

# Viewport et pixel-ratio

# Comprendre le Viewport dans le Web mobile

- Le Viewport désigne schématiquement la surface de la fenêtre du navigateur.
- Cependant, la notion de viewport sur un appareil mobile est différente de celle sur un écran de bureau : sur mobile, le navigateur ne dispose pas de "fenêtre" réelle, ni de barres de défilement car tout est prévu pour naviguer de manière tactile.

# Surface réelle VS Surface « pixels CSS »

## La surface réelle

- C'est le nombre physique de pixels qui composent la matrice de l'écran, telle que le constructeur le décrit dans les caractéristiques, en gros la "résolution" (en réalité le terme juste est "définition").

## La surface en "pixels CSS"

- C'est le nombre de pixels virtuels que le terminal pense avoir et sur lequel il fonde son affichage.

# Viewport : les mobiles mentent !

- La surface en "pixels CSS" ne correspond pas toujours à la surface réelle, un "pixel CSS" n'est donc pas égal à un pixel physique.
- **Par défaut la taille du viewport d'un terminal mobile ne correspond ni à la taille de son écran réelle ni celle en "pixels CSS"... elle est généralement bien supérieure à la surface physique, afin de pouvoir y caler n'importe quelle page web en lui affectant un niveau de (dé)zoom.**
- *Les écrans Full-HD affichent par exemple un écran d'une définition de 1920×1080 pixels, dans un rectangle de 12 cm de diagonales (5 pouces). Un ordinateur avec la même définition affichera la même chose dans un rectangle de près de 43 cm (17 pouces). Le nombre de pixels par unité de surface est donc beaucoup plus élevée sur un téléphone.*

# La solution : adapter le pixel-ratio

- De plus, **la valeur initiale du viewport ne dépend pas du terminal**, comme on pourrait le supposer, mais... **du navigateur mobile**
- Si l'on ne fait rien, la page web s'affichera de la même façon sur un téléphone que sur un ordinateur, seulement en beaucoup plus petit
- Il faut donc dire au navigateur que vu qu'il possède plus de pixels par unité de surface, il doit adapter son mode d'affichage ... en précisant **un pixel-ratio**
- **Il est recommandé de laisser l'OS du téléphone calculer automatiquement le bon pixel-ratio pour le transmettre au navigateur, il suffira dès lors de dire au navigateur d'afficher la page avec le pixel-ratio du système**

# Niveau de zoom initial

- Compte tenu de ces différentes surfaces, les pages web s'affichent par défaut de manière à ce que toute la surface entre dans celle de l'écran.
- Afin de forcer le bon niveau de zoom aux navigateurs mobiles, et d'imposer la taille de la surface du viewport réelle d'un périphérique mobile, il est nécessaire d'inclure une balise `<meta>` "viewport" au sein du `<head>` de votre document HTML:  
`<meta name="viewport" content="width=device-width, initial-scale=1.0, shrink-to-fit=no" />`

Lorsque le meta viewport est défini comme device-width, le navigateur utilise, la taille réelle de l'appareil.

- Le `initial-scale=1.0` signifie au navigateur qu'il doit initialiser les dimensions en se basant sur le pixel-ratio du système (l'utilisateur peut toujours zoomer s'il le souhaite). `shrink-to-fit=no` corrige un bug iOS.

