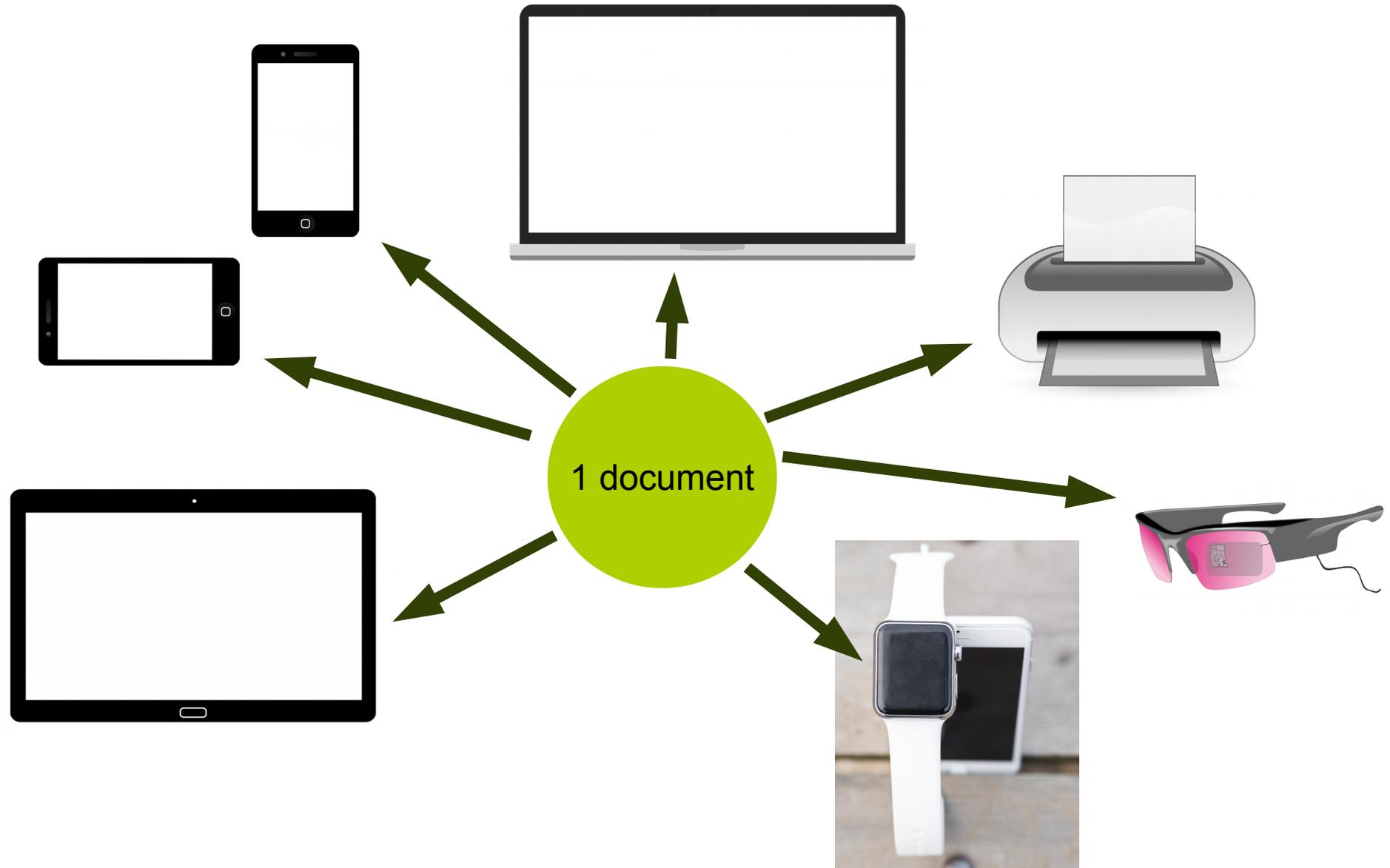


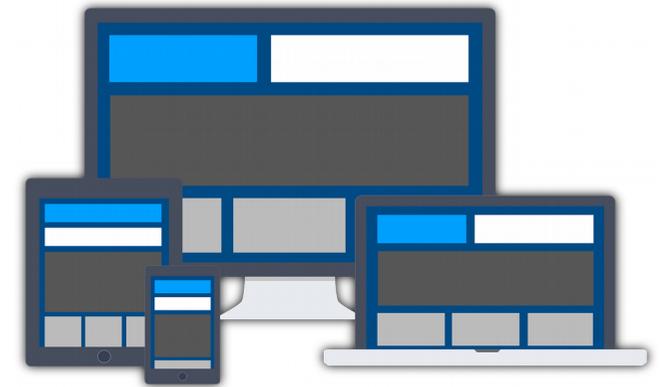
mécanisme d'adaptation de documents

- les instruments d'accès au web sont multiples et variés : smartphone, tablettes, écrans HD, TV connectée ...
- 1 document/interface est donc susceptible d'être affiché sur une multitude d'écrans aux caractéristiques variées et multiples

- 1 document → 1 présentation adaptée à chaque terminal



