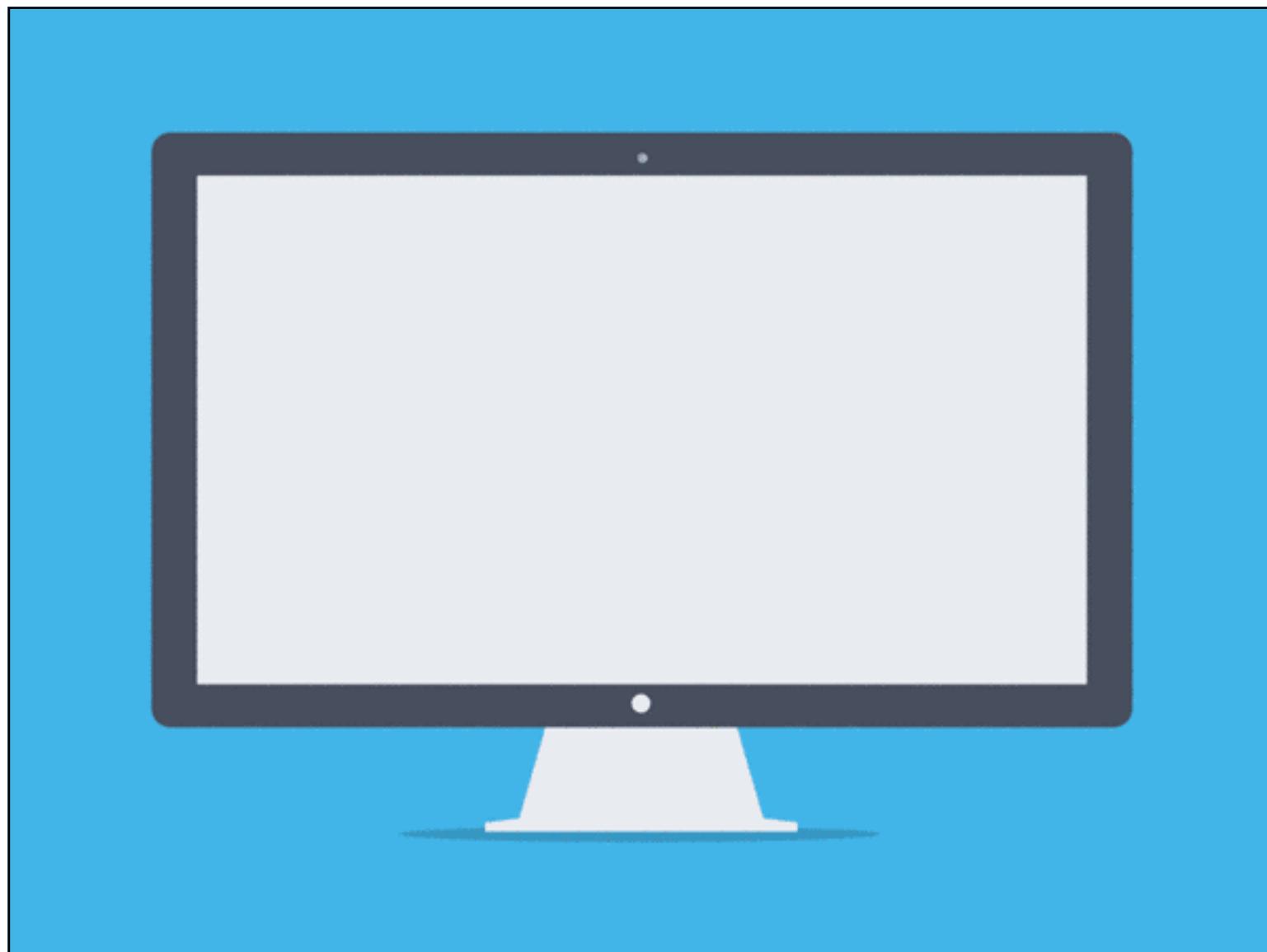


Le Responsive Web Design

N Vanassche

www.nathalievvanassche.be



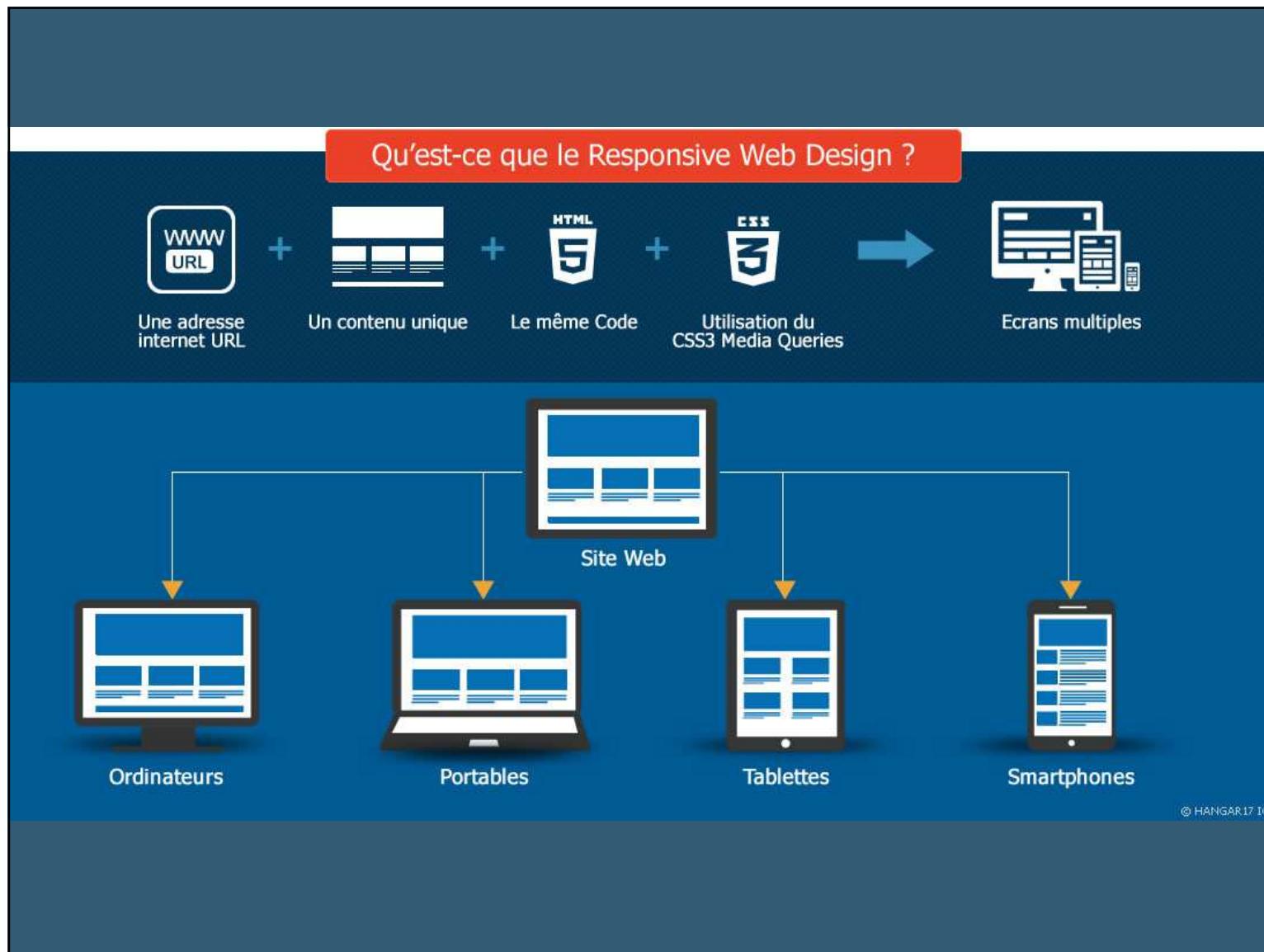
Pourquoi le RWD ?

