputs = document.getElementsByTagName('input'),
  inputsLength = inputs.length;
  for (var i = 0 ; i < inputsLength ; i++) {
        if (inputs[i].type == 'radio' && inputs[i].checked) {
        alert('La case cochée est la n°'+ inputs[i].value);
        }
  }
  }
</script>
```

Java Script

Les Formulaires

selectedIndex pour des listes déroulantes :

```
<select id="list">
  <option>Sélectionnez votre sexe</option>
  <option>Homme</option>
  <option>Femme</option>
</select>
<script>
  var list = document.getElementById('list');
  list.addEventListener('change', function() {
  // On affiche le contenu de l'élément <option> ciblé par la propriété
selectedIndex
        alert(list.options[list.selectedIndex].innerHTML);
  }, true);
</script>
```

Java Script

Les Formulaires

L'élément possède deux méthodes intéressantes : submit() pour effectuer l'envoi du formulaire, reset() pour réinitialiser le formulaire :

```
<form id="myForm">
  <input type="text" value="Entrez un texte" /><br /><br />
  <input type="submit" value="Submit !" />
  <input type="reset" value="Reset !" />
</form>
<script>
  var myForm = document.getElementById('myForm');
  myForm.addEventListener('submit', function(e) {
        alert('Vous avez envoyé le formulaire !\n\nMais celui-ci a été bloqué
pour que vous ne changiez pas de page.');
        e.preventDefault();
  }, true);
  myForm.addEventListener('reset', function(e) {
        alert('Vous avez réinitialisé le formulaire !');
  }, true);
```

Les Formulaires

Les méthodes focus() et blur() pour donner ou retirer le focus sur un événement.

```
<input id="text" type="text" value="Entrez un texte" /><br /><br />
<input type="button" value="Donner le focus"
  onclick="document.getElementById('text').focus();" /><br />
<input type="button" value="Retirer le focus"
  onclick="document.getElementById('text').blur();" />
```

Les Formulaires

Avec select(), on sélectionne le texte :

```
<input id="text" type="text" value="Entrez un texte" /><br /><br />
<input type="button" value="Sélectionner le texte"
  onclick="document.getElementById('text').select();" />
```

Tableaux et Caractères

Construire un tableau :

```
function Person(nick, age, sex, parent, work, friends) {
   this.nick = nick;
   this.age = age;
   this.sex = sex;
   this.classe = classe;
   this.work = work;
   this.friends = friends; }
```

On met la première lettre de la fonction du constructeur en majuscule. On utilise this pour construire, ce qui permet de définir les propriétés de la fonction Person dans l'exemple.

Java Script

Tableaux et Caractères

Construire un tableau : ajouter des variables :

```
var seb = new Person('Rafael', 15, 'm', '3e 5', 'Javascripteur', []);
var lau = new Person('Mathilde', 12, 'f', '5e 6', 'Webmaster', []);
alert(seb.nick); // Affiche : « Rafael »
alert(lau.nick); // Affiche : « Mathilde »
```

changer des variables :

```
var seb = new Person('Rafael', 15, 'm', '3e 5', 'Javascripteur', []);
seb.nick = 'Bastien'; // On change le prénom
seb.age = 14; // On change l'âge
alert(seb.nick + ' a ' + seb.age + 'ans'); // Affiche : « Bastien a 14 ans »
```

Java Script

Tableaux et Caractères

ajouter des méthodes, par exemple un « ami » dans le tableau :

```
var seb = new Person('Rafael', 15, 'm', '3e 5', 'Javascripteur', []);
// On ajoute un ami dans le tableau « friends »
seb.friends.push(new Person('Jérôme', 13, 'm', '3e 5',
'Javascripteur aussi', []));
alert(seb.friends[0].nick); // Affiche : « Jérôme »
```

```
function Person(nick, age, sex, parent, work, friends) {
   this.nick = nick;     this.age = age;       this.sex = sex;
   this.parent = parent; this.work = work;     this.friends = friends;
   this.addFriend = function(nick, age, sex, parent, work, friends)
          {
          this.friends.push(new Person(nick, age, sex, parent, work, friends));
          };     }
```

Java Script

Tableaux et Caractères

Gérer un tableau :

On utilise l'objet Array :

```
var myArray = new Array('valeur1', 'valeur2', …, 'valeurX');
```

méthode concat() pour concaténer :

```
var myArray = ['test1',
'test2'].concat(['test3', 'test4']);
alert(myArray); //retourne ['test1', 'test2',
'test3', 'test4']
```

Tableaux et Caractères

Gérer un tableau :

foreach() pour parcourir :

