

## Créer des applications WEB dynamiques avec le JavaScript

Savoirs	Champs ciblé
S 4.7 Langages de programmation	Web dynamique : PHP, Javascript (2)



Pré-requis aux TP ci-dessous : HTML, PHP.

On désire créer une page html dynamique qui permettra aux étudiants de vérifier leurs connaissances du code des couleurs indiquant la valeur ohmique sur les résistances.

**Exercice 1 :** Découvrir le canevas html.

1.1 Créer un répertoire « javascript » dans votre espace de travail :


✓ Exemple c:/ WAMP/www/Dupont/javascript

1.2 Copier le répertoire « code\_couleurs » depuis le disque R: sur votre espace de travail dans le répertoire nouvellement créé.

1.3 Ouvrir ce document « TP\_JavaScript.pdf ».

1.4 Ouvrir la page « code\_couleurs.html » avec le navigateur Chrome.

**- Questionnaire sur le code des couleurs -  
- Série E12 -**

<b>1</b> 	Valeur Ohmique $\Omega$ <input type="button" value="Vérifier"/>
<b>Résultat : 1 / 10</b>	<input type="button" value="Recommencer"/>

1.5 Consulter le répertoire « images » afin de voir ce qu'il contient.

1.6 Ouvrir le fichier « code\_couleurs.html » avec notepad++.

1.7 Analyser le code HTML et retrouver son action sur la page du navigateur.

1.8 Quelle est la valeur ohmique de la résistance affichée ?

1.9 Modifier le code html afin de faire apparaître à la place une résistance de 4,7 k $\Omega$ .

On peut remarquer qu'il n'est pas possible d'avoir une page html avec des valeurs de résistances qui soient proposées au hasard dans une série normalisée par exemple.

Il faut donc utiliser un langage de programmation permettant de rendre la page dynamique, cela est possible avec le PHP ou le JavaScript par exemple.

**Exercice 2 :** Afficher aléatoirement une résistance dans la série E12.

Rappel : la série E12 contient 12 valeurs normalisées auquel on ajoute un multiplicateur :

10 12 15 18 22 27 33 39 47 56 68 82

2.1 Pour cela il faut des variables afin d'effectuer des calculs et afficher les bonnes images.

Copier le code repéré **2.1** donné dans le document « Sources a Copier.html » tout en haut de votre fichier « code\_couleurs.html ».

La première partie contient les variables globales (remarquer le mot clé var).

Le seconde partie contient une fonction JavaScript « NOUVELLE\_RESISTANCE() » qui permet d'afficher une résistance au hasard dans la série E12.

Lire les lignes et les commentaires.

2.2 Modifier le code html du fichier « code\_couleurs.html » en remplaçant la partie ou sont affichées les images constituant la résistance.

```
<td align="center" width="350">
<script type="text/javascript">
    NOUVELLE_RESISTANCE() ;
</script>
</td>
```

2.3 Afin de débbugger le code JavaScript appuyer sur la touche F12 dans le navigateur Chrome afin de faire apparaître la console de développeur.

Réactualiser plusieurs fois la page, que remarquez-vous ?

**Exercice 3 :** Vérifier la réponse, version 1.

3.1 Afin de vérifier si la réponse saisie dans le <input> est correcte il faut appeler une fonction JavaScript à partir du bouton « Vérifier ». Modifier le code html du fichier « code\_couleurs.html » en ajoutant l'attribut « Event onclick » au bouton.

```
<td align="center" width="100">
<input type="submit" value="Vérifier" width="50" onclick="VERIFICATION()" />
</td>
```

Lorsqu'on effectuera un click sur ce bouton la fonction « VERIFICATION() » sera appelée.

3.2 Ajouter la fonction « VERIFICATION() » à partir du document « Sources a Copier.html » repéré **3.2** avant la fin du script.

3.3 Effectuer plusieurs tests en actualisant la page. On peut remarquer plusieurs choses :

Le résultat est affiché sur la même page mais tout a été effacé, ce qui n'est pas satisfaisant.

Si l'on saisit la valeur d'une résistance (par exemple de 4,7 kΩ) en entrant 4k7 ou 4.7k ou 4,7k la réponse est fausse.

**Exercice 4 :** Vérifier la réponse, version 2.

4.1 Ajout d'une fonction permettant de gérer les milli (m), les kilo (k ou K), les Mega (M) et la virgule.

Mettez la ligne PROPOSITION = ..... en commentaires et ajouter la ligne en dessous :

```
//PROPOSITION = document.getElementById("REPONSE").value;
PROPOSITION = DECODAGE_REPONSE() ;
```

4.2 Ajouter la fonction « DECODAGE\_REPONSE() » disponible dans le document « Sources a Copier.html » repérée 4.2 avant la fin du script.

4.3 Effectuer différents essais. Puis analyser le code en prenant un jeu d'essai (par exemple 4k7 puis 4.7K).

**Exercice 5 :** Ajout dynamique d'éléments au tableau html (résistance et zone de saisie).

On désire maintenant ajouter des résistances au fur et à mesure que les réponses sont données. Il faut donc utiliser des méthodes (fonctions JavaScript) qui interagissent sur les éléments html. À cette fin des id (identificateurs) ont été prévus dans le code HTML pour identifier les différents éléments.

5.1 Ajouter des variables locales à la suite de celles déjà en place à partir le document « Sources a Copier.html » repéré 5.1.

5.2 Remplacer la fonction « NOUVELLE\_RESISTANCE() » par la nouvelle à partir le document « Sources a Copier.html » repéré 5.2.

5.3 Remplacer toute la partie html par le code html donné dans le document « Sources a Copier.html » repéré 5.3.

Remarquer que le code html est beaucoup plus court : toute la seconde partie du tableau est créée par le code JavaScript.

5.4 Effectuer des essais en actualisant la page à chaque fois.

5.5 Consulter le code de la fonction « NOUVELLE\_RESISTANCE() » ainsi que les commentaires associés.

**Exercice 6 :** Ajout dynamique d'éléments au tableau html (réponse bonne ou mauvaise).

6.1 Remplacer la fonction « VERIFICATION() » par une nouvelle, à partir du document « Sources a Copier.html » repéré 6.1.

6.2 Effectuer des tests.

On peut ici remarquer que le côté dynamique apporté par le JavaScript est bien réel : au fur et à mesure le tableau s'allonge mais cela reste la même page. Tout est fait en local par le navigateur.

6.3 Consulter le code de la fonction « VERIFICATION() » ainsi que les commentaires associés.

Remarquer que la vérification se fait soit en cliquant sur le bouton « Vérification » mais aussi si l'on appuie sur la touche « Entrée » à la fin de la saisie de la valeur ohmique : cela permet de gagner du temps.

Remarquer aussi que le curseur (focus) se place automatiquement dans la zone de saisie.

Remarquer que l'ajout des résistances se termine au bout de 10.

**Exercice 7 :** Affichage du résultat.

7.1 Ajouter les fonctions « FINI() » et « RECOMMENCER() » juste avant le fin du script à partir du document « Sources a Copier.html » repéré 7.1.

7.2 Effectuer des tests jusqu'à recommencer.

7.3 Consulter le code des fonctions « FINI() » et « RECOMMENCER() » ainsi que les commentaires associés.