Avant le RWD

- **On gérait 2 sites distincts**
1 pour l'ordinateur et 1 autre pour le mobile
- **On effectuait une détection du user-agent**
infos transmises par le navigateur (version du navigateur, système d'exploitation et plate-forme) et on redirigeait vers l'un ou l'autre site en fonction
- **Ces sites possédaient généralement leur propre URL** (mobile et bureau)
- **La gestion de deux sites était techniquement lourde**

Depuis le RWD

- **Un seul site pour ordinateur, tablette et mobile**
- Plus **simple** à mettre en œuvre (fichier CSS)
- Basé sur la **taille du navigateur et pas sur le user-agent**
gestion plus souple de l'affichage
- L'affichage des pages est optimisé sur mobile mais pas plus rapide
la page charge tous les éléments y compris ceux qui sont masqués par le RWD



Les avantages d'avoir un site web responsive

- 1. Les internautes préfèrent les sites responsives**
La croissance incessante du trafic mobile en Belgique comme dans le reste du monde, rend désormais indispensable le Responsive Design.
- 2. Le responsive design offre davantage de réactivité**
Les éléments qui composent le site vont s'ajuster, se redimensionner et se déplacer automatiquement dès que l'utilisateur penchera sa tablette ou son smartphone (d'un côté ou de l'autre).
- 3. Le responsive design améliore l'ergonomie et l'expérience utilisateur**
- 4. La nouvelle mise à jour de l'algorithme « Mobile-Friendly » de Google pénalise, au sein des résultats de recherche mobile, les sites considérés comme non responsive**

Introduction aux notions

Qu'est-ce que le RWD ?

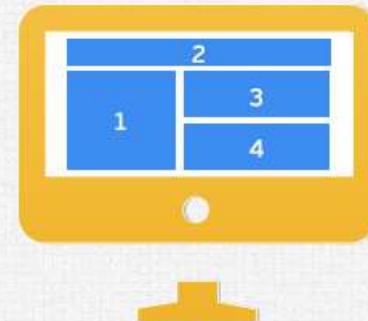
- Le Responsive Web Design (RWD) – également appelé « Site Web Adaptatif » – est une approche du web design ayant pour but l'élaboration de sites internet offrant une expérience de consultation et de navigation optimale, afin de faciliter tant la lecture que la navigation.
- L'expression « Responsive Web Design » inventée par Ethan Marcotte, désigne une méthode de travail à base de **grilles fluides**, **d'images flexibles** et de **media queries**

THE ANATOMY OF A RESPONSIVE SITE

In the development stage, there are 3 technical ingredients of responsive website design:

1

FLUID GRIDS



A percentage based design that adapts to the screen size accordingly.

2

FLEXIBLE IMAGES



Images sized in relative units to prevent them from displaying outside their containing element.

