

# Entrainement Ninja au HTML et CSS !

## Prérequis HTML

- Comprendre les [Sélecteurs CSS](#)
- Comprendre `width`, `height`, `padding`, `border` and `margin`

## Cours

<http://fr.learnlayout.com/>

Page 1 à 6 : la plupart du temps, il est souhaitable que `width` et `height` intègrent le `padding` et `border`. Cette partie vous explique comment faire.

Pages 7 à 13 : explications sur le positionnement des éléments.

Page 14 : explication sur comment rendre un site responsive. Je recommande de sauter cette page pour le moment.

Page 16 : description de `display: inline-block` pour aligner horizontalement des éléments qui d'habitude se rangent à la verticale.

Page 17 jusqu'à la fin : Je recommande de sauter ces pages pour le moment.

Après tout cela, vous aurez compris l'essentiel de CSS et pourrez passer sereinement à la suite (JavaScript) !

## Attributs utilisés

```
/* pour définir à la main la largeur des éléments */
display: inline-block;

/* indispensable avec inline-block */
vertical-align: top;
box-sizing: border-box; /* cf. cours pour comprendre */
width: 100%;
min-height: 100px;
border: 1px #000 solid;
margin: 0; /* marge externe */
padding: 0; /* marge interne */
```

---

## EXERCICE 1 - HTML

### Test 1 :

Créer deux paragraphes d'une hauteur de 300 pixels et d'une largeur de 700 pixels. Y inclure un texte très long et gérer son débordement afin qu'il soit entièrement lisible.

### Test 2 :

Créer cinq titres de niveau 2 et les afficher sous forme de liste (voir la propriété `display`).

### Test 3 :

Dans un élément `<div>`, inclure un élément `<span>` contenant du texte et lui donner le style bloc.

### Test 4 :

Créer un menu vertical dont les éléments sont des liens `<a>`.

### Test 5 :

Créer une image contenant un paragraphe incluant du texte et deux éléments `<img />` qui se suivent. Faire flotter les images, la première à gauche et la seconde à droite.

### Test 6 :

Placer trois images de tailles initiales différentes dans une page. Écrire les styles pour qu'elles s'affichent avec la même taille. Ensuite, positionner afin d'obtenir un effet de cascade avec un décalage horizontal et vertical constant pour chaque image par rapport à la précédente. L'utilisateur doit pouvoir mettre chacune d'elles au premier plan en passant la souris dessus (voir la propriété `z-index`).

### Test 7 :

Créer une mise en page à trois colonnes de largeurs respectives 20%, 65% et 15%. La première et la troisième doivent contenir respectivement un menu et une liste de liens créés à partir d'images. La colonne centrale doit posséder un contenu éditorial.

### Test 8 :

Créer une mise en page selon le modèle de la figure ci-dessous:

La colonne de gauche (repère 1) a une largeur de 200 pixels et le bandeau (repère 2) une hauteur de 150 pixels. Le bandeau contient le titre du site, la colonne de gauche un menu de la zone principale (repère 3) du texte et des images au choix. Le premier lien du menu doit afficher une page ayant la même structure et le même contenu dans les zones 1 et 2, mais un contenu éditorial différent dans la zone 3.

1	2
	3

Figure 1:

## RESSOURCE HTML

<https://developer.mozilla.org/fr/docs/Web/HTML/Element>

## EXERCICE 2 - CSS

<http://flukeout.github.io/>

## EXERCICE 3 - GABARITS

Réaliser progressivement cette page, idéalement sans regarder le CSS.

<http://chatchan.us/gabarits.html>

Les plus téméraires peuvent directement s'attaquer à la reproduction des gabarits de ce site :

<http://www.alsacreations.com/static/gabarits/liste.html>

Attention, il ne s'agit pas de reproduire l'icone mais bien toute la page du gabarit !