- **1 problème de conception** : comment afficher la même interface html sur des écrans de tailles et caractéristiques différentes



- taille et disposition des éléments
- images

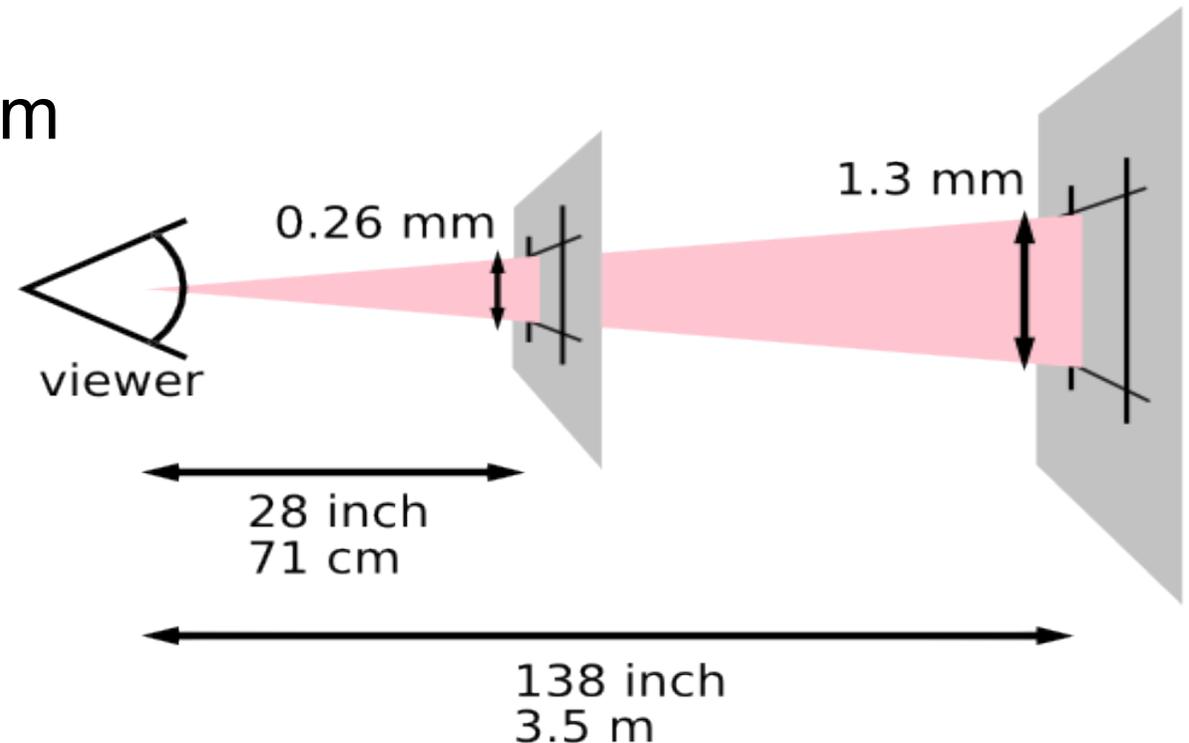
- **1 problème technique** : comment exprimer l'adaptation d'1 même document à différents médias d'affichage
 - media-query css : règles CSS comportant des conditions sur les caractéristiques de l'écran

les écrans vs. le navigateur

- 1 écran = 1 matrice de pixels physiques
- la **définition physique** d'un écran = nombre de pixels physiques en largeur x hauteur :
 - HD : 1280 x 720, Full HD : 1920 x 1080 ...
 - iPad Pro : 2048 x 2732
- la **résolution physique** d'un écran = la définition de l'écran ramenée à sa taille réelle
 - en pixel par pouce (ppp) ou dot per inch (dpi)
 - $1920 \times 1080 / 24'' \rightarrow 92 \text{ dpi}$
 - $1920 \times 1080 / 15.6'' \rightarrow 141 \text{ dpi}$
 - iPad Pro $2048 \times 2732 / 12.9'' \rightarrow 265 \text{ dpi}$

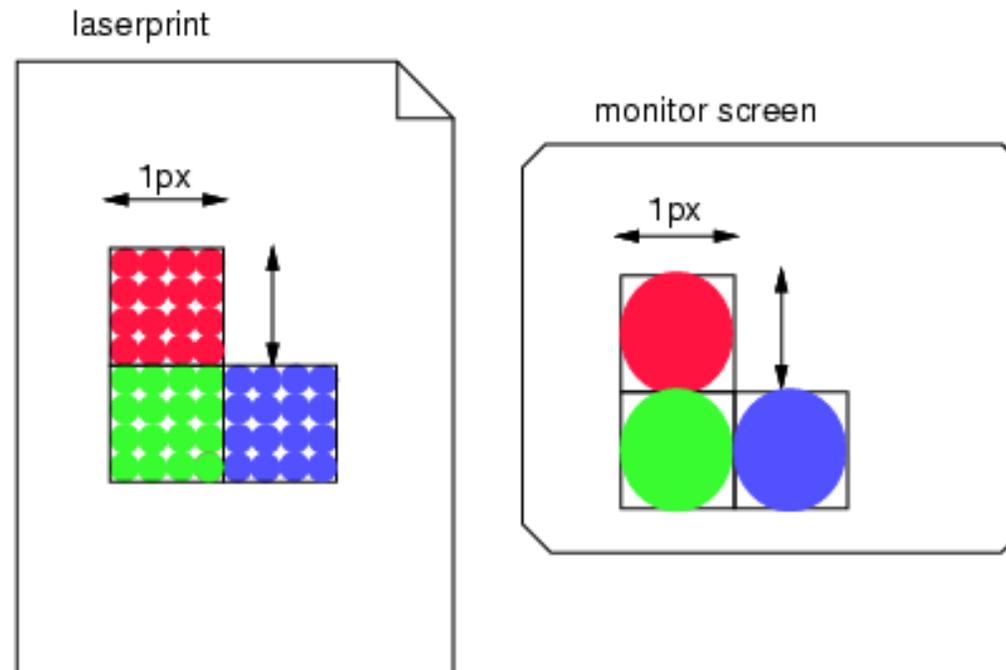
les écrans en CSS, l'unité px

- **1 px css != 1 pixel d'écran**
- **la définition px du W3C** : angle visuel correspondant à 1 pixel sur 1 écran 96dpi physique à 1 distance de 71cm
 - 0.0213 degrés / 0.26mm



les écrans en CSS, l'unité px

- **conséquence** : sur certains écrans (notamment tablettes et téléphones) ayant une résolution physique élevée, 1 **px** = 1.5 pixels, 2 pixels, 3 pixels ...
- le rapport **pixel physique / px css** est nommé **css pixel density** ou **device pixel ratio**



