

# ***Cascading Style Sheets – Héritage et typographie***

- 1. L'héritage**
- 2. L'unité de mesure em**
- 3. Stratégie de la base décimale**
- 4. Taille par défaut des polices dans le navigateur**
- 5. em versus rem (*root em*)**
- 6. Les pixels en solution de repli au rem**
- 7. Une typographie réactive**
- 8. Polices liées, @font-face et formats**

# 1. L'héritage en CSS

Quels éléments héritent de quels autres éléments ?

Quelles sont les propriétés qui peuvent être transmises aux descendants?

L'héritage des styles CSS est une notion intrinsèquement reliée à l'arborescence HTML du document.

Un grand nombre de styles sont transmis d'un parent à toute sa descendance.

Par exemple: les propriétés `color` et `background-color` sont transmises... le `font-size` aussi... mais pas les bordures!

## La propriété inherit

Pour forcer l'héritage d'un parent sur une propriété, utiliser la valeur `inherit`  
Par exemple, pour empêcher le changement de police dans les formulaires *normalize.css* définit:

```
button,  
input,  
optgroup,  
select,  
textarea {  
    font-family: inherit;  
    font-size: 100%;  
    line-height: 1.15;  
    margin: 0;  
}
```

## 2. Les em

Pendant longtemps, on a recommandé l'utilisation des **em** pour définir les tailles de police dans les CSS parce que cette unité de mesure est relative, donc ajustable par l'utilisateur.

D'un autre côté, les designers maquette avec des tailles de polices en **pixels**... Une stratégie a donc été mise en place pour faciliter le travail de **conversion des pixels en em**.

## 3. Stratégie de la base décimale

Pour répondre au besoin de convertir les pixels en em facilement, des développeurs ont élaboré une stratégie consistant à travailler avec une base décimale pour que toute valeur en em puisse être **multipliée par 10** pour donner son équivalent en pixels :

1em = 10 pixels

1.2em = 12 pixels

1.4em = 14 pixels

## 4. Taille par défaut des polices de caractères dans un navigateur

Pour que cela fonctionne, il faut tenir compte de la valeur par défaut de la taille de la police par défaut dans les navigateurs qui est fixé à 16 pixels.

**Par défaut, la valeur de 1em est donc égale à 16 pixels.**

L'équation pour obtenir que la valeur de 1em = 10px en fonction de la taille de la police par défaut dans les navigateurs s'énonce comme ceci: **valeur voulue ÷ valeur du contexte = résultat.**

$$\begin{aligned} \text{L'équation } & \text{valeur voulue} \div \text{contexte} = \text{résultat} \\ & 10\text{px} / 16\text{px} = 62.5\% \end{aligned}$$

## MINI-EXERCICE |

### Calculer le rendu en px de la taille des polices définie en em

<https://codepen.io/evefevrier/pen/EjGyZM>

Observer dans le html, l'emplacement des deux `<h1>`.

Observer dans les CSS, la valeur des propriétés `font-size`.

#### Calculer le rendu en px

du `font-size` de `<section>`

(Hérite du parent `<body>` : 10px) x 1.25em = \_\_\_\_\_

du `<h1>` qui est dans `<section>`

(Hérite de `<section>` : \_\_\_\_\_) x 2em = \_\_\_\_\_

Utilisez l'inspecteur de code pour valider vos réponses :

- sélectionner l'élément à vérifier,
  - faire un clic droit de la souris et choisir *Inspecter*
  - cliquer sur l'onglet **Computed**
- permettra de voir les styles calculés par le moteur de rendu

## 5. em versus rem (*root em*)

« L'unité **em** est relative à la taille de la fonte du parent, ce qui crée un effet de cascade problématique.

L'unité **rem** est relative à l'élément racine : la balise html.

Cela signifie que nous pouvons ne définir qu'une seule taille de police sur l'élément **html**, puis composer toutes les tailles en rem comme pourcentages de cette valeur initiale. »

*Jonathan Snook, Dimensionner ses fontes avec rem*  
<http://www.pompage.net/traduction/dimensionner-ses-fontes-avec-rem>

## 6. Les pixels en solution de repli au rem

La proposition de Jonathan Snook permet de travailler en base décimale sans subir les problèmes d'héritage des em.

Le font-size est défini une première fois en pixels pour les navigateurs qui ne supportent pas les rem.<sup>1</sup>

Le font-size est redéfini ensuite en rem pour les navigateurs qui supportent cette unité de mesure.

---

<sup>1</sup> En 2019, les rem sont très largement supportés, la solution de repli ne sera utile que si vous devez assurer la portabilité pour des versions plus anciennes de navigateurs. Les versions de Internet Explorer 8 et moins ne supportent pas les rem, les versions 9 et 10 offrent un support partiel. À partir de la version 11 et avec Edge, le support est complