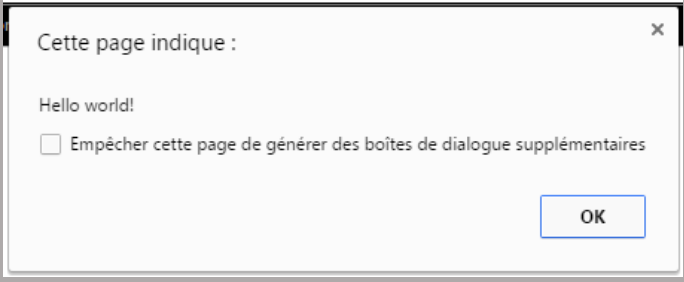
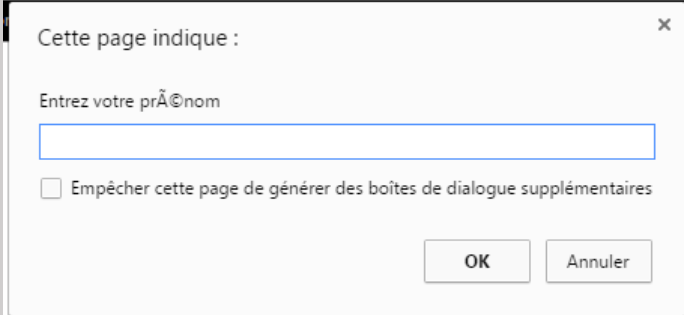


JAVASCRIPT

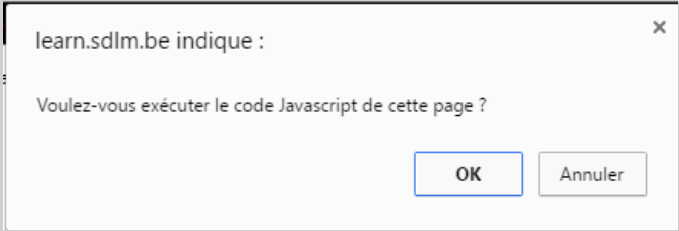
Fonction dans du code HTML

<code><script> </script></code>	Le code JavaScript se fait à l'intérieur
<code><script src="hello.js"></script></code>	Il est préférable d'appeler un fichier javascript avec le code à l'intérieur
<code>Alert('Hello world!')</code>	Affiche une alerte Hello world! 
<code>console.log('coucou');</code>	Affiche coucou dans la console du navigateur

Variables :

<code>var myVariable;</code>	Crée une variable (n'importe quel type)
<code>var myVariable = 2;</code>	
<code>alert(typeof number);</code>	Affiche le type de la variable undefined si la variable n'existe pas
<code>var hi = 'Bonjour', name = 'toi', result;</code> <code>result = hi + name;</code>	Variable hi , variable name , variable result Result sera égale à Bonjourtoi
<code>Var username = prompt("entrez un prénom")</code>	Demande à entrer un prénom et stocke dans la variable username 
<code>parseInt(text)</code>	Convertit en int
<code>entier.toString()</code>	Convertit en String
<code>myString.length</code>	Renvoie le nombre de caractères
<code>myString.toUpperCase()</code>	Met en majuscule

Conditions:

<pre>if (condition) { } else if (condition) { } else { }</pre>	<p>If</p> <p>Else if</p> <p>Else</p>
<pre>if (confirm('Voulez-vous exécuter le code JavaScript de cette page ?')) { }</pre>	<p>Confirmation (clic sur ok que ce soit TRUE)</p> 
<pre>switch (drawer) { case 1: CODE break; case 2: CODE break; case 3: CODE break; case 4: CODE break; default: CODE }</pre>	<p>Switch</p> <pre>if (drawer == 1){ } else if (drawer == 2){ } else if (drawer == 3){ } else if (drawer == 4){ } Else{ }</pre>
<pre>&& </pre>	<p>ET</p> <p>OU</p>

Boucle:

<pre>while (condition) { } Do{ }while(condition) for (var i = 0; i < 5; i++) { }</pre>	TANT QUE
	POUR (break pour arrêter la boucle avant la fin)

Liste:

<pre>var maListe = ['Sébastien', 'Laurence', 'Ludovic', 'Pauline', 'Guillaume'];</pre>	Cree une liste maListe
<pre>maListe[1]</pre>	Renvoi la 2eme valeur de maListe
<pre>maListe[1] = "coucou"</pre>	maListe [1] sera maintenant égale a coucou
<pre>maListe.push('Ludovic');</pre>	Ajoute Ludovic à la fin de la liste
<pre>maListe.unshift('Ludovic');</pre>	Ajoute Ludovic au début de la liste
<pre>maListe.shift();</pre>	Retire le premier élément de la liste
<pre>maListe.pop();</pre>	Retire le dernier élément de la liste
<pre>var cousinsString = 'Pauline Guillaume Clarisse'; var cousinsArray = cousinsString.split(' '); cousinsArray.join('-')</pre>	String Met dans une liste en fonction des espaces du string Remplace le tableau par une chaine de caractère avec des - entre les mots
<pre>for (var id in liste) { alert(liste[id]); }</pre>	Parcourt une liste (attention id = l'index)

