

Le responsive Webdesign

ELAN

202 avenue de Colmar - 67100 STRASBOURG

☎ 03 88 30 78 30 📠 03 88 28 30 69

✉ elan@elan-formation.fr

www.elan-formation.fr

SAS ELAN au capital de 37 000 € -

RCS Strasbourg B 390758241 – SIRET 39075824100041 – Code APE : 8559A

N° déclaration DRTEFP 42670182967 - Cet enregistrement ne vaut pas agrément de l'Etat

SOMMAIRE

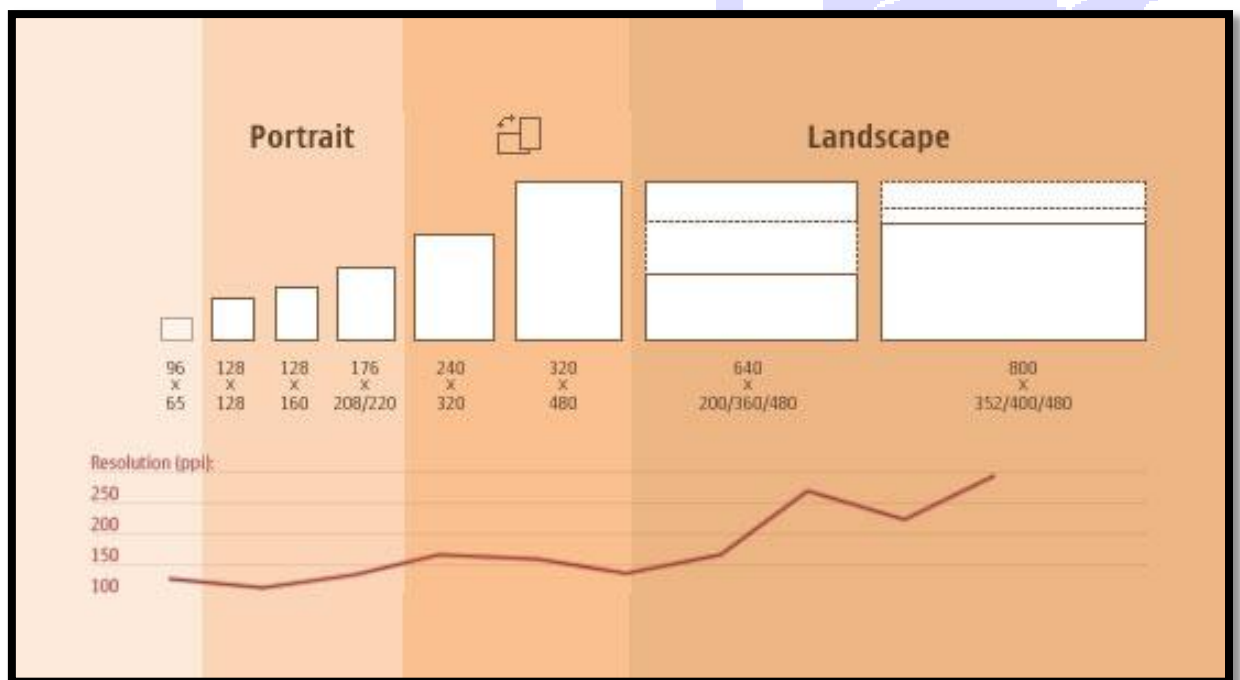
I.	Définition.....	3
II.	Mise en place.....	4
1.	Un site et un contenu qui s'adaptent	4
2.	La mise en page.....	4
3.	Des images qui s'étirent, se redimensionnent ou disparaissent.....	6
4.	La typo.....	6
5.	La couche Javascript	7
6.	Le poids	7
7.	Curseur VS tablette tactile	7
III.	Les limites du responsive design	7
1.	C'est beaucoup (beaucoup) plus long qu'avant.....	7
2.	Plus de connaissances et plus de technique.....	7
3.	Un coût plus élevé.....	8
4.	Moins de créativité.....	8
5.	Le cas des images	8
6.	Le temps de chargement	8
7.	Plus de code, plus de mises à jour	8
8.	Plus de différences entre les navigateurs et supports	8
9.	Il est souvent plus rapide de créer une version mobile qu'une version responsive.....	8

I. Définition

Si vous vous intéressez de près ou de loin au webdesign ou à l'intégration, vous n'avez pas pu échapper au dernier terme à la mode : le "responsive web design". Cette avancée technologique est également une nouvelle philosophie de création de site.

