Gestionnaires d'événement

Les événements

Les événements sont des actions ou des occurrences qui se produisent dans le système que vous programmez.

Gestionnaire d'événement

Gestionnaire d'événement (event handler) c’est un bloc de code (généralement une fonction JavaScript définie par l'utilisateur) qui sera exécuté lorsque l'événement se déclenchera.

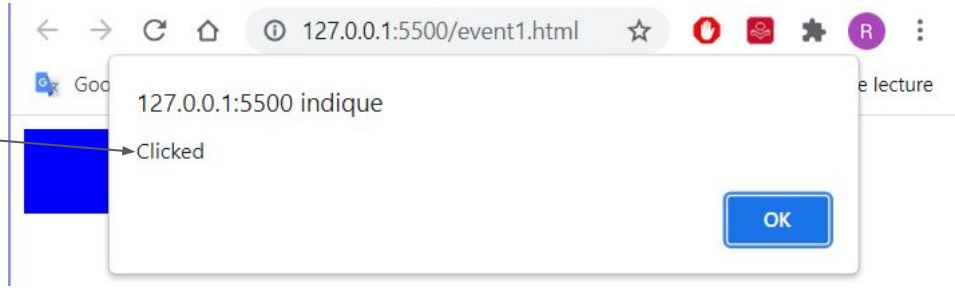
```
function displayMessage(){
    alert('Clicked');
}
```

```
alert('Clicked')
```

event1.html

```html
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8" />
    <style>
        #rect{
            width:50px;
            height:50px;
            background-color:blue;
        }
    </style>
</head>
<body>
<div id="rect"
onclick="displayMessage()"></div>
<script>
function displayMessage(){
    alert('Clicked');
}
</script>
</body>
</html>
```

Enregistrement d'un gestionnaire d'événements (registering an event handler) permet de déclencher une fonction (**gestionnaire d'événement**) en réponse à des actions (par exemple, un clic de souris sur un élément)

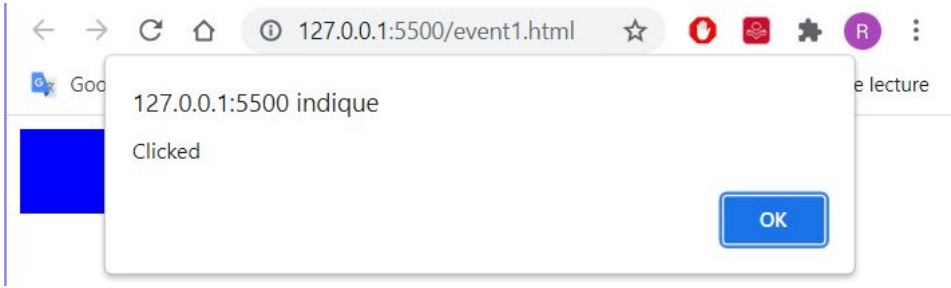


event1.html

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8" />
    <style>
        #rect{
            width:50px;
            height:50px;
            background-color:blue;
        }
    </style>
</head>
<body>
<div id="rect"
onclick="displayMessage()"></div>
<script>
function displayMessage(){
    alert('Clicked');
}
</script>
</body>
</html>
```

Cette manière d'enregistrement d'un gestionnaire d'événement s'appelle **un enregistrement de gestionnaire d'événement en ligne**

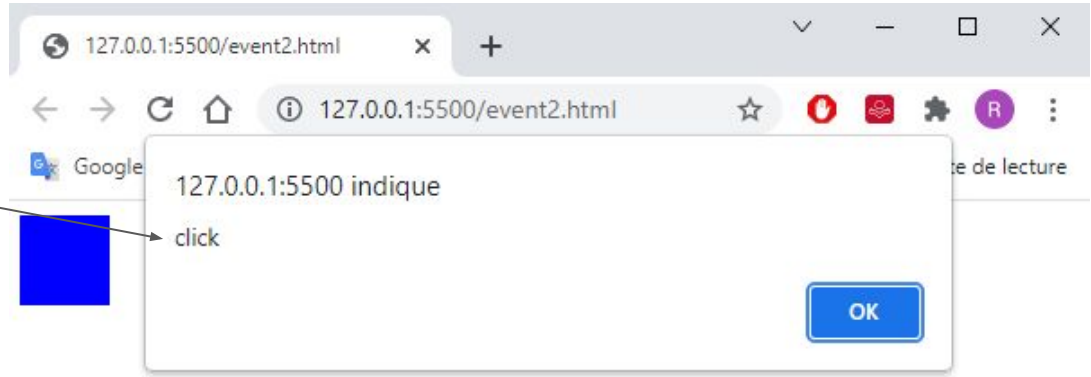
Ajout d'un attribut du tag HTML avec nom: on +"type d'événement"



event2.html

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8" />
    <style>
    #rect{
        width:50px;
        height:50px;
        background-color :blue;
    }
    </style>
</head>
<body>
<div id="rect"
onclick="handler(event)"></div>
<script>
function handler(e){
    alert(e.type);
}
</script>
</body>
</html>
```