3

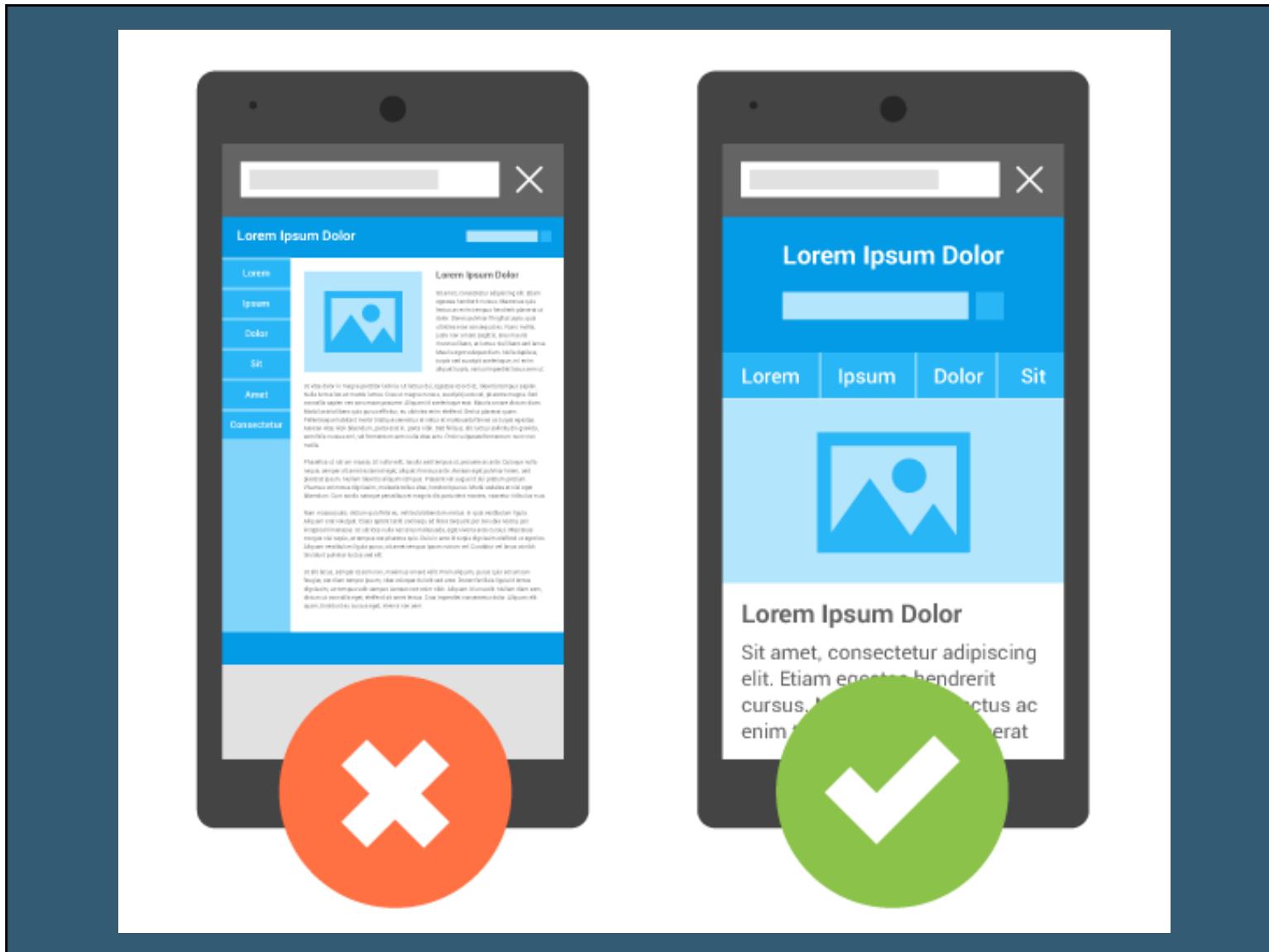
MEDIA QUERIES



A way to apply CSS rules to the page based on the size of the displaying browser.

Fluide, adaptative... responsive ??

- Une page « fluide » s'ajuste à la largeur d'écran, quelque soit la technologie derrière et indépendamment du terminal de consultation.
Il s'agit seulement d'afficher la page, son contenu, dans la largeur de la fenêtre utilisée pour la consulter.
- Une page « responsive » ou « adaptive » tient compte des ressources techniques, du terminal, du navigateur, avec ou sans JS, etc. pour servir un contenu adapté, afin d'offrir la meilleure expérience possible à l'auditoire le plus large possible.
- Tout cela recoupe les préoccupations d'accessibilité dont le but est, rappelons-le, de rendre les pages et leur contenus accessibles au plus grand nombre !



**Adoptez les bonnes
pratiques pour passer votre
site en responsive !**

1

Une mise en
page simple

1. Une mise en page simple



UTILISER

- Doctype HTML5 et normes W3C,
- Feuille de style de reset,
- Mise en page simple et sémantique,
- Techniques simples pour les éléments essentiels (navigation, menu, etc.).



EVITER

- Divs imbriqués superflus,
- Styles en ligne (Supprimez les tous !),
- Positionements absolus,
- Floats inutiles,
- Tout élément redondant ou superflu.