```
var myArray = ["C'est", "un", "test"];
myArray.forEach(function(value, index, array)
{
    alert(
    'Index : ' + index
    + '\n' +
    'Valeur : ' + value
    );
});
```

Tableaux et Caractères

Gérer un tableau :

indexOf() pour rechercher un élément :

```
var element2 = ['test'],
myArray = ['test', element2];
alert(myArray.indexOf(element2)); // Affiche : 1
```

Tableaux et Caractères

Gérer un tableau :

sort() pour ordonner alphabétiquement :

```
var myArray = [3, 1, 5, 10, 4, 2];
myArray.sort();
alert(myArray); // Affiche : 1,10,2,3,4,5
```

Les Chaines de caractères

- On supprime les espaces d'une chaîne de caractères avec trim()
- On fait des recherches dans une chaîne avec indexOf()
- On extrait une chaîne avec substring(), substr() et slice()
- On coupe une chaîne en un tableau avec split()

Expression régulières

Cela permet de rechercher des mots :

```
var myRegex = /contenu_à_rechercher/;
var regex_2 = /contenu_\/_contenu/;
```

```
if (/dragon/.test('Il a tué le dragon qui garde le trésor !')) {
        alert('Ça semble parler de dragon');
  } else {
        alert('Pas de dragon à l\'horizon');      }
```

```
if (/Dragon|Licorne/i.test('Il a ensuite chevauché la licorne !')) {
        alert('Ça semble parler de trucs fantastiques');
  } else {
        alert('Pas de fantastique à l\'horizon');        }
```

Java Script

Expression régulières

- On peut convenir d'un intervalle de lettre (de a à j : [a-j]), ou encore [a-zA-Z] pour ne pas tenir compte de la casse, ou [0-9] pour tous les chiffres, ou encore [a-z0-9].
- On peut ignorer des lettres avec [^aeiou] ou [^b-f]. Mais il faut préciser les caractères accentuées, et utiliser i pour la casse (qui fonctionne aussi pour les caractères accentués) : [^a-zâäàéèùêëîïôöçñ]/i. Le point désigne un caractère quelconque /gr.s/
- On peut utiliser des symboles quantificateurs : ? : ce symbole indique que le caractère qui le précède peut apparaître 0 ou 1 fois ; + : ce symbole indique que le caractère qui le précède peut apparaître 1 ou x fois ; * : ce symbole indique que le caractère qui le précède peut apparaître 0, 1 ou x fois.
- On peut préciser plutôt le nombre de fois : {n} : le caractère est répété n fois ; {n,m} : le caractère est répété de n à m fois. Par exemple, si on a {0, 5}, le caractère peut être présent de 0 à 5 fois ; {n,} : le caractère est répété de n fois à l'infini.

Expression régulières

Pour une adresse mail :

```
var email = prompt("Entrez votre adresse e-mail :","0780506b@ac-versailles.fr");
if (/^[a-z0-9._-]+@[a-z0-9._-]+\.[a-z]{2,6}$/.test(email)) {
        alert("Adresse e-mail valide !");
   } else {
        alert("Adresse e-mail invalide !"); }
```

Les données numériques

L'objet Number possède des propriétés accessibles directement sans aucune instanciation (on appelle cela des propriétés propres à l'objet constructeur). Elles sont au nombre de cinq, leur usage est peu courant :

- NaN : cette propriété signifie Not A Number et permet, généralement, d'identifier l'échec d'une conversion de chaîne de caractères en un nombre.
- MAX_VALUE : cette propriété représente le nombre maximum pouvant être stocké dans une variable en Javascript.
- MIN_VALUE : identique à la constante MAX_VALUE, mais pour la valeur minimale.
- POSITIVE_INFINITY : il s'agit ici d'une constante représentant l'infini positif. On peut l'obtenir en résultat d'un calcul si on divise une valeur positive par 0.
- NEGATIVE_INFINITY : identique à POSITIVE_INFINITY, pour l'infini négatif. On peut obtenir cette constante en résultat d'un calcul si on divise une valeur négative par 0

Java Script

La gestion du temps

l'objet Date :