<https://www.w3.org/TR/css3-values/#reference-pixel>

 = 1 device pixel

les écrans en CSS, l'unité px

- le pixel ratio n'est **pas forcément égal** à : `resolution-physique/96`, il est **fixé** par le constructeur de l'écran et souvent **arrondi** à une valeur entière ou .5 : 1, 1.5, 2, 3 ...
- **conséquence** : la taille physique d'affichage **1px** est **variable** et dépend de l'écran
- en CSS, **px** est l'unité canonique à partir de laquelle sont définies les unités de taille absolues
- **96px = 1in = 72pt = 2.54cm**
- **ces tailles sont aussi variables selon l'écran**

les écrans en CSS, l'unité px

- le device pixel ratio détermine
 - la **définition css** de l'écran : largeur px X hauteur px
 - la **résolution css** de l'écran = **nb pixel physiques par inch css**

écran	phys. def.	CSS def.	pixel ratio	CSS ppi	physical ppi
iphone 7	750x1334	375x667	2	192	326
galaxy S7	1440x2560	360x640	4	384	534
Surface	768x1366	768x1366	1	96	148

- unités de résolution css : **1dppx = 96dpi**

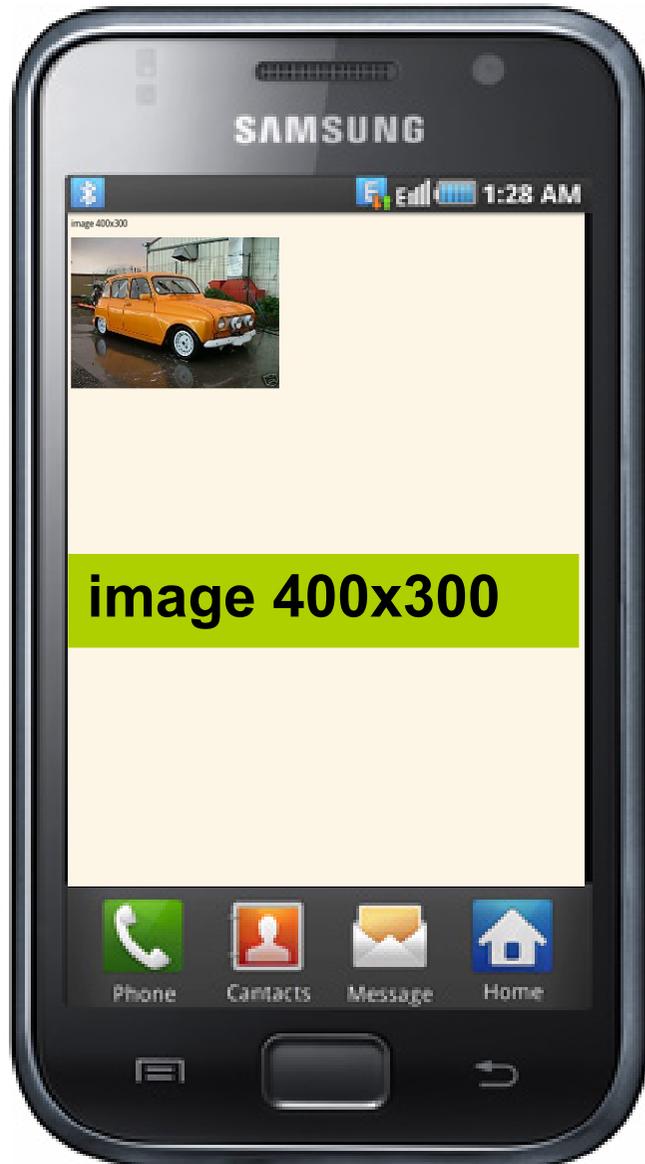
le viewport

- le **viewport** = zone d'affichage utilisée par le navigateur
- la taille du viewport est **variable** sur un navigateur desktop
 - on peut faire varier la taille de la fenêtre
- la taille du viewport est **fixe** sur un navigateur mobile

