

La méthode `getElementById`

La méthode JavaScript `getElementById()` permet de récupérer les informations d'une balise identifiée par son *id*.

Syntaxe

informations à récupérer = `document.getElementById("nom");`

- *document* représente la page HTML, plus précisément l'intérieur de la fenêtre du navigateur : la zone d'affichage sans la barre d'adresse et les boutons de navigation.
- *nom* désigne la valeur de l'attribut *id* d'une balise unique située dans la page.

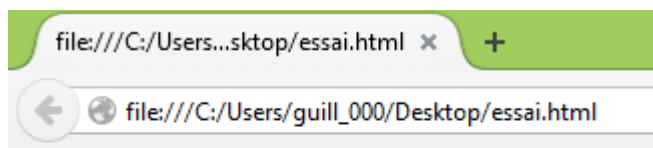
Changer les styles CSS avec `getElementById().style`

Une fonctionnalité extrêmement puissante du JavaScript consiste à modifier les styles CSS avec la méthode `getElementById().style`.

Ci-dessous un script pour changer la couleur du texte en cliquant sur des boutons grâce à la propriété CSS *color* :

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <script>
      function changeCouleur(nouvelleCouleur) {
        document.getElementById("paragraphe").style.color = nouvelleCouleur;
      }
    </script>
  </head>
  <body>
    <p id="paragraphe">Un peu de texte</p>
    <button onclick="changeCouleur('blue');">blue</button>
    <button onclick="changeCouleur('red');">red</button>
  </body>
</html>
```

Résultat si on appuie sur le bouton *red* :



Un peu de texte

blue red

L'exemple suivant permet de changer la couleur d'arrière-plan d'une balise suite à un clic sur un bouton.

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

```

        document.getElementById("titre").style.backgroundColor = "#FF0000";
    }
</script>
</head>
<body>
    <h1 id="titre">La couleur de fond du titre va changer au clic sur le
bouton</h1>
    <button onclick="changefond()">Change la couleur</button>
</body>
</html>

```

Beaucoup de menus réalisés en JavaScript utilisent la propriété *visibility* pour masquer ou démasquer des balises. Cet exemple pourrait également utiliser la propriété *display* avec les valeurs *none* et *block*.

```

<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <script>
            // La fonction qui affiche ou masque l'élément
            fonction bascule(elem) {
                // Quel est l'état actuel ?
                let etat = document.getElementById(elem).style.visibility;
                if (etat == "hidden")
                    document.getElementById(elem).style.visibility = "visible";
                else
                    document.getElementById(elem).style.visibility = "hidden";
            }
        </script>
    </head>
    <body>
        <input type="button" onclick="bascule('boite')" value="On/Off">
        <div id="boite" style="visibility:hidden">
            <h1>Coucou, je suis là !</h1>
            <p>Et moi aussi !</p>
        </div>
    </body>
</html>

```

Les propriétés CSS qui suivent l'expression *getElementById(elem).style* s'écrivent en minuscules si la propriété est en un seul mot, et en « minusculeMajuscule » si la propriété est un mot composé. Par exemple, on écrira *borderColor* pour la propriété CSS *border-color*. On peut de cette manière modifier la plupart des propriétés CSS, ce qui laisse imaginer les immenses possibilités du JavaScript grâce aux modifications des CSS.

Toutes les propriétés CSS modifiables sur : http://www.w3schools.com/jsref/dom_obj_style.asp

La propriété *innerHTML*

Syntaxe

```
document.getElementById('exemple').innerHTML = "Code HTML à écrire";
```

La propriété *innerHTML* permet de remplacer complètement le contenu d'une balise identifiée par son attribut *id*. Pour faire simple, elle permet d'écrire à un endroit précis de la page, sans tout effacer contrairement à *document.write*.

Exemples simples

Cet exemple appelle une fonction qui effectue un calcul et inscrit le résultat dans la page :

```

<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">

```

```

</head>
<body>
  <p id="exemple">Cliquez sur le bouton et ce texte changera.</p>
  <input type="button" value="clic" onclick="changetexte()">
  <script>
    function changetexte() {
      document.getElementById("exemple").innerHTML = "Texte changé !";
    }
  </script>
</body>
</html>

```

Cet exemple appelle une fonction qui effectue un calcul puis inscrit le résultat dans une balise <p> :

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
  </head>
  <body>
    <p id="exemple"></p>
    <script>
      function superFonction(a, b) {
        return a * b;
      }
      document.getElementById("exemple").innerHTML = superFonction(4, 3);
    </script>
  </body>
</html>

```

Effet Lightbox sans jQuery

Ici on peut éviter les accumulations de fenêtres popup en jouant avec la propriété CSS *z-index*. L'image agrandie est pré chargée sans être affichée grâce à la propriété CSS *display: none*.

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <script>
      function zoom() {
        document.getElementById("agrandir").innerHTML = "<img src='grande.jpg'
style='position:absolute; top:50px; left:50px; z-index:1; cursor:pointer' onclick='dezoom()'
title='Fermer'>";
      }
      function dezoom() {
        document.getElementById("agrandir").innerHTML = "";
      }
    </script>
  </head>
  <body>
    
    <p>New York Times Square New York Times Square New York Times Square New York
Times Square New York Times Square New York Times Square New York </p>
    <div id="agrandir"></div>
    
  </body>
</html>

```

La machine à écrire de Michel Audiard

Pour finir en beauté, voici une petite mécanique assez complexe d'insertion de texte au sein de la balise HTML <q> avec un effet d'animation similaire à une « machine à écrire ».

- Les textes de Michel Audiard sont écrits caractère par caractère, grâce à la méthode `setInterval()` qui appelle la fonction `machineEcrire()` tous les dixièmes de seconde.
- La méthode `charAt()` retourne le caractère de la phrase à l'emplacement spécifié.
- `message.length` renvoie le nombre de caractères de `message`.
- `message.substr(0,cour)` extrait un certain nombre de caractères, précisé dans la variable `cour`, à partir du début de la chaîne `message`.
- La méthode `clearInterval()` arrête le `setInterval()` quand le dernier caractère des citations est atteint.
- La balise de citation `<q>` ajoute les guillemets en début et fin de phrase.

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <style>
      .ecrire {
        color:#fff;
        background-color:#000;
        padding:3px;
        font-weight:bold;
      }
    </style>
    <script>
      function machineEcrire() {
        document.getElementById('citation').innerHTML = message.substr(0, cour) + "<span
class='ecrire'>" + message.charAt(cour) + "</span>";
        if (cour == message.length)
          clearInterval(animation);
        else
          cour++;
      }
      const message = "Dans la vie, il faut être gentil avec les femmes ; même avec la
sienne. Quand on mettra les cons sur orbite, t'as pas fini de tourner. Un intellectuel assis
va moins loin qu'un con qui marche.";
      let cour = 0;
      const animation = setInterval("machineEcrire()", 100);
    </script>
  </head>
  <body>
    <q id="citation"></q>
  </body>
</html>

```

Ce code affichera progressivement ces belles citations à l'écran :

