



Ressourcesinformatiques

Apprenez les langages **HTML5, CSS3** et **JavaScript** pour créer votre premier site web

Denis MATARAZZO

Téléchargement
www.editions-eni.fr



Les éléments à télécharger sont disponibles à l'adresse suivante :
<http://www.editions-eni.fr>
Saisissez la référence ENI du livre **RIHTCSJA** dans la zone de recherche et validez. Cliquez sur le titre du livre puis sur le bouton de téléchargement.

Avant-propos

Chapitre 1 Le Web

1. Qu'est-ce que le Web ?	7
1.1 Côté serveur : HTTP, FTP, langages, SQL	7
1.2 Côté client : HTML, CSS, JavaScript	10
2. Les langages et leur utilité	10
2.1 HTML	10
2.2 CSS	13
2.3 JavaScript	16
2.4 Exemple général avec les trois langages	20
3. Les formats d'images	26
3.1 Format Bitmap	26
3.2 Format vectoriel	31
4. Les navigateurs et leurs outils	38

Chapitre 2 Règles générales

1. Préserver la lisibilité : l'indentation, les commentaires	47
1.1 L'indentation	47
1.2 Les commentaires	50
2. Penser au référencement	51
3. Dossiers et chemins vers les fichiers	53

2 _____ HTML5, JavaScript et CSS3

Pour créer votre premier site web

4. Les éditeurs pour le code	57
5. Des raccourcis bien pratiques	63
5.1 Sauvegarder et tester une page	63
5.2 Sélectionner du texte ou se déplacer plus vite sans la souris	65

Chapitre 3 HTML

1. Création d'une page web	67
2. Code HTML obligatoire	68
3. Le doctype	69
4. La balise <head>	69
4.1 Lien avec une feuille de style	71
4.2 Lien avec un fichier JavaScript	72
5. La balise <body>	73
5.1 Méthode et balises pour structurer une page	75
5.2 Le texte dans la page HTML	83
5.3 Les caractères spéciaux	85

Chapitre 4 CSS3

1. Les trois styles de base possibles	87
1.1 Le style de balise	87
1.2 Le style de classe	91
1.3 Le style d'ID	92
1.4 Combinaison des trois méthodes	94
2. Les polices de caractères et le Web	97
3. Les sélecteurs	107

- 4. Les pseudoclasses 117
 - 4.1 Pour les liens 117
 - 4.2 Pour le texte 119
 - 4.3 Pour les sélecteurs 120
- 5. Les couleurs en hexadécimal, en RGBA ou en HSLA 120
- 6. Les images et les bordures 125
- 7. Les boutons issus d'images ou de polices 126
- 8. Les fonds et fonds multiples 131
- 9. Le positionnement 132
- 10. Le débordement 134
- 11. Utilisation de padding et margin 135
- 12. Un préfixe par navigateur 136
- 13. Des propriétés décoratives (ombre, dégradé, arrondi...) 137
- 14. Des colonnes dans le texte 139
- 15. Les transformations 3D 141
- 16. Les transitions et animations 145
- 17. Le responsive design et les media queries 147

Chapitre 5
JavaScript

- 1. Introduction 151
- 2. La liste des tâches 151
- 3. Variables et affectation 154
- 4. Les types de variables 155
 - 4.1 Les valeurs numériques 155
 - 4.2 Le texte et la concaténation 155
 - 4.3 Les tableaux 157
 - 4.4 Les booléens 158

4 _____ HTML5, JavaScript et CSS3

Pour créer votre premier site web

4.5	Les objets	159
5.	Les opérateurs	161
6.	Les conditions	165
6.1	if, else et les accolades	165
6.2	switch case	171
7.	Itérations	174
7.1	La boucle for	174
7.2	Le while	178
7.3	Le do ... while	179
7.4	break et continue	180
8.	Déboguer un programme	181
9.	Les fonctions	184
9.1	Déclaration	185
9.2	Appel	186
9.3	Les variables locales et globales	186
9.4	Le retour d'une fonction	188
10.	Les cookies	189
11.	Le drag and drop	192
12.	Afficher le site HTML en plein écran	195
13.	Gestion de l'historique du navigateur	197
14.	La géolocalisation	199
15.	Les bases de données locales	202
15.1	Création d'une base de données	203
15.2	Création d'une table	203
15.3	Insertion d'enregistrements	204
15.4	Lecture d'informations	204
16.	Générer des PNG en JavaScript	205
17.	Ajax	210