Event c'est un objet qui contient de l'information d'événement. (par exemple type d'événement), c'est le nom **réservé.**



Le gestionnaire d'événement en ligne a plusieurs problèmes:

1. Html et JavaScript confondus
2. Impossible d’enregistrer un gestionnaire d'événement pour des éléments HTML créés dynamiquement
3. Un événement a juste un enregistrement d’un gestionnaire d'événement
4. Impossible de supprimer un enregistrement d’un gestionnaire d'événement, sans changement de code

event3.html

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8" />
    <style>
    #rect{
        width:50px;
        height:50px;
        background-color:blue;
    }
    </style>
</head>
<body>
<div id="rect"></div>
<script>
function handler(e){

    alert(e.type);
}
document.getElementById("rect").onclick = handler;
</script>
</body>
</html>
```

La propriété du gestionnaire d'événement (Event handler property) résout certains problèmes (2/4):

1. HTML et JavaScript sont mieux séparés
2. Un gestionnaire d'événement peut être enregistré pour des éléments HTML créés dynamiquement

Event envoyé automatiquement

Sélection de l'élément DOM et ajout de la propriété de cet élément avec nom: on +"type d'événement"

event4.html

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8" />
    <style>
    #rect{
        width:50px;
        height:50px;
        background-color:blue;
    }
    </style>
</head>
<body>
<div id="rect"></div>
<script>
function handler(e){
    alert(e.type);
}
document.getElementById("rect")
.addEventListener("click", handler);

</script>
</body>
</html>
```

Les écouteurs (EventListener)
Utilisation de l'interface **EventTarget** avec sa méthode **addEventListener()** résout tous les problèmes(4/4).

Tous les elements DOM sont des objects EventTarget.

Fonction **addEventListener()** accepte 2 paramètres: "type d'événement" et une fonction ou nom de la fonction.

on peut supprimer enregistrement d'un gestionnaire d'événement:

```
document.getElementById("rect")
.removeEventListener("click", handler);
```

on peut enregistrer plusieurs gestionnaires d'événement:

```
document.getElementById("rect")
.addEventListener("click", handler2);
function handler2(e){
    var newNode = document.createElement("p");
    newNode.textContent = "Clicked";
    document.body.appendChild(newNode);
}
```

Types des enregistrements de gestionnaire d'événement

```
<div id="rect"
onclick="handler(event)">
</div>
<script>
function handler(e){
    alert(e.type);
}
</script>
```

1. **le gestionnaire d'événement en ligne (inline event handler)**

Ajout d'un attribut du tag HTML avec nom: on +"type d'événement"

```
<div id="rect"></div>
<script>
function handler(e){
    alert(e.type);
}
document.getElementById("rect")
.onclick = handler;
</script>
```

2. **La propriété du gestionnaire d'événement (Event handler property)**

Sélection de l'élément DOM et ajout de la propriété de cet élément avec nom: on +"type d'événement"

```
<div id="rect"></div>
<script>
function handler(e){
    alert(e.type);
}
document.getElementById("rect")
.addEventListener("click", handler);
</script>
```

3. **Les écouteurs (EventListener)**
Utilisation de l'interface **EventTarget** avec sa méthode **addEventListener().**
Tous les elements DOM sont des objects EventTarget.
Fonction addEventListener() accepte 2 paramètres: "type d'événement" et une fonction ou nom de la fonction.

! **Event** envoyé automatiquement

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8" />
</head>
<body>
    <a href="http://google.com" id="link">Google pour les
enfants</a>
    <script>
    function linkHandler(e){

        var date = new Date();
        var hour = date.getHours();
        console.log(hour);
        if(hour>20){

            e.preventDefault();
            var newDiv = document.createElement("div");
            newDiv.innerHTML = "<h1>C'est l'heure de se
coucher!</h1>";

document.getElementById("link").parentElement.appendChild(n
ewDiv);
        }
    }
    var link = document.getElementById("link");
    link.addEventListener("click", linkHandler);
    </script>
</body>
</html>
```

méthode preventDefault() de object Event permet d'arrêter l'exécution de l'événement

Ce script interdit de passer sur le lien google après 20H