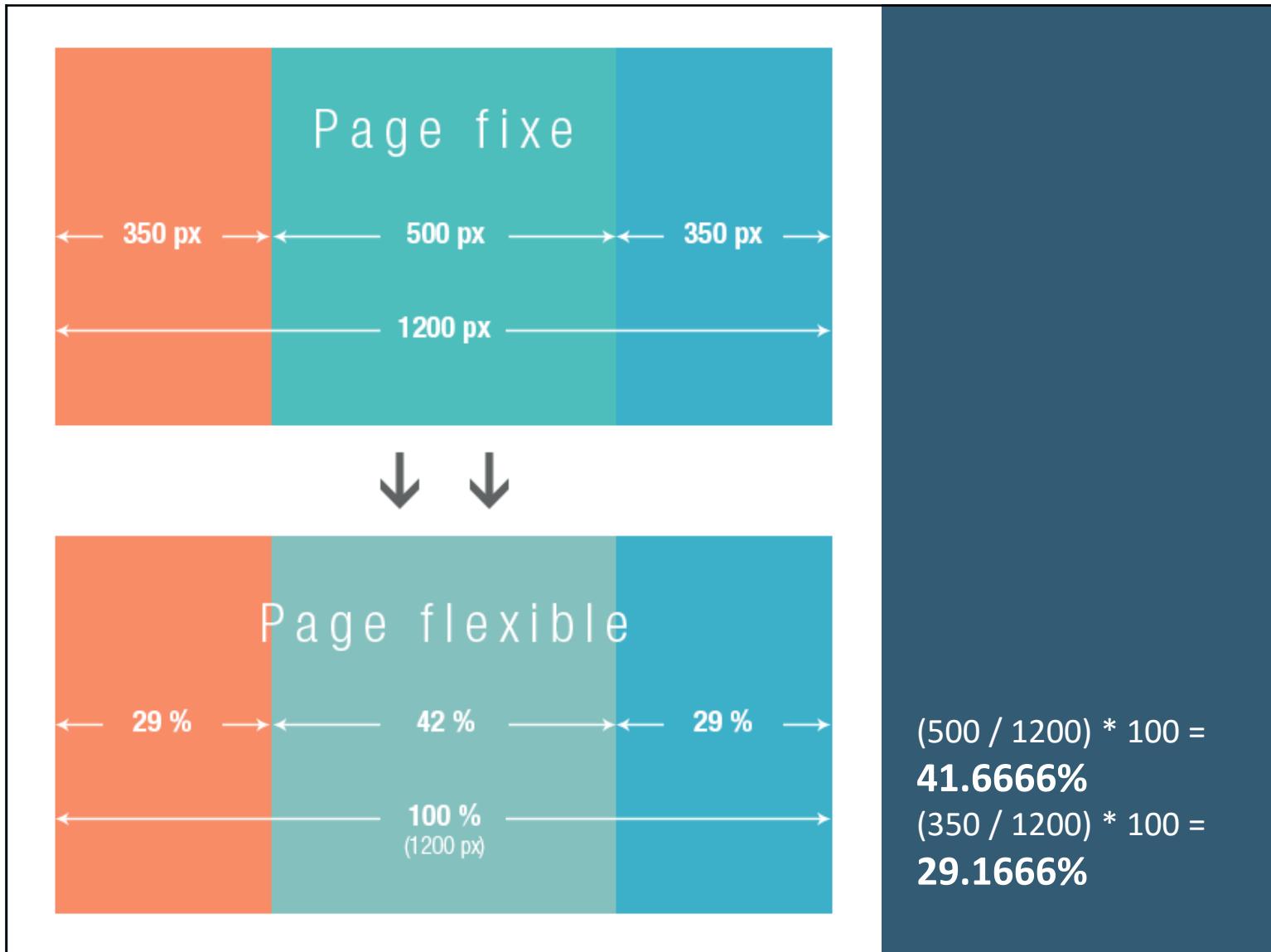
- Pensez les contenus en bloc pour un réagencement facile selon les supports.
- Optez pour un système de grille pour accélérer / faciliter l'implémentation de votre design