le viewport

- 1) la taille du viewport dépend du *navigateur* et pas du *terminal* !
- 2) les navigateurs **mentent** sur la taille du viewport et annoncent une valeur différente de la réalité !
 - mais pourquoi mentir comme ça ?
 - pour afficher les sites non responsifs en dé-zoomant

exemple : galaxy S



définition physique : 480 x 800
résolution physique : 233ppp

pixel ratio : 1.5

définition css : 320px x 533px
résolution css : 144 dpi



pour éviter le problème

- fixer le viewport à la taille du terminal :
 - technique actuelle : dans le document html

```
<head>
  <meta charset="UTF-8">
  <title>minipage responsive</title>

  <meta name="viewport" content="width=device-width,
                                initial-scale=1.0">

  <link rel="stylesheet" href="resp_1.css">
</head>
```

les média-query css

- construction CSS permettant de décrire des règles dont l'application est conditionnée à des caractéristiques (css) du média de sortie
- placées à l'intérieur des feuilles de styles
- Traitées par le navigateur

- Pour une présentation détaillée, lire :
https://developer.mozilla.org/fr/docs/Web/CSS/Requêtes_média/Utiliser_les_Media_queries

syntaxe

```
@media [screen|print|speech|all] /* type du média */
    and (<cond>=val) and ... , /* conditions sur */
    (<cond>=val) and ... /* les propriétés */
                                /* du média */
{
    body, p { ... } /* css rules */
    h1 { ... }
    a:hover { ... }
}
```

processus d'application

- Si la condition est **fausse** : les règles sont **ignorées**
- Si la condition est **vraie** : les règles sont **prises en compte**
- Elles s'additionnent aux précédentes et sont triées de manière classique : héritage, cascade et spécificité

```
/* fichier css */  
x { ...  
}  
.abc { ...  
}  
  
@media <condition>  
{  
    ...  
}  
  
@media <condition2>  
{  
    ...  
}
```

les conditions

- **média de sortie** : `all | screen | print | speech`
- **caractéristiques média** (*media features*)
 - **taille et orientation du viewport** (en em ou px)
 - `width, min-width, max-width, height, min-height, max-height`
 - `orientation : landscape | portrait`
 - **format du viewport** :
 - `aspect-ratio : 4/3 | 16/9`
 - **résolution css de l'écran** :
 - `resolution, min-resolution, max-resolution, en dppx ou dpi`
 - **taille et aspect du terminal**
 - `device-width, device-aspect-ratio`

quelques exemples

```
@media screen and (min-width: 25em) {  
  body {font-size: 1em ; }  
}
```

```
@media screen and (min-width: 25em) and  
  (orientation:landscape) {  
  body {font-size: 1.2 em ; }  
}
```

```
@media screen and (min-width: 25em) and  
  (orientation:landscape),  
  (min-width: 25em) and  
  (aspect-ratio:16/9) {  
  body {font-size: 1.2 em ; }  
}
```

quelques exemples

```
@media print {  
  body {  
    width : 210mm ;  
    min-height : 297mm ;  
    color : black ;  
    background-color : unset ;  
    font-size: 12pt ;  
  }  
}
```