Chapitre 6

Mise en page HTML et CSS

- 1. Les blocs et leur position à l'écran 215
 - 1.1 Les tableaux..... 215
 - 1.2 Les div et les nouvelles balises HTML5..... 218
- 2. Les listes 223

Chapitre 7

Les méthodes de dessin

- 1. La balise Canvas 227
- 2. La balise SVG 232
- 3. Avantages et inconvénients des deux technologies 241

Chapitre 8

Le multimédia

- 1. La balise vidéo..... 243
- 2. Les codecs vidéo 247
- 3. La balise audio..... 248
- 4. Les codecs audio 250

Chapitre 9

Les formulaires

- 1. Introduction 253
- 2. Fonctionnement d'un formulaire client/serveur 254
- 3. Les différentes balises du formulaire 255
- 4. Les expressions régulières..... 264
- 5. La validation du formulaire 266

6 _____ HTML5, JavaScript et CSS3

Pour créer votre premier site web

6. Ajout d'un script CGI sur le serveur 268

Chapitre 10 **Les liens et menus en HTML5**

1. Introduction 271

2. Création de liens 272

 2.1 Ouverture de page HTML 272

 2.2 Ouverture d'une image 273

 2.3 Navigation dans la page 274

 2.4 Proposer le téléchargement d'un fichier 275

 2.5 Envoyer un mail 275

 2.6 Déclenchement d'un script JavaScript 277

3. Création d'un menu (liste + liens + CSS) 277

4. Ajout de « data » dans les liens 279

Conclusion 283

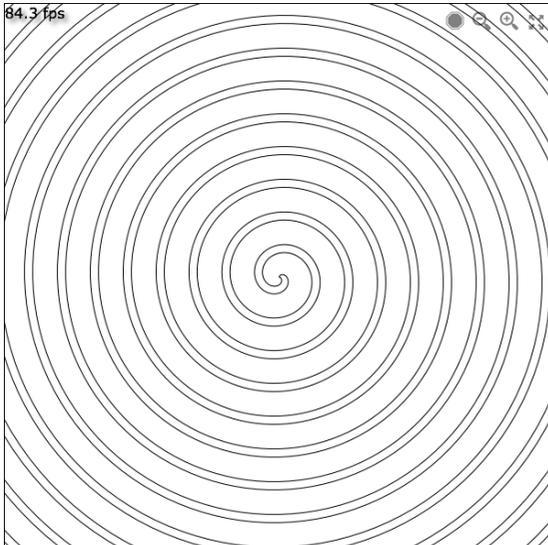
Index 285

Chapitre 7

Les méthodes de dessin

1. La balise Canvas

Le fichier **7_1_spirale.html** affiche une spirale animée. Le dessin de la spirale avait déjà été détaillé dans le chapitre Le Web, rajoutons à présent l'interaction avec JavaScript permettant de modifier la vitesse de rotation de la spirale, par exemple.



La copie d'écran montre à gauche une information sur le nombre d'images par seconde (ou *frames per second*, fps) et à droite quatre boutons. Le premier arrête l'animation, le suivant ralentit l'animation, le troisième l'accélère et le dernier permet de passer en plein écran.

Le code HTML ne contient rien de nouveau.

```
<body onload="dessine()" >
  <div id="fullscreen">
    <canvas id="spire_id" width="600" height="600"></canvas>
    <span class="fs-button"></span>
    <span class="plus_btn"></span>
    <span class="moins_btn"></span>
    <span class="stop_btn"></span>
  </div>
</body>
```

Nous avons donc un `div`, pour le plein écran, la balise `canvas` pour le dessin de la spirale, et ensuite quatre `span` permettant d'afficher les boutons.

