

JavaScript dans les liens

Code JavaScript dans l'attribut *href*

Une URL peut désigner du code JavaScript. Le principe est simple, mettez le préfixe *javascript:* suivi de votre code JavaScript. Exemple avec le fameux lien `<a href...>` :

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
  </head>
  <body>
    <a href="javascript:alert('coucou');">cliquez ici</a>
  </body>
</html>
```

On peut bien entendu appeler une fonction :

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <script>
      function toto() {
        alert("Coucou");
        alert("Tu vas bien ?");
      }
    </script>
  </head>
  <body>
    <a href="javascript:toto()">cliquez ici</a>
  </body>
</html>
```

Code JavaScript dans l'attribut *action* du formulaire

Le lien JavaScript peut être utilisé comme action d'un formulaire, puisque l'attribut *action* de la balise `<form>` sert à appeler un lien suite à la soumission du formulaire.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <script>
      function message() {
        alert("Coucou");
        alert("Tu vas bien ?");
      }
    </script>
  </head>
  <body>
    <form name="saisie" action="javascript:message();">
      <input type="submit">
    </form>
  </body>
</html>
```

On peut en profiter pour améliorer un peu les choses en récupérant les données du formulaire :

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <script>
      function reponse() {
        var prenom;
        prenom=document.getElementById("PrenomSaisi").value;
        alert("Bonjour " + prenom);
      }
    </script>
  </head>
  <body>
    <form name="saisie" action="javascript:reponse();">
      <label>Entrez votre prénom</label>
      <input type="text" id="PrenomSaisi">
      <input type="submit">
    </form>
  </body>
</html>

```

La propriété *value* représente le contenu du champ de texte nommé *PrenomSaisi* identifié par la méthode *getElementById()*.

Variante sans préfixe *javascript*: avec l'événement *onsubmit()*

Le script précédent peut avoir cette variante sans URL grâce à l'événement JavaScript *onsubmit()*. Il n'est alors pas nécessaire d'utiliser le préfixe *javascript*: car un événement JavaScript ne sert qu'à appeler du code JavaScript, contrairement à l'attribut *action* ou *href* qui peuvent relier n'importe quel code au format HTML, PHP, CGI, etc.

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <script>
      function reponse() {
        var prenom;
        prenom=document.getElementById("nomSaisi").value;
        alert("Bonjour " + prenom);
      }
    </script>
  </head>
  <body>
    <form name="saisie" onsubmit="reponse()">
      <label>Entrez votre prénom</label>
      <input type="text" id="nomSaisi">
      <input type="submit">
    </form>
  </body>
</html>

```

Pourquoi ne pas utiliser l'événement *onclick* sur le bouton, plutôt que l'événement *submit* sur le formulaire ? Parce que l'événement *onclick* ne détecte que le clic sur le bouton, alors que *submit* détecte non seulement le clic, mais aussi l'appui sur la touche Entrée suite à la saisie du prénom.

Encore plus simple, l'événement *onsubmit* permet de se passer du bouton submit. Il suffira d'appuyer sur la touche *Entrée* suite à la saisie du prénom pour exécuter la réponse JavaScript.

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <script>
      function reponse() {

```

```
        var prenom;
        prenom=document.getElementById("nomSaisi").value;
        alert("Bonjour " + prenom);
    }
</script>
</head>
<body>
    <form name="saisie" onsubmit="reponse()">
        <label>Entrez votre prénom</label>
        <input type="text" id="nomSaisi">
    </form>
</body>
</html>
```

Voici donc quelques variantes de code, qui je l'espère, vous aideront à mieux comprendre les interactions entre des liens, des formulaires et le JavaScript.