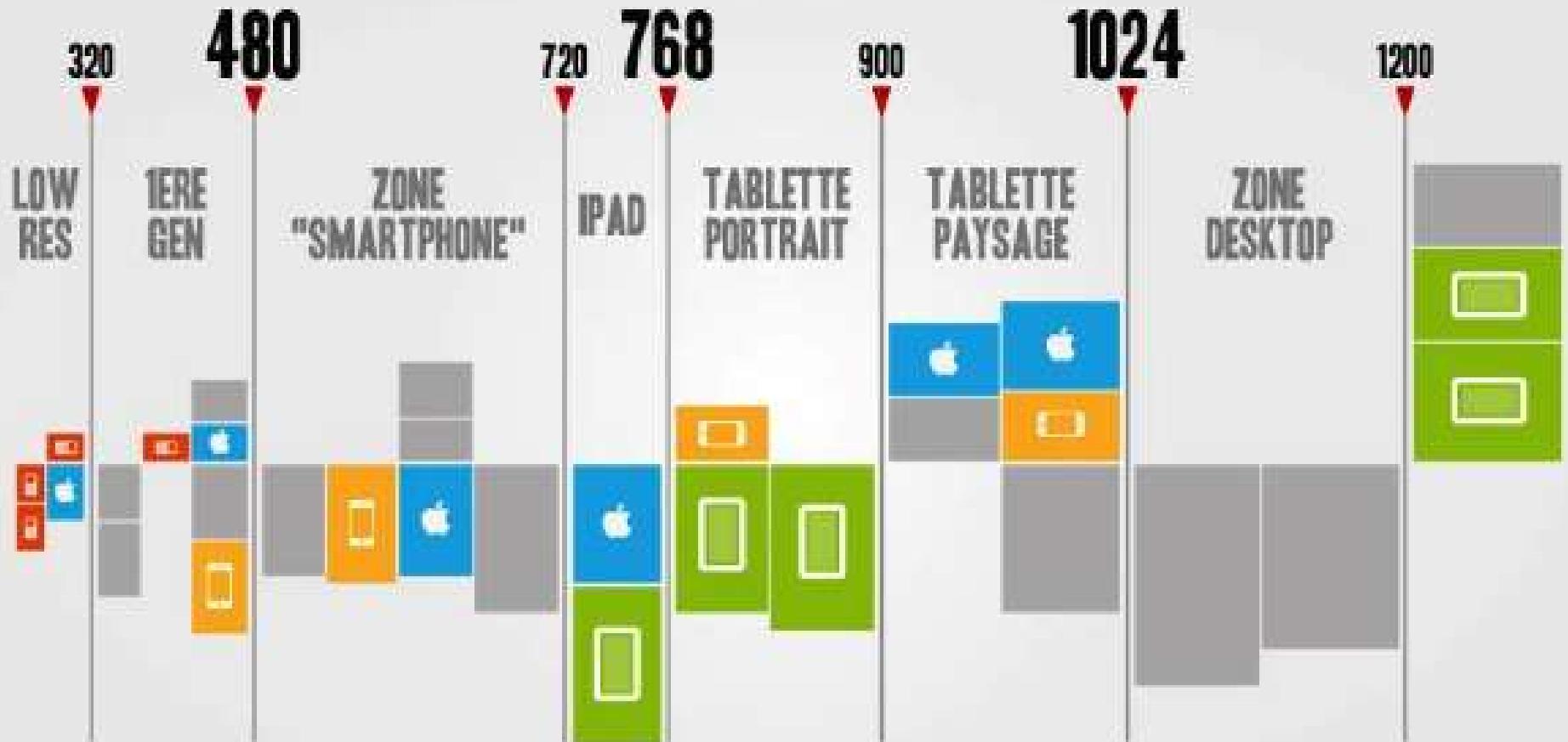
## En savoir plus ...

- <http://www.alsacreations.com/article/lire/1490-comprendre-le-viewport-dans-le-web-mobile.html>

A quelle taille d'affichage  
doit-on s'adapter ?

# Adapter nos médias-queries

- Les media queries sont un puissant outil de conditions : [SI TAILLE X = AFFICHER STYLE Y]
- Il faut donc faire en sorte d'adapter au mieux l'affichage en posant nos conditions au sein des media queries, celles-ci seront les frontières entre un affichage et un autre. En Web, nous appelons ces frontières les points de rupture
- Les points de rupture représentent le moment où l'affichage d'un site responsive design change en fonction de la largeur/hauteur de la fenêtre du navigateur. Ces points de rupture sont comme des paliers qui, une fois franchis, permettent d'adapter le design ou le contenu à la nouvelle largeur de la page web.



- Basse résolution standard
- Smartphones populaires
- Appareils Apple
- Tablettes, netbooks
- Autres appareils variés



- 3 **points majeurs** : 480 / 768 / 1024
- 3 **points mineurs** : 320 / 720 / 900
- Limite desktop aux environs de 1024px

# Choisir ses points de rupture ...

- Smartphone, phablette, tablette, desktop, portrait, paysage, largeur, densité de pixels, vrais pixels, faux pixels, devices passés, actuels, futurs, etc... Le champs des possibles rend tout simplement... impossible la détection d'un device à l'aide de Media Queries  
Ce site devrait vous convaincre : <http://screensiz.es/>
- La seule méthodologie viable actuellement est de **ne pas baser ses points de rupture sur des devices mais de se servir de son propre design.**  
Quand le design casse, on ajoute un point de rupture. Ainsi, celui-ci devrait s'adapter à toutes les surfaces...

# Media queries utilisées dans Twitter bootstrap

```
/* Landscape phones and down */
@media (max-width:480px) { ... }

/* Landscape phones to portrait tablet */
@media (max-width:767px) { ... }

/* Portrait tablet to landscape and desktop */
@media (min-width:768px) and (max-width:979px) { ... }

/* Large desktop */
@media (min-width:1200px) { ... }
```

# Quelques exemples réussis

- <http://www.spatuleprod.com/>
- <https://www.webpagefx.com/websiteredesign.htm>
- <https://huemor.rocks/>
- <http://www.creavea.com/>