Le "responsive web design" est né suite à un besoin grandissant. De nos jours, il n'y a pas un client qui demande un site Internet sans demander une version mobile de celui-ci. Seulement en quelques années le nombre d'appareils et de résolutions servant à consulter des sites web a tout simplement explosé : ordinateurs, smartphones, tablettes, web TV, grille-pain, format portrait, paysage, etc....

Entre 2005 et 2008 on a identifié pas moins de 400 résolutions d'écran différentes pour tous les appareils vendus.



Le problème est qu'on ne peut pas – ou du moins ce ne serait pas très pratique – développer autant de versions d'un site qu'il y a de résolutions différentes.

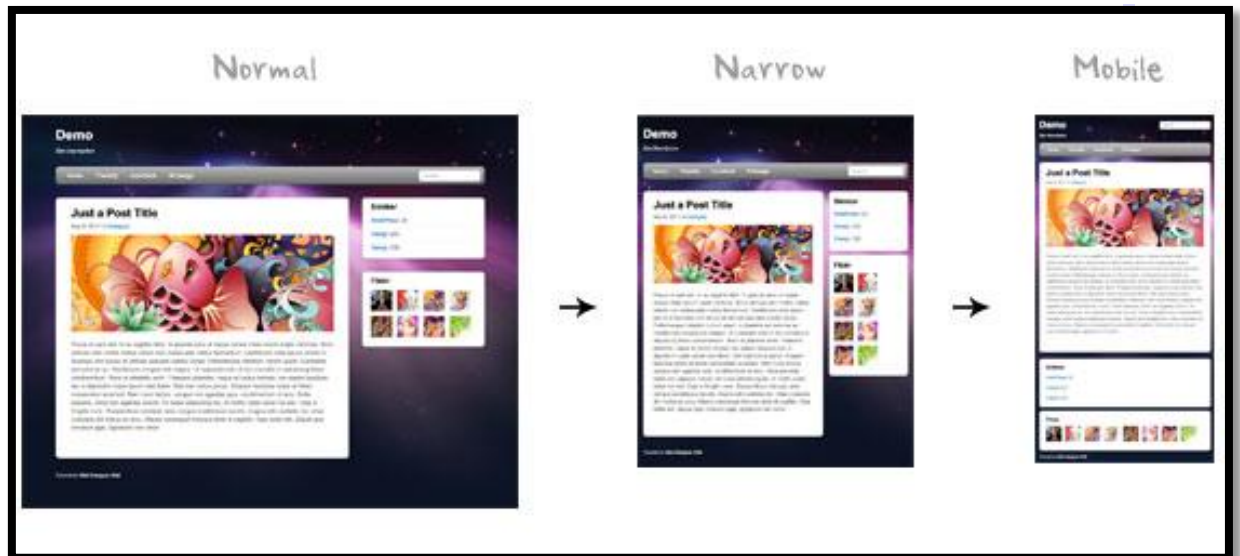
La solution à ce problème grandissant s'est donc imposée d'elle-même : le responsive design.

Cette évolution des techniques de webdesign, d'intégration et de développement se base sur un objectif simple : un même site doit pouvoir s'adapter lui-même aux différentes résolutions d'écrans... et donc son contenu aussi ! Les colonnes par exemples peuvent s'ajuster, se déplacer, voire disparaître. Les images se redimensionnent, se replacent, etc....

Lorsque l'on dit que le responsive design est également une philosophie, c'est parce que cela ne s'arrête pas à quelques modifications pour que le tout s'adapte aisément mais il faut repenser toute l'approche de création d'un site Internet. En effet, on ne navigue pas

sur un smartphone comme sur un ordinateur. Le but même de la visite d'un site peut différer selon que l'utilisateur soit sur un support ou un autre.

Voici un exemple pour illustrer des webdesigns créés de façon "responsive" :



II. Mise en place

1. Un site et un contenu qui s'adaptent

Avec autant de résolutions, le site doit devenir flexible. Cela concerne :

- la mise en page ou "lay-out" qui repose généralement sur des grilles de mise en page
- les images et vidéos
- les typographies
- la couche JavaScript
- le poids

Avant toute chose, cette nouvelle philosophie veut que le site soit pensé et prévu pour différents supports. Donc il est possible que l'objectif et le contenu du site ne soient pas les mêmes d'un terminal à l'autre. La navigation sur mobile est un parfait exemple : nous avons plus tendance à aller sur un site pour chercher une information précise et nous lisons en survolant. Cela contraste avec une utilisation sur ordinateur : contempler des images ou naviguer sur toutes les pages d'un site, lire des articles de fond, ou encore vivre une expérience interactive.

