

# Web partie 4

Merci de vous installer  
par binôme de niveau  
différent

# SK

- Aussi appelé algorithmique. Mais qu'est ce que c'est?
  - Un algorithme est un code qui transforme un élément **a** en élément **b**.
  - Exemple de l'addition:  $a_1$  et  $a_2$  renvoie  $b$  le résultat de l'addition de  $a_1$  et  $a_2$
  - Exemple plus recherché et utile du mot de passe avec 3xMach1na&eg

- Insertion du javascript comme avec le CSS:
  - soit à l'intérieur d'une balise spécifique:

```
<script type="text/javascript">  
// mon code javascript  
</script>
```

- soit dans un fichier externe:

```
<script type="text/javascript" src="fichier.js"> </  
script>
```

# Exercice 16: insertion JS

- Insérez le code javascript suivant à l'intérieur de votre page:

```
document.write("Hello world");
```

- Résultat attendu: votre page doit afficher dans son contenu « Hello world »

- Variable: élément dont la valeur peut varier en fonction du contexte. Ce « contexte » peut être:
  - des paramètres propre à mon site
  - des informations entrées par l'utilisateur
  - le résultat d'une opération
  - etc..
- Se définit de cette manière:
  - `var ma_variable = "valeur_de_ma_variable";`
  - exemple:  

```
var langueDuSite = "fr";  
document.write("Mon site est en langue"+  
langueDuSite);
```

  
affichera: *Mon site est en langue fr*

# Exercice 17: les variables en JS

- Modifier le code précédent en remplaçant « world » par une variable *nom* qui aura pour valeur votre prénom
- Référez-vous à l'exemple de la page précédente
- Résultat attendu: dans mon cas, ma page doit afficher « Hello Thibault »
- Problème: notre code n'est pas dynamique, les valeurs des variables sont écrites directement dans le code

- La fonction *prompt("Message")* crée une pop-up qui affiche
  - le *Message*
  - un champ d'insertion de texte pour l'utilisateur
  - Un bouton « Annuler »
  - Un bouton « OK »
- Il est possible de stocker le texte entré par l'utilisateur dans une variable
  - Si l'utilisateur entre une valeur, la variable vaudra la valeur entrée
  - Si l'utilisateur clique sur « Annuler », cette variable vaudra « null »
  - Si l'utilisateur clique sur « OK » sans entrer de texte, cette variable vaudra ' '

# Exercice 17-1 : prompt()

- Insérez une fonction `prompt` qui affiche un message de bienvenue pour le visiteur et lui demande en même temps son nom
  - Stockez la valeur insérée dans une variable
  - Affichez la valeur insérée dans une variable sur la page sous la forme « Bonjour *nom*, »
- 
- Problème: si l'utilisateur clique sur « Annuler » ou « OK » sans entrer de valeur, notre page affichera « null » dans le premier cas, rien dans le deuxième..



➤ Permet de rendre son site dynamique. Le site se comportera différemment selon les valeurs prises par le contexte (les variables).

➤ La structure si... alors... sinon...

```
if(condition){  
  // code qui s'exécute si la condition est réunie  
}  
else if(autre condition){  
  // code qui s'exécute si l'autre condition est réunie  
}  
else{  
  // code qui s'exécute si aucune condition n'est réunie  
}
```

## ➤ Exemple

```
var mon_chiffre = "4";  
if(mon_chiffre < 2){  
    document.write("mon chiffre est plus petit que 2");  
}  
else if(mon_chiffre > 2){  
    document.write("mon chiffre est plus grand que 2");  
}  
else if(mon_chiffre == 2){  
    document.write("mon chiffre vaut 2");  
}  
else{  
    document.write("vous n'avez pas entré un chiffre");  
}
```

# Exercice 17-2 : if

- Remplacez la fonction prompt par une énigme
- Utilisez une structures conditionnelle pour:
  - Afficher un message de félicitations si la réponse est bonne
  - Afficher un message d'erreur dans les autres cas
  
- Problème: il serait plus pertinent d'afficher à nouveau l'énigme tant que la réponse est mauvaise

- Permet de boucler tant que la condition est valide
- La structure while

```
while(condition){  
  // code qui s'exécute tant que la condition est valide  
}
```

## ➤ Exemple

```
enigme = prompt("Combien font 4 + 3 »);  
  
while (enigme != "7") {  
  enigme = prompt("Combien font 4 + 3");  
}
```

*Note: « != » test si la valeur est « différent de »*

# Exercice 17-3 : while

- Affichez à nouveau l'énigme tant que la réponse est mauvaise

SK

# Exercice 18 : prompt(), if, while

- Refaite la fonction prompt qui affiche un message de bienvenue pour le visiteur et lui demande en même temps son nom. Comme auparavant:
  - Stockez la valeur insérée dans une variable
  - Affichez la valeur insérée dans une variable sur la page
- Et maintenant:
  - Si l'utilisateur clique sur « Annuler », nommez-le « illustre inconnu(e) »
  - Si l'utilisateur clique sur « OK » sans entrer de valeur, affichez à nouveau le prompt

# Si vous voulez aller plus loin

- Affichez le message inséré par l'utilisateur dans le champ de formulaire « Nom : »
  - Indice: il faut utiliser `document.forms...`
- Lorsque l'utilisateur change la valeur du champ de formulaire « Nom : », réécrivez en temps réel cette valeur dans le « Bonjour *nom*, »
  - Indice: il faut utiliser les `event`, la fonction `getElementById` et la propriété `innerHTML`



La librairie JQuery <https://jquery.com/>

- Documentation officielle: <http://api.jquery.com/>
- Améliorez la transition de la page index à la page contact avec un de ces plugin jquery:
  - 1: <http://tympanus.net/Development/PageTransitions/>
  - 2: <http://tympanus.net/Development/FullscreenLayoutPageTransitions/>
  - 3: <http://tympanus.net/Tutorials/FullscreenBookBlock/>
  - 4: <http://tympanus.net/Development/DraggableDualViewSlideshow/>

Regardez du côté de:

- La programmation orientée objet
- Les design patterns
- L'architecture MVC

Annexe:

- Le framework AngularJS pour des PWA - applications multi-plateforme
  - Tutoriel: <https://docs.angularjs.org/tutorial>
  - Documentation: <https://docs.angularjs.org/api>