# Pratique !

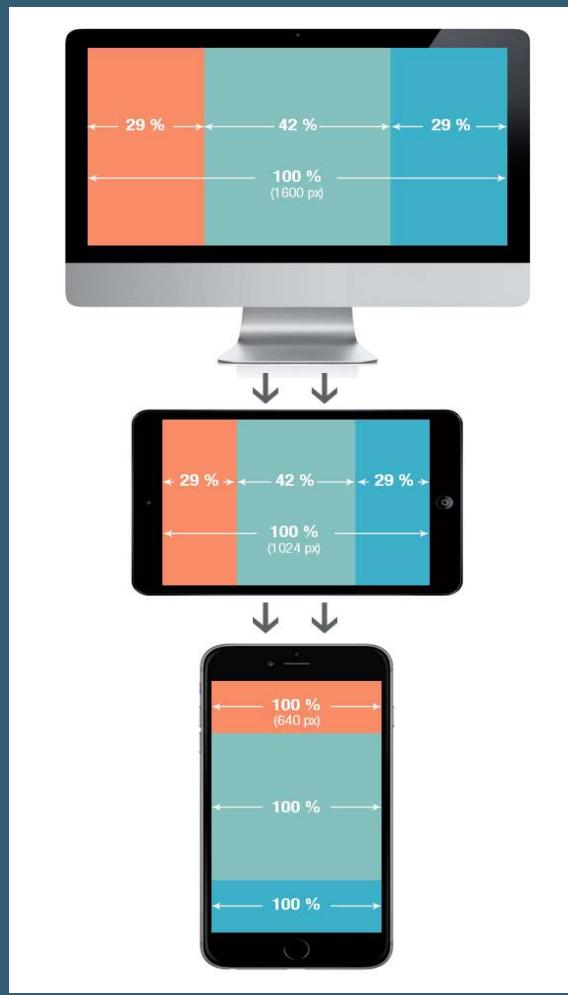
[Voir 02-RWD > Exercice Final](#)

# Un design, contenu & ergonomie adaptés à l'usage du mobile

- Il faut réussir à organiser le contenu pour les mobiles de basse résolution en une seule colonne tout en restant attentif à conserver une taille de page raisonnable. Il ne faut pas hésiter à retirer du texte si nécessaire. L'idée est de mettre l'essentiel !
- Il faut adapter les zones d'interaction tactile : sur un site adapté pour mobiles, les boutons seront plus grands afin de faciliter l'accessibilité aux doigts de l'utilisateur, par exemple.