Les styles de classe des `span` utilisent la police `ModernPictogramsNormal` pour afficher les pictogrammes.

Il y a un événement `onload` dans la balise `body` pour mettre en route l'animation en appelant la fonction `dessine()`.

Une nouvelle instruction est utilisée dans la fonction `dessine`, c'est `setInterval`.

```
intervId = setInterval(reDessine, 1);
```

Cette ligne de code permet d'appeler la fonction `reDessine` toutes les millisecondes, donc 1000 fois par seconde. La variable `intervId`, qui est une variable globale, permettra de mettre fin à ce `setInterval`. Car une fois lancé, le `setInterval` exécutera toutes les millisecondes la fonction `reDessine`, et si rien n'est prévu pour l'arrêter, il faudra quitter le navigateur pour tout stopper.

Avant de regarder ce que fait la fonction `reDessine`, voyons le code qui permet de gérer le clavier. Il est écrit en utilisant jQuery et vous aurez sûrement d'autres occasions d'écrire ce genre de code qui enrichit jQuery.

```
$(function() {
    $(document).keydown(function(evt) {
        if (evt.keyCode === 32) {
            if (tourne_bool) {
                tourne_bool = false;
                clearInterval(intervId);
            } else {
                tourne_bool = true;
                intervId = setInterval(reDessine, 1);
            }
        }
        if (evt.keyCode === 27) {
            tourne_bool = false;
            clearInterval(intervId);
        }
    });
});
```

La première ligne accède à jQuery et permet d'ajouter une nouvelle fonctionnalité. La seconde ligne, avec le `keydown`, va intercepter les touches du clavier qui sont appuyées. Lorsque l'événement a lieu, le code de la touche pressée est stocké dans la propriété de l'événement, `keyCode`.

Pour ceux qui connaissent l'ASCII (*American Standard Code for Information Interchange* qui est le code américain normalisé pour l'échange d'information, c'est-à-dire une norme de codage des caractères ; pour plus de détails, voir : <http://www.asciitable.com>), le premier test est fait sur la barre d'espace, qui a pour `keyCode` 32.

À différents endroits du code, il y a un booléen `tourne_bool` qui est lu et/ou mis à jour, indiquant si la spirale tourne à ce moment-là ou non.

Si la barre d'espace est pressée et que `tourne_bool` est à `true`, alors le booléen passe à `false`, et la fonction `clearInterval(intervId);` est appelée.

C'est ce `clearInterval(intervId);` qui va mettre un terme au `setInterval` défini précédemment.

Si la touche enfoncée a pour `keyCode` 27, c'est la touche [Echap] (esc ou escape) du clavier qui stoppera l'animation.

La partie suivante du code, avec :

```
$(document).ready(function() {  
    ...  
})
```

permet d'initialiser le `click` sur les différents boutons : le passage ou la sortie du plein écran pour le premier, puis `moinsVite()`, `plusVite()` et `stopSpir()` ; pour les trois autres boutons.

■ Remarque

À noter que le code pour le plein écran n'est écrit que pour Chrome et Safari, soit les navigateurs utilisant webkit.

L'appel par la fonction `setInterval` étant déjà le plus rapide possible puisque paramétré sur une milliseconde, la solution pour modifier la vitesse sera d'augmenter ou de diminuer le pas entre chaque morceau de dessin de la spirale.

Nous avons donc la fonction `reDessine()` qui est appelée 1000 fois par seconde. Elle va simplement définir le sens du dessin de la spirale et surtout appeler ensuite la fonction `dessineSpirale()` qui s'occupe réellement de dessiner la spirale.