Recommandations

- "mobile-first" : le cas par défaut (sans media-query) doit correspondre au **mobile en portrait**
- Les média-query gèrent les élargissements
 - mobile portrait → paysage → tablette → écran
 - Utiliser des conditions « min-width »
- ne pas tenter de prévoir 1 média-query par terminal, mais se baser sur le contenu pour déterminer les points de rupture
 - privilégier les em plutôt que px
 - utiliser le px dans les cas où un basculement au pixel près est nécessaire

adaptation des médias

- images, vidéo ... : un certain nombre d'éléments risque de déborder de leur conteneur lorsque la largeur du viewport diminue
 - `max-width : 100% ;`

```
/* fixer une largeur maximale de 100 % aux éléments
potentiellement problématiques */
img, table, td, blockquote, code, pre, textarea, input,
iframe, object, embed, video {
    max-width: 100%; }
/* conserver les proportions pour une image */
img { height: auto; width: auto; }
```

des images responsives

- les img html : par défaut le navigateur les charge toutes.
- Il serait intéressant de pouvoir livrer des images différentes ou de définitions différentes selon le terminal
 - éviter de transférer une image HD 2560x1440 sur un téléphone
- les navigateurs modernes permettent de choisir l'image en fonction de conditions liées au device

rappel

- le **chargement des images** (balise ``) est l'une des premières tâches effectuées par le navigateur, **avant l'application des règles css**
- **donc**, des conditions de choix d'image ne peuvent porter que sur des grandeurs connues à ce moment là :
 - taille du viewport
 - pixel ratio css
 - dimensions de l'image

srcset et size

- l'attribut srcset permet de spécifier un ensemble de fichiers correspondants à 1 image avec différentes caractéristiques
- le navigateur choisit la version la plus adaptée
 - il ne les agrandit jamais
- On peut indiquer, pour chaque fichier :
 - soit la densité de pixels cible
 - soit la largeur (définition) des images

- choix en fonction de la densité de pixels

```

```

2 versions et la densité cible

pour les vieux navigateurs

```


360px



720px



1440px



- le navigateur choisit la version de l'image en fonction du viewport dont il dispose

- on peut également préciser la place allouée à l'image pour son affichage, si on la connaît. Attention, le navigateur redimensionne l'image !

```

```

L'image occupera  
50 % du viewport  
en largeur

1440px

- avec 1 viewport de 1440
  - l'image 1024 est choisie car sa taille finale est 50vw
  - l'image est effectivement redimensionnée



- l'attribut size peut recevoir une média-query

```

```

>

700px



820px



360px



900px



# le conteneur <picture>

- *The **picture** element is a **container** which provides **multiple sources** to its **contained img element** to allow authors to declaratively **control** or give hints to the user agent about **which image resource to use**, based on the screen pixel density, viewport size, image format, and other factors. It represents its children.*

# élément `<picture>`

- contient obligatoirement 1 `<img>` : c'est l'élément qui affiche réellement une image et que l'on peut dimensionner
- contient plusieurs `<source>` : ressources media alternatives, choix sur une condition de type media-query - permet de choisir l'image en fn du média
- chaque source contient
  - 1 attribut `media` pour sélectionner la source
  - 1 attribut `srcset` contenant plusieurs alternatives
  - éventuellement, un attribut `sizes`

```
<picture>
```

```
 <source media="(min-width: 640px)"
 srcset="img/cat-hd.jpg 1900w,
 img/cat-md.jpg 1024w"
 sizes="50vw">
```

```
 <source srcset="img/crop/cat-closeup-hd.jpg 2x,
 img/crop/cat-closeup-sm.jpg 1x" >
```

```

```

```
</picture>
```

# utilisation en pratique

Can I use picture ?  Settings

2 results found

## Picture element - LS

A responsive images method to control which image resource a user agent presents to a user, based on resolution, media query and/or support for a particular image format

Usage	% of all users
Global	87.63%
France	88.21%

Current aligned Usage relative Date relative Show all

IE	Edge *	Firefox	Chrome	Safari	iOS Safari *	Opera Mini *	Chrome for Android	UC Browser for Android	Samsung Internet
			49						
			63						
			66		10.3				
			67		11.2				4
11	17	61	68	11.1	11.4	all	67	11.8	7.2
	18	62	69	12	12				
		63	70	TP					
			71						