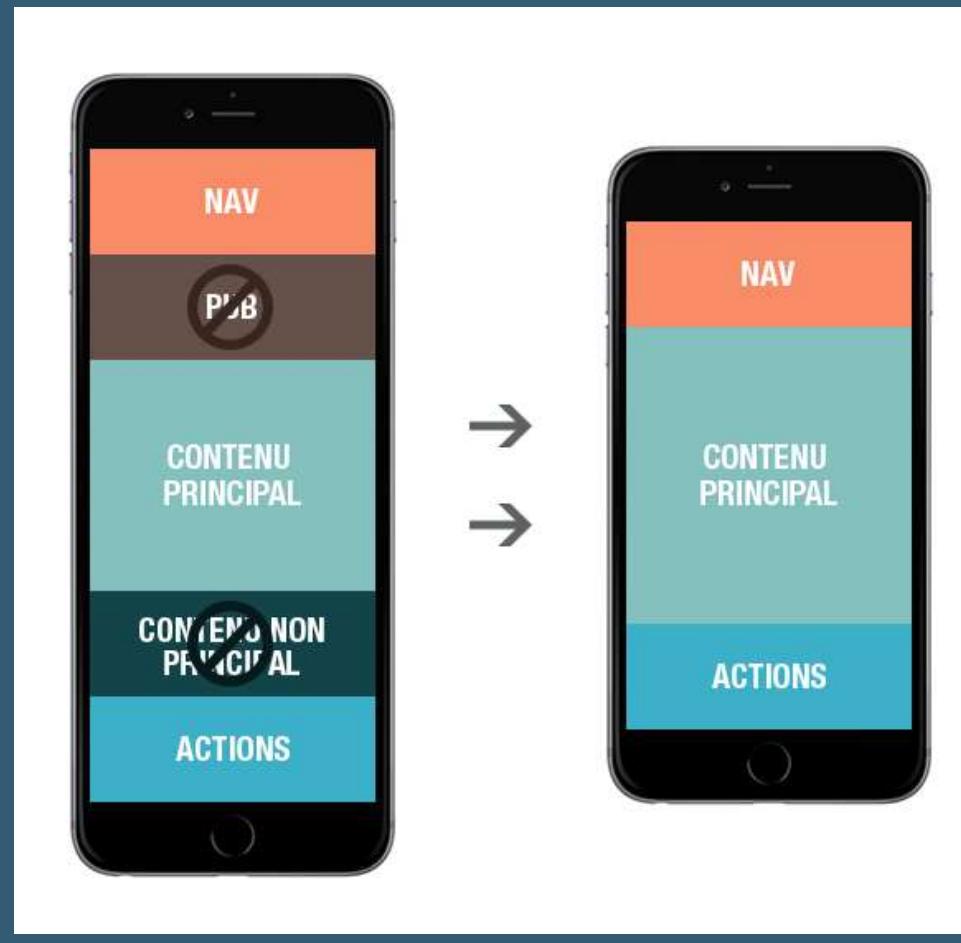
## 5. Sur mobile, on linéarise



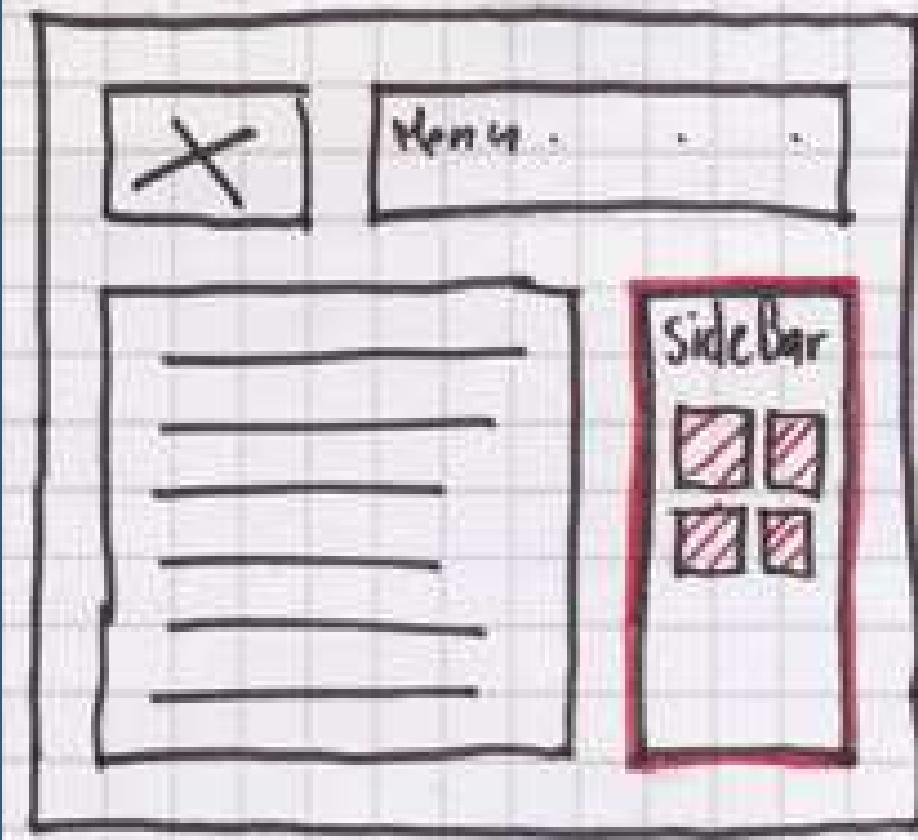
# La hiérarchie de l'information joue un rôle important !

- Il est nécessaire d'identifier les contenus qui seront prioritaires en fonction des appareils : la barre latérale a-t-elle toujours un rôle sur mobile ? Le header peut-il être minimisé, voire disparaître ? Faut-il mieux arriver sur la page par le contenu et proposer une navigation secondaire en fin de page ?
- Il ne s'agit pas simplement **d'empiler ces boîtes les unes au dessus des autres** dans une colonne unique pour les résolutions mobiles, mais **d'évaluer la pertinence du contenu** en fonction du contexte.

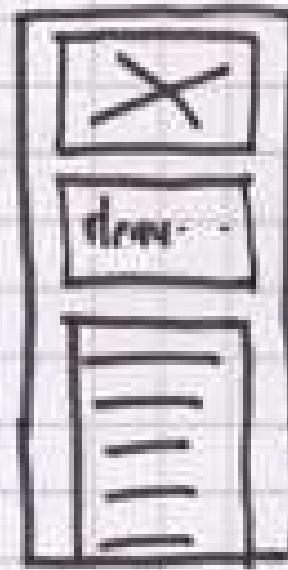
## 6. Supprimez le non essentiel



Web



Mobile



- ```
@media screen and (max-width: 480px) { .sidebar { display: none; }}
```
- Attention n'abusez pas de cette technique car le contenu de la page sera téléchargé. Cela réduit les performances, notamment en navigation mobile où le débit n'est pas toujours élevé.

# Conclusions

- Est-ce que les valeurs (arbitraires ?) de 640px et 1024px sont-elles parfaitement adaptées à votre design, ou d'autres valeurs lui siéraient mieux ?
- Il n'y a pas de bonne et de mauvaise pratique... Cela dépend de beaucoup de contraintes, notamment temporelles et budgétaires.