Cela va de soi qu'un contenu correctement hiérarchisé selon le type de support et sa résolution est donc indispensable. Celui-ci est ensuite découpé et balisé en HTML en tenant compte de chaque cas, préparant ainsi le futur site aux étapes suivantes.

2. La mise en page

Pour la mise en page, les systèmes de grilles comme **960.gs** ou encore **Blueprint** et **Bootstrap** plus récemment ont inspiré le responsive design. C'est donc principalement à base de CSS que le design est rendu fluide, adaptable. Depuis peu, le responsive design a connu un essor important grâce à l'arrivée des media-queries CSS3.

Le CSS3, évolution du CSS2, apporte son lot de nouveautés assez génialissimes, dont les fameux media queries. Ceux-ci permettent d'appliquer des styles avec beaucoup plus de souplesse qu'auparavant. On peut ainsi cibler des résolutions précises et y attribuer un style particulier.

Par exemple si la résolution est supérieure ou égale à 1200px, alors on va décider de placer nos 3 blocs sur 3 colonnes. On peut ensuite décider avec quelques lignes de CSS3 que si la résolution passe entre 800 et 1199px de large, on va placer les 3 blocs sur 2 colonnes et si la largeur devient inférieure à 800px, alors on empêche tout flottement et on positionne les 3 blocs l'un en dessous de l'autre. Tout comme si on le souhaite, on peut décider de cacher le 3è bloc si son contenu est secondaire (avec un display:none ou un visibility:hidden selon l'effet voulu).

Pour les adeptes de CSS, voici un bout de code très pratique à garder sous la main pour vos prochains designs ! Cela vous permettra aussi, si vous n'êtes pas encore familier avec les media queries, de comprendre comment se passe le ciblage d'une résolution ou d'une autre.

```
/* Smartphones (portrait and landscape) ----- */
@media only screen
and (min-device-width : 320px)
and (max-device-width : 480px) {
/* Styles */
}

/* Smartphones (landscape) ----- */
@media only screen
and (min-width : 321px) {
/* Styles */
}

/* Smartphones (portrait) ----- */
@media only screen
and (max-width : 320px) {
/* Styles */
}

/* iPads (portrait and landscape) ----- */
@media only screen
and (min-device-width : 768px)
and (max-device-width : 1024px) {
/* Styles */
}

/* iPads (landscape) ----- */
@media only screen
and (min-device-width : 768px)
and (max-device-width : 1024px)
```

```

and (orientation : landscape) {
/* Styles */
}

/* iPads (portrait) ----- */
@media only screen
and (min-device-width : 768px)
and (max-device-width : 1024px)
and (orientation : portrait) {
/* Styles */
}

/* Desktops and laptops ----- */
@media only screen
and (min-width : 1224px) {
/* Styles */
}

/* Large screens ----- */
@media only screen
and (min-width : 1824px) {
/* Styles */
}

/* iPhone 4 ----- */
@media
only screen and (-webkit-min-device-pixel-ratio : 1.5),
only screen and (min-device-pixel-ratio : 1.5) {
/* Styles */
}

```

3. Des images qui s'étirent, se redimensionnent ou disparaissent

Il y a de multiples choix lorsqu'on a des images dans son contenu et que celui-ci doit s'adapter à différentes résolutions : on peut rétrécir les images, on peut les "croper" (c'est-à-dire n'afficher qu'une partie de l'image) mais on peut aussi les déplacer ou encore les cacher.

Attention : Pensez au facteur "poids" pour les mobiles. Ce n'est pas pour rien si tous les sites efficaces pour mobiles sont très allégés. Il faut qu'ils se chargent vite et que seule l'information importante soit mise en avant. Des choix s'imposeront sûrement à vous...

4. La typo

La lisibilité des textes est un facteur extrêmement important à prendre en compte lors de la confection de vos sites. Cela est encore plus vrai lorsque vos typographies doivent s'afficher sur différents supports. Il est conseillé de privilégier les unités % et em pour assigner des tailles à vos typographies afin que celles-ci restent proportionnelles au ratio conseillé sur le support (un texte à 12px sera lisible sur la majorité des résolutions et des supports "classiques" mais pas forcément sur les appareils mobiles ou encore sur de très grandes résolutions comme du 2550x1440px).

5. La couche Javascript

Ce point est très délicat et il vous faudra décider bien en amont des fonctionnalités Javascript indispensables de celles qu'on peut élaguer dans le cas d'une version mobile.