2

Une mise en
page flexible

2. Une mise en page flexible

- Utilisez une mise en page fluide permettra à votre design de s'adapter à toute taille d'écran et orientation.
- Les grilles flexibles utilisent des colonnes pour organiser le contenu, et un width relatif plutôt que fixe afin de s'adapter au viewport.
- Rappel, pour passer d'une unité statique (px) à une unité relative (%), il faut appliquer l'équation suivante : **(Target /Context) * 100 = result %**



3

max et min

3. N'oubliez pas max et min

- Empêchez votre design de s'élargir ou de se réduire à des proportions ridicules sur des appareils très larges ou très petits :

```
/*Fluide, mais pas moins large que 768 et pas plus grand  
que 1024*/  
  
Min-width: 768px ;  
  
Max-width: 1024px ;  
  
/*Au moins 350 de haut :*/  
Min-height: 350px;  
  
/*#main pas moins large que 1000 sur un  
ordinateur :*/  
@media screen and (min-width: 1024px) {  
    #main{min-width 1000px}  
}
```

4

unités
relatives au
viewport

4. De nouvelles unités relatives au viewport

- Les unités comme **vw**, **vh**, **vm**, **vmin** et **vmax** sont des unités de mesure relatives à la dimension du viewport.
 - 1vw est égal à 1% de la largeur du conteneur initial.
Si la largeur du viewport est de 320px, 1vw est égal à $1 \times 320 / 100$ soit 3,2pixels.
 - L'unité vh fonctionne de la même manière mais est relative à la hauteur du viewport. 50vh sont donc équivalents à 50% de la hauteur du document.
 - La différence entre utiliser ces mesures et les pourcentages ?
Les unités en pourcentage sont relatives à la taille de leur parents, alors que **vh** et **vw** seront toujours relatives à la taille du viewport, quelle que soit la taille de leurs éléments HTML parents.

5

Dompter les images

5. Domptez aussi vos images

- **Rendre vos images fluides :**

```
img { max-width : 100% ; height:auto; }
```

Le navigateur détermine la taille de l'image par rapport à l'élément parent. La hauteur de l'image est ajustée automatiquement.