Dictionnaire:

<pre>var myObject = { "item1" : 'Texte 1', "item2" : 'Texte 2' }</pre>	Cree un dictionnaire
<pre>myObject["item1"]</pre>	Affiche la valeur d' item1 (ici Texte 1)
<pre>myObject ['item3'] = 'Texte 3';</pre>	Ajoute un item
<pre>for (var id in myObject) { alert(myObject[id]); }</pre>	Parcourt un dictionnaire (attention id = l'item)

Fonctions:

<pre>function showMsg(arguments) { return "coucou" }</pre>	Ecrit une fonction Retourne la valeur demandé
<pre>showMsg();</pre>	Appelle la fonction

Manipulation du code html:

<pre>var link = document.getElementById('myLink'); var href = link.getAttribute('href'); link.setAttribute('href', 'http://www.siteduzero.com');</pre>	<p>Récupère l'id myLink Récupère le href de l'id inséré dans link</p> <p>Modifie le lien href du code html</p>
<pre>document.getElementById('myColoredDiv').className = 'blue';</pre>	<p>Appelle la fonction css .blue pour les éléments avec l'id myColoredDiv</p>
<pre>var div = document.getElementById('myDiv'); alert(div.innerHTML);</pre>	<p>Récupère l'id myDiv</p> <p>Renvoie tout le code de div en + des balises quand il y en a</p>
<pre>document.getElementById('myDiv').innerHTML = '<blockquote>Je mets une citation à la place du paragraphe</blockquote>';</pre>	<p>Modifie le contenu (en + des balises) du code a l'intérieur de la fonction avec l'id myDiv</p>
<pre>var div = document.getElementById('myDiv'); txt = div.textContent div.innerText '';</pre>	<p>Txt prend le texte SEULEMENT de div</p>
<pre><blockquote> <p id="myP">Ceci est un paragraphe !</p> </blockquote> <script> var paragraph = document.getElementById('myP'); var blockquote = paragraph.parentNode; </script></pre>	<p>Accède au parent du myP (la balise à la quel il est intérieur)</p> <p>Accède à l'élément avec l'id myP "p" Accède au l'élément parent de myP "blockquote"</p>
<pre>var nomParagraphe = paragraph.nodeName</pre>	<p>Récupère le nom de la balise (ici p)</p>

Evènements:

Nom evenment	Action pour le déclencher
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
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.)
input	Taper un caractère dans un champ de texte (son support n'est pas complet sur tous les navigateurs)
select	Sélectionner le contenu d'un champ de texte (input, textarea, etc.)
submit	Envoyer le formulaire
reset	Réinitialiser le formulaire

Cliquez-moi !	Affiche une alerte quand on clique sur le texte
Cliquez-moi !	This : l'objet en question (ici span) innerHTML : le contenu (ici Cliquez-moi !)
<input id="input" type="text" size="50" value="Cliquez ici !" onfocus="this.value='Appuyez maintenant sur votre touche de tabulation.';" /> Un lien bidon	Cree une zone de texte avec en valeur par default "Cliquez ici :" Une fois que l'objet est ciblé la valeur de l'objet vaut : "Appuyez maintenant sur votre touche de tabulation." Une fois que le lien est ciblé, la valeur de l'élément avec l'id "input" vaut "Vous avez maintenant le focus sur le lien, bravo !"
Cliquez-moi !	Bloque la redirection vers http://www.siteduzero.com et affiche l'alerte au clic

Fonction pour évènement :