Par exemple si vous envisagez de mettre en place une navigation originale à l'aide de Javascript, il faudra que votre code soit suffisamment flexible pour s'adapter à différentes résolutions sans que l'utilisateur n'ait à intervenir. De plus, il y a de fortes chances que cela ne marche pas (ou pas correctement) sur un smartphone selon l'effet désiré car on ne navigue pas de la même manière (absence de souris, absence de clavier pour la navigation, etc....), dans d'autres cas peut-être faudra-t-il utiliser d'autres scripts comme jQuery mobile ? Ou tout bonnement laisser tomber ces fonctionnalités qui n'apportent pas grand chose au contenu primordial.

6. Le poids

Les différents terminaux qui vont se connecter à votre site peuvent être soumis à des débits très différents et notamment à des débits très bas (smartphones, tablettes et autres supports mobiles). Il faut donc traiter la question du temps de chargement avec beaucoup de minutie.

Pensez à supprimer l'appel des scripts et des médias dont vous ne vous servirez pas, cela vous fera toujours gagner quelques requêtes et un peu de bande passante. Plutôt que de redimensionner à l'échelle 1/10 vos images, pensez à en créer des alternatives. Vous pourrez gagner jusqu'à environ 70% de poids sur les images de cette manière.

7. Curseur VS tablette tactile

Prenez toujours en compte les différences d'utilisation de votre site sur les différents supports. Une des grandes différences ergonomiques sera d'adapter votre site pour écran tactile.

III. Les limites du responsive design

On entend beaucoup parler du responsive design comme de quelque chose de vraiment génial qui fait qu'à peine tu prononces ce mot, tu es le geek le plus cool de la pièce.

Mais peu de gens parlent du côté obscur du responsive design et pourquoi cela n'a pas que du bon !

1. C'est beaucoup (beaucoup) plus long qu'avant

Et oui, fini le temps où on réalisait un site à 980px de large et fini, basta c'est en ligne ! Forcément il faut traiter de beaucoup plus de cas de figure autant sur le plan de l'information et du contenu à afficher que sur la mise en page, les typos, les images, etc....

2. Plus de connaissances et plus de technique

Pour être un crac en responsive design, il y a beaucoup de choses à connaître car malheureusement il ne suffit pas de juste connaître les media-queries CSS3 sur le bout

des doigts ainsi qu'une ou deux astuces. On est souvent confronté aux spécificités de certaines plates formes (comme l'iPhone qui réagit de façon très particulière et qui peut nécessiter du code supplémentaire propre à Safari dans sa version mobile). Et surtout ce n'est pas sur le point de s'arrêter ! Bien au contraire.

3. Un coût plus élevé

Et oui si c'est plus long et que ça demande plus de compétences, le devis va forcément s'alourdir si on ne veut pas privilégier certains utilisateurs et en délaisser d'autres.

4. Moins de créativité

Plus on apporte de contraintes de mise en page et de contraintes techniques, plus on réduit sa liberté. Ce qui apparaît clairement avec les designs responsive si on jette un œil aux galeries, est qu'on sent très fortement la mise en page en grille : tout est assez carré, aligné, prêt à être replacé au moindre redimensionnement (c'est le concept qui veut ça !), pas ou très peu de large background ou même d'images léchées à tout va, et encore plus rarement de la vidéo ou du rich media.

5. Le cas des images

La question des images à redimensionner ou à cacher est délicate. Il ne faut pas pour autant faire perdre de force au concept dans la mesure du possible et il faut que le site se charge vite... Dernier point, redimensionner des images consomme plus de performance CPU et de mémoire.

6. Le temps de chargement

Il faut sans cesse faire attention au temps de chargement, à ne pas laisser d'éléments invisibles ou de scripts inutilisés gonfler ce dernier. N'oubliez jamais que les gens sont impatients ! Et si votre site met trop de temps à charger sur un appareil mobile, il y a de fortes chances qu'il ne se charge jamais dû à un signal réseau trop faible ou à un manque de mémoire.

7. Plus de code, plus de mises à jour

Pour adapter le contenu et la mise en page à la majorité des supports, il faut plus de code ce qui entraîne donc des temps de mises à jour plus élevés...

8. Plus de différences entre les navigateurs et supports

Le responsive design repose en grande partie sur l'HTML/CSS et comme on le sait, il est interprété différemment par chaque navigateur et de plus souvent les supports ajoutent une couche comportementale par-dessus (comme par exemple Safari mobile qui va dézoomer toute la page de votre site lors de son chargement pour l'afficher dans son intégralité). En quelques mots : un gros casse-tête.

9. Il est souvent plus rapide de créer une version mobile qu'une version responsive

Et oui, c'est tellement long et contraignant que dans certains cas, il sera plus rapide et plus efficace et agréable pour l'utilisateur de concevoir une version mobile que de chercher à adapter votre site.