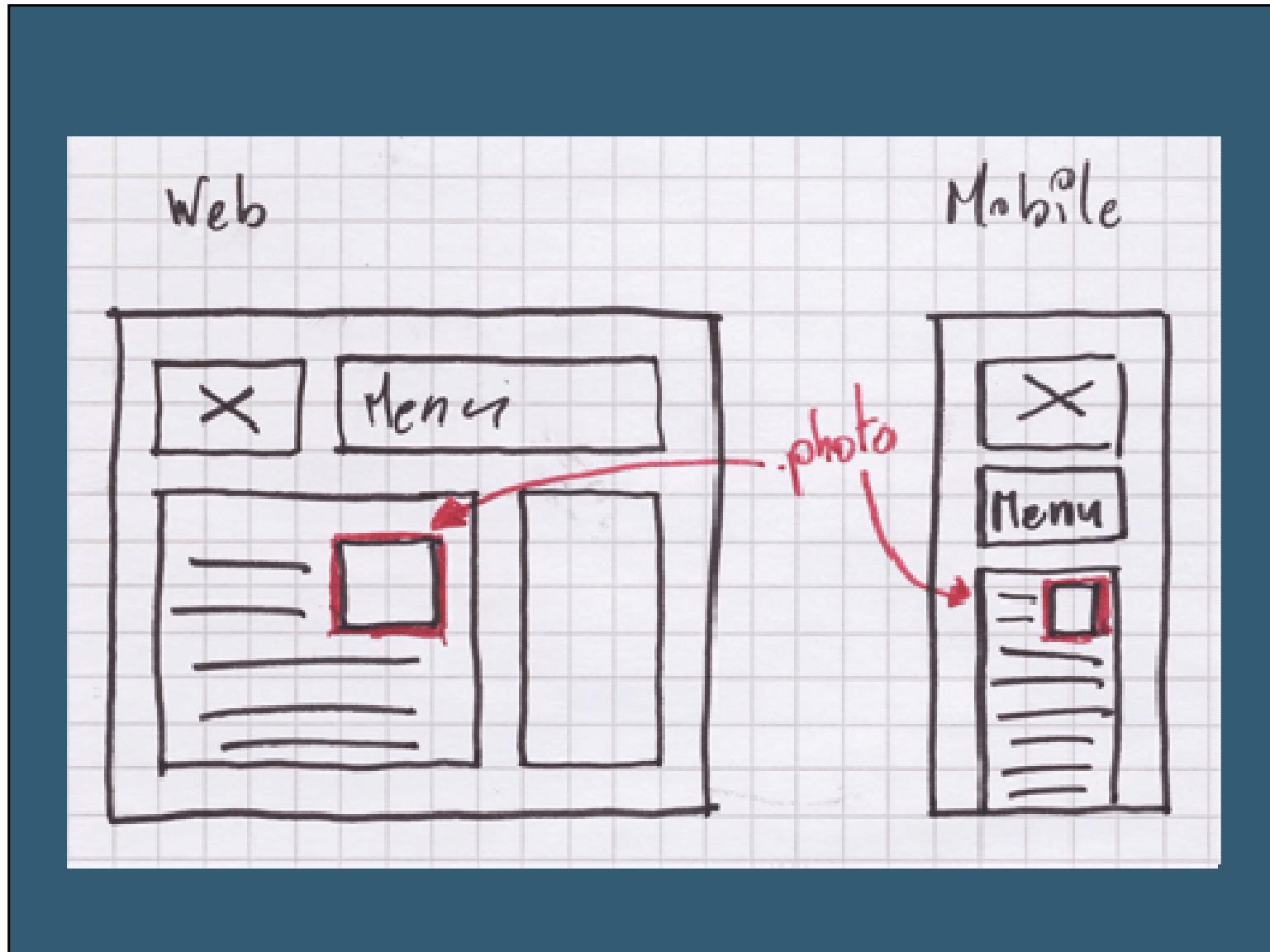
- **S'adapter aux points de rupture (images alternatives en fonction de la bande passante) :**

- L'attribut « src » d'une image ne peut pas être manipulé par un media query;

- Solutions possibles :

- Background-image : écraser l'attribut background-image d'un bloc.

- Afficher/Cacher le parent : utiliser différentes versions d'image et afficher/cacher le parent à l'aide de media queries



Éviter le débordement des images de fond

- Lorsque vous utilisez la propriété CSS `background-image` et que vous réduisez le bloc pour qu'il soit affiché correctement sur un téléphone, l'image de fond est en partie masquée.
- Veillez à utiliser la propriété CSS3 **`background-size`** : celle-ci va alors décider de la façon dont l'image de fond occupe l'espace de son conteneur.
- Jouez également avec **`background-position`** et **`background-repeat`** pour obtenir d'autres effets.

L'optimisation des images, un casse-tête !

- L'optimisation des images pour un site responsive est un vrai défi.
- Le format SVG propose une excellente alternative car il permet de créer des éléments d'interface vectoriels qui vont s'adapter à n'importe quelle résolution d'écran.
Les éléments créés en SVG vont être facilement réduits et agrandis sans perte pour les écrans à haute résolution.
- À cette tendance s'ajoute celle des **font-icons** qui permet de remplacer les glyphes d'une police d'écriture par des icônes, tout en conservant la flexibilité de la police.
- Ces solutions ne sont cependant pas envisageables pour des photos.
- CCL : Éviter d'alourdir les pages avec des images pour les résolutions mobiles