<pre>Cliquez-moi ! <script> var element = document.getElementById('clickme'); var myFunction = function() { alert("Et de un !"); alert("Et de deux !"); }; element.addEventListener('click', myFunction); </script> element.removeEventListener('click', myFunction);</pre>	<p>Récupère la fonction avec l'id clickme</p> <p>Cree une fonction myFunction Premier évènement Second évènement</p> <p>Appelle la fonction myFunction au click de l'élément element</p>
<pre>var myFunction = function(e) { alert(e.target.innerHTML); e.target.innerHTML = "vous avez cliqué"; };</pre>	<p>e fait référence à l'objet cliqué (comme this) Affiche le texte de l'objet cliqué modifie le texte de l'objet cliqué par "vous avez cliqué"</p>
<pre>e.currentTarget.id</pre>	<p>Prend en compte le texte de l'élément sans tenir compte des autres balises à l'intérieure</p>
<pre><div id="position">coucou</div> <script> var position = document.getElementById('position'); document.addEventListener('mousemove', function(e) { position.innerHTML = 'Position X : ' + e.clientX + 'px
Position Y : ' + e.clientY + 'px'; }); </script></pre>	<p>Récupère l'élément avec l'id position</p> <p>Dès que l'on bouge la souris dans la page (document)</p> <p>L'élément div aura pour texte les coordonnées X et Y de la souris en pixel</p>
<pre>e.relatedTarget.id</pre>	<p>Avec mouseout : fourni l'objet de l'élément sur lequel le curseur vient d'entrer Avec mouseover : fournira l'objet de l'élément dont le curseur vient de sortir</p>
<pre>press = document.getElementById('press'), document.addEventListener('keypress', function(e) { press.innerHTML = e.keyCode; }); String.fromCharCode(codeTouche); e.returnValue = false;</pre>	<p>Récupère l'élément avec l'id press</p> <p>Si appuie sur touche du clavier (n'importe ou sur la page) L'élément de l'id press aura pour texte le code décimal de la touche</p> <p>Convertit le code d'une lettre en lettre</p> <p>Bloque l'évènement par default d'une balise html</p>

Fonction utile :

<pre>var myDiv = document.getElementById('myDiv'); var fOver = function(e) { var relatedTarget = e.relatedTarget; while (relatedTarget != myDiv && relatedTarget.nodeName != 'BODY' && relatedTarget != document) { relatedTarget = relatedTarget.parentNode; } if (relatedTarget != myDiv) { results.innerHTML += "Le curseur vient d'entrer.
"; } }; var fOut = function(e) { var relatedTarget = e.relatedTarget; while (relatedTarget != myDiv && relatedTarget.nodeName != 'BODY' && relatedTarget != document) { relatedTarget = relatedTarget.parentNode; } if (relatedTarget != myDiv) { results.innerHTML += "Le curseur vient de sortir.
"; } }; myDiv.addEventListener('mouseover', fOver); myDiv.addEventListener('mouseout', fOut);</pre>	<p>Reprends l'élément avec l'id myDiv</p> <p>Fonction quand on entre dans l'élément a l'id myDiv</p> <p>Code évènement</p> <p>Fonction quand on sort de l'élément myDiv</p> <p>Code évènement</p> <p>Appel la fonction quand on entre Appel la fonction quand on sort</p>
---	---

Manipuler le CSS:

<pre>var text = document.getElementById('text'); var aColor = getComputedStyle(text).color; alert(color);</pre>	<p>Récupère l'élément avec l'id text</p> <p>aColor prend la valeur de color du css de cette id</p> <p>Affiche une alert en affichant la couleur</p>
<pre>text.offsetWidth text.offsetHeight text.offsetLeft text.offsetTop text.offsetParent</pre>	<p>Contient la largeur complète (width + padding + border) de l'élément.</p> <p>Contient la hauteur complète (height + padding + border) de l'élément.</p> <p>Surtout utile pour les éléments en position absolue.</p> <p>Contient la position de l'élément par rapport au bord gauche de son élément parent.</p> <p>Surtout utile pour les éléments en position absolue.</p> <p>Contient la position de l'élément par rapport au bord supérieur de son élément parent.</p> <p>Utile que pour un élément en position absolue ou relative !</p> <p>Contient l'objet de l'élément parent par rapport auquel est positionné l'élément actuel.</p> <p>S'appelle directement par l'id de la balise</p>
<pre>document.getElementById('text').id = "blue";</pre>	<p>L'élément avec l'id text, aura maintenant l'id blue</p>

JQUERY, faciliter javascript:

<code><script src="folder_name/jquery-1.12.0.js"></script></code>	Penser à inclure le fichier jquery
<code>\$('#div').css('background-color', 'blue');</code>	Pour les balises div : l'attribut background-color sera égale à blue
<code>\$('#div.classe').css('background-color', 'blue');</code>	Pour les balises div avec la class classe
<code>\$('#div#classe').css('background-color', 'blue');</code>	Pour les balises div avec l'id classe
<code>\$('#p.coucou,p.test')</code>	Sélectionne autant d'élément que l'on veut
<code>\$('.test')</code>	Sélection tous les élément avec la classe test
<code>\$("#p:first")</code>	Sélectionne le premier <p> du document
<code>\$("#p:last")</code>	Sélectionne le dernier <p> du document
<code>\$("#p:eq(1)")</code>	Sélectionne le deuxième <p> du document
<code>\$("#p:gt(1)")</code>	Sélectionne tous les <p> après le deuxième <p> du document
<code>\$("#p:lt(1)")</code>	Sélectionne tous les <p> avant le deuxième <p> du document
<code>\$("#p:not('.legume')")</code>	Sectionne tous les <p> qui n'ont pas la classe legume
<code>\$("#p:hidden")</code>	sélectionne les éléments cachés
<code>\$("#p:visible")</code>	sélectionne les éléments visibles
<code>\$("#p:contains("texte")")</code>	éléments qui contiennent le texte spécifié
<code>\$("#p:has("élément")")</code>	éléments qui contiennent l'élément spécifié
<code>\$("#p [attribut] ")</code>	éléments qui ont l'attribut spécifié, exemple : <code>\$("#[align]")</code>
<code>\$("#p [attribut="valeur"]")</code>	éléments qui ont l'attribut et la valeur spécifiés, par exemple <code>\$("#[align=center]")</code>
<code>\$("#p [attribut!="valeur"]")</code>	éléments qui n'ont pas l'attribut et la valeur spécifiés, par exemple <code>\$("#[align!=center]")</code>
<code>\$("#type=:input")</code>	sélectionne les éléments input
<code>\$("#type=:password")</code>	sélectionne les zones de mot de passe
<code>\$("#type=:text")</code>	sélectionne les zones de texte
<code>\$("#type=:checkbox")</code>	sélectionne les cases à cocher
<code>\$("#type=:radio")</code>	sélectionne les boutons d'option
<code>\$("#type=:submit")</code>	sélectionne les boutons de validation

`$('#div').????`

Effets :

.show(tempMilliseconde) : affiche le(s) élément(s) sélectionné(s). **(défilement vers le coins haut-gauche quand on met un temps)**
.hide(tempMilliseconde) : masque le(s) élément(s) sélectionné(s). **(défilement vers le coins haut-gauche quand on met un temps)**
.toggle(tempMilliseconde) : affiche ou masque le(s) élément(s) sélectionné(s) en fonction de son/leur état actuel. **(défilement vers le coins haut-gauche quand on met un temps)**
.fadeIn(tempMilliseconde) : fait apparaître en fondu le(s) élément(s) sélectionné(s).
.fadeOut(tempMilliseconde) : fait disparaître en fondu le(s) élément(s) sélectionné(s).
.fadeTo(tempMilliseconde,opacité) : amène progressivement le(s) élément(s) sélectionné(s) à une opacité donnée.
.slideUp(tempMilliseconde) : masque le(s) élément(s) sélectionné(s) avec un défilement vers le haut.
.slideDown(tempMilliseconde) : masque le(s) élément(s) sélectionné(s) avec un défilement vers le bas.
.slideToggle(tempMilliseconde) : affiche ou masque le(s) élément(s) sélectionné(s) avec un défilement en fonction de son/leur état actuel.

Manipulation du contenu :

.html() : remplace le contenu HTML de la page
.text() : remplace le texte de la page
.replaceWith() : remplace le(s) élément(s) sélectionné(s) entièrement, pas juste le texte ou le HTML à l'intérieur
.remove() : supprime les éléments de la page
.before() : insère du contenu avant le(s) élément(s) sélectionné(s)
.after() : insère du contenu après le(s) élément(s) sélectionné(s)
.prepend() : insère du contenu à l'intérieur du/des élément(s) sélectionné(s) (après la balise HTML ouvrante)
.append() : insère du contenu à l'intérieur du/des élément(s) sélectionné(s) (avant la balise HTML fermante)
.attr() : définit un attribut et sa valeur ou simplement récupère sa valeur
.removeAttr() : supprime un attribut, **RIP**
.addClass() : ajoute une nouvelle classe à/aux élément(s) sélectionné(s) (sans remplacer sa classe actuelle)
.removeClass() : supprime une classe du/des élément(s) sélectionné(s)
.css('propriété', 'valeur') : Récupère ou définit les propriétés CSS d'un élément, même plusieurs propriétés à la fois.

.height() : hauteur d'un cadre sans les marges intérieures, extérieurs ni les bords
.width() : largeur d'un cadre sans les marges intérieures, extérieurs ni les bords

.offset() : définit les coordonnées d'un élément relativement au coin en haut à gauche de l'objet document
.position() : définit les coordonnées d'un élément relativement à son parent direct, ce qui peut être utile pour positionner des éléments au sein du même élément DOM.

Évènements:

```
$('#p').on( 'click', function () {  
    alert("Quelqu'un a cliqué sur un paragraphe!")  
});
```

Quand on clic sur un paragraphe **<p>**

.click()	au click
.scroll()	
.hover()	
.mouseover()	à l'entrée de la souris
.mouseout()	a la sortie de la souris
.mouseenter()	a entrée de la souris
.mouseleave()	a la sortie de la souris
.keydown()	appuie touche (document)
.keyup()	relachement touche (document)
.keypress()	appuie touche (document)
.focus()	ciblage(lien/formulaire/bouton)
.blur()	deciblage(lien/formulaire/bouton)
.resize()	