`dessineSpirale()` commence avec la ligne de code :

```
canvas.width = canvas.width;
```

Cette ligne va simplement permettre d'effacer le `canvas`. C'est le cas avec beaucoup de systèmes de dessin, où un dessin est créé dans un premier temps. Ensuite, pour l'animer, l'ensemble est effacé pour être à nouveau dessiné un peu plus loin, et ainsi créer l'animation. Le fait d'écrire que la largeur est égale à la largeur va en fait très rapidement réinitialiser le `canvas`, et donc l'effacer.

Le code permettant de dessiner la spirale a été vu en dans le premier chapitre. Ici, il faut remarquer que la ligne suivante :

```
var theta = angleDepart;
```

est écrite une première fois pour la première spirale, puis, une fois dessinée, le stylo est remis au centre et un nouvel angle de départ est défini à l'opposé du premier, puisque la moitié d'un cercle ($\text{PI}/2$) est ajouté.

```
ctx.moveTo(0, 0);  
  
// on repart à 90°  
theta = angleDepart + Math.PI / 2;
```

`canvas` permet d'afficher assez simplement une ombre. C'est ce que font les lignes qui finissent le dessin. Les propriétés `shadowOffsetX` et `shadowOffsetY` définissent le décalage de l'ombre, `shadowBlur` la taille du flou et le dernier paramètre définit la couleur.

```
// ombre  
ctx.shadowOffsetX = 4;  
ctx.shadowOffsetY = 4;  
ctx.shadowBlur = 5;  
ctx.shadowColor = 'rgba(0,0,0,0.6)';
```

Une fois le tout dessiné, c'est la partie texte pour la vitesse d'affichage qui est mise en place.

Deux variables sont utilisées ici : `thisloop` et `lastloop`. `lastloop` est initialisée avec la date complète au début du programme. Sa précision est de l'ordre de la milliseconde. Lorsque le dessin de la spirale sera terminé, on stockera dans `thisloop` la date en cours, qui sera donc plus grande que `lastloop`, la différence entre les deux équivalant à la durée du dessin. Au final, la variable `thisFrameTime` contiendra la différence entre `thisloop` et `lastloop`, si bien que nous connaîtrons le temps mis pour dessiner une spirale.

```
var text = (1000 / frameTime).toFixed(1) + " fps";  
ctx.font = "12pt Verdana";  
ctx.textAlign = "left";  
ctx.textBaseline = 'top';  
ctx.fillStyle = 'rgb(0,0,0)';
```

L'affichage de la vitesse étant en images par seconde, il faut partir de la valeur 1000, puisque le temps récupéré est en millisecondes, et la diviser par `frame-time` pour avoir une valeur en relation avec les secondes.

L'instruction `.toFixed(1)` permet d'arrondir à un chiffre après la virgule le résultat de la division.

Ensuite, les propriétés du `canvas` liées au texte sont utilisées pour définir la police utilisée et sa taille, l'alignement et la couleur.

2. La balise SVG

Le format de dessin SVG permet de dessiner au travers de balises. Il peut être créé dans un éditeur de code ou exporté à partir d'un logiciel de dessin vectoriel.

Il est également possible d'utiliser des CSS pour l'aspect et des scripts pour l'interactivité. Dans l'exemple qui suit, le script est en fait de l'ECMAScript.

■ Remarque

Pour la petite histoire, ECMAScript est à l'origine du JavaScript. La technologie Flash utilise ActionScript qui découle également d'ECMAScript, ce qui rend les syntaxes très proches.

Exemple d'infobulle

On pourrait écrire un livre complet sur le format SVG. L'exemple suivant va permettre d'avoir une idée sur ce qu'il est possible de faire. Le fichier **7_2_infoBulle.svg** montre une animation et un peu d'interactivité.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE svg [
  <!ENTITY dureeAnim "2s">
]>
```

L'en-tête du fichier SVG est du XML. Ensuite, l'entité (`ENTITY`) est l'équivalent d'une variable. Ici, `dureeAnim` mémorise le texte « 2s » qui sera la durée de l'animation faite en SVG.