6

Et le texte
des pages

6. Les polices de caractères

- En suivant la méthode d'échelle des images, il est possible de redimensionner le texte pour faire un meilleur usage de l'espace disponible.
- Il est pertinent de régler la taille de police pour afficher plus de texte sur les petits écrans et adapter la taille des polices aux grands écrans.
- Concrètement, on définit une font-size sur le body, et on utilise des unités en em ou en % pour le calcul des éléments.

```
body {  
    font-size: 100%;  
}  
  
@media screen and (max-width: 480px) {  
    body {  
        font-size: 90%;  
    }  
}  
  
@media screen and (min-width: 1000px) {  
    body {  
        font-size: 110%;  
    }  
}
```

Conversion approximative des polices de caractère :

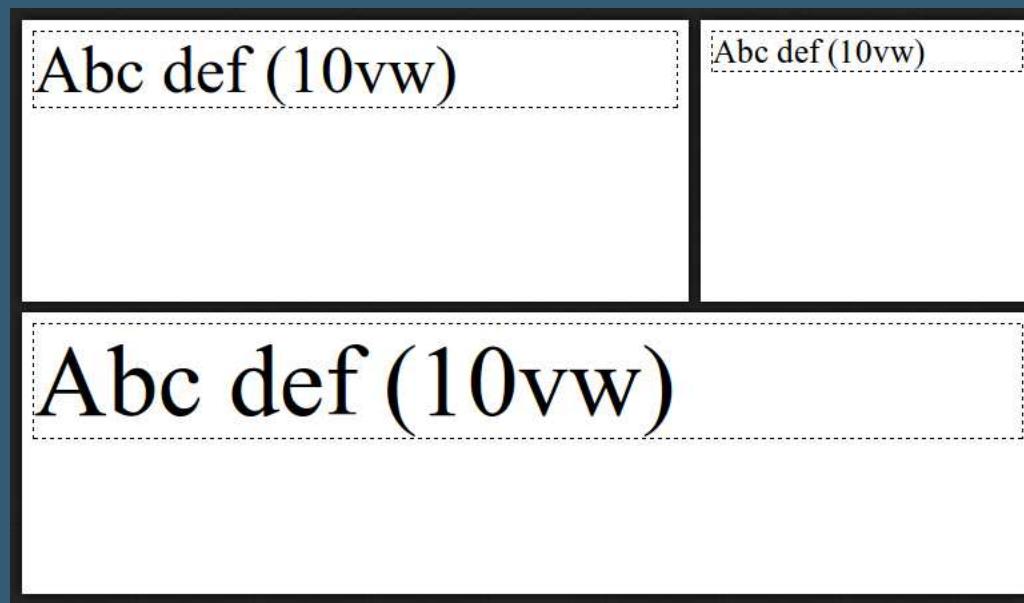
<http://reedbackdesign.co.uk/test/points-pixels.html>

6. Des tailles de polices proportionnelles à la taille de l'écran

- Les unités vw, vh, vmin et vmax sont proportionnelles à la taille de l'écran :
 - 10vw correspond à une taille de 10% de la largeur de la fenêtre du navigateur.
Si la fenêtre fait 1600px de large, c'est si la taille de police était de 160px.
Si la fenêtre est ramenée à 500px de large, alors la police sera équivalente à 50px.
 - 10vh est l'équivalent, mais pour la hauteur de la fenêtre.
 - 10vmax correspond au maximum de 10vw et 10vh : si l'écran est plus large que haut, alors ce sera la valeur de 10vw qui sera retenu.
 - 10vmin correspond au minimum de 10vw et 10vh.

6. Des tailles de polices proportionnelles à la taille de l'écran

- L'avantage est que le texte aura toujours la même proportion par rapport à la taille de l'écran, sans glisser sur plusieurs lignes :



6. Briser les lignes trop longues

- Si vous avez une ligne de texte sans espace, alors elle restera sur une seule ligne, même si elle dépasse de son conteneur. Si l'écran est trop petit, cette ligne peut casser tout le design de votre site :



6. Briser les lignes trop longues

- Exploitez les méthodes en CSS pour forcer le retour à la ligne :

```
body { word-wrap: break-word; }
```

Conclusions

1. Une mise en page simple
2. Une mise en page flexible
3. Max et min
4. Unités relatives au viewport
5. Dompter les images
6. Et le texte des pages