```
new Date();
new Date(timestamp);
new Date(dateString);
new Date(année, mois, jour [, heure, minutes, secondes, millisecondes ]);
```

Huit méthodes :

`getFullYear()` : renvoie l'année sur 4 chiffres ;
`getMonth()` : renvoie le mois (0 à 11) ;
`getDate()` : renvoie le jour du mois (1 à 31) ;
`getDay()` : renvoie le jour de la semaine (0 à 6, la semaine commence le dimanche) ;
`getHours()` : renvoie l'heure (0 à 23) ;
`getMinutes()` : renvoie les minutes (0 à 59) ;
`getSeconds()` : renvoie les secondes (0 à 59) ;
`getMilliseconds()` : renvoie les millisecondes (0 à 999).
`getTime()` renvoie le timestamp de la date de votre objet ;
`setTime()` vous permet de modifier la date de votre objet en passant en unique paramètre un timestamp.

La gestion du temps

```
var myDate = new Date('Sat, 04 May 1991 20:00:00 GMT+02:00');
alert(myDate.getMonth()); // Affiche : 4
alert(myDate.getDay()); // Affiche : 6
```

La gestion du temps

intervalle avant l'exécution d'une fonction avec setTimeout() :

```
setTimeout(myFunction, 2000); // myFunction sera exécutée au bout de 2 secondes
```

Java Script

La gestion du temps

annule une fonction temporelle avec clearTimeout() ou clearInterval() :

```
<button id="myButton">Annuler le compte à rebours</button>
<script>
  (function() {
  var button = document.getElementById('myButton');
  var timerID = setTimeout(function() { // On crée notre compte à rebours
        alert("Vous n'êtes pas très réactif vous !");
  }, 5000);
  button.onclick = function() {
  clearTimeout(timerID); // Le compte à rebours est annulé
alert("Le compte à rebours a bien été annulé."); // Et on prévient l'utilisateur
}; })(); </script>
```

Java Script

La gestion du temps

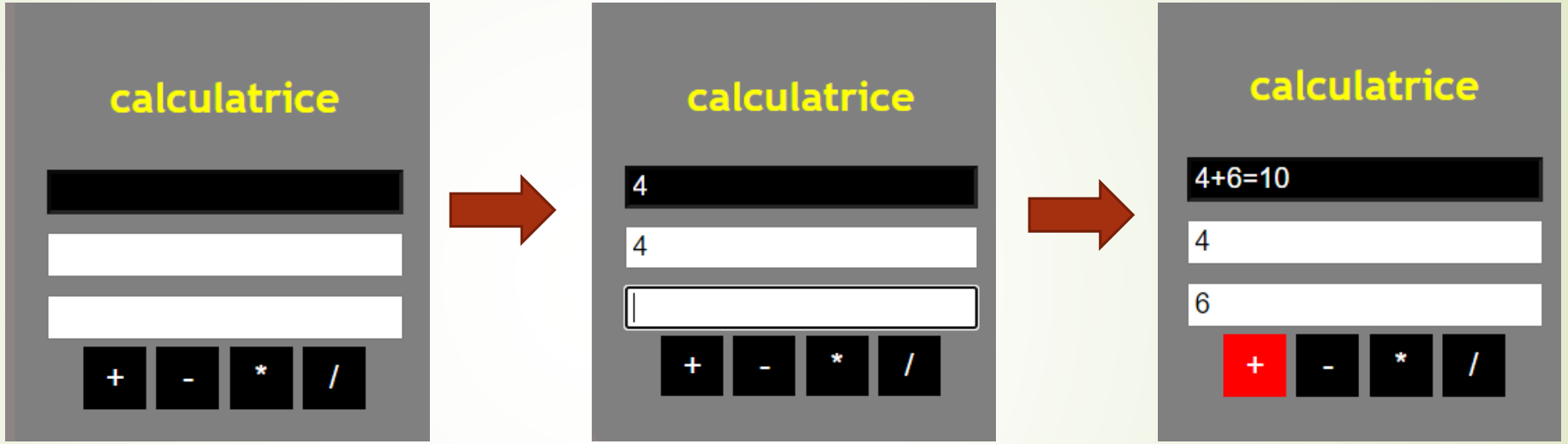
gérer la durée pour des animations :

```
<img id="myImg" src="images/circuit_html.jpg" style="opacity: 1" />
<script>
  var myImg = document.getElementById('myImg');
  function anim() {
        var s = myImg.style,
        result = s.opacity = parseFloat(s.opacity) - 0.1;
        if ( result > 0.2 ) {
              setTimeout(anim, 50); // La fonction anim() fait appel à elle-
même si elle n'a pas terminé son travail
              }
        }
  anim(); // Et on lance la première phase de l'animation
</script>
```

Java Script

Exercice

création d 'une calculatrice (simple)



Javascript

Fin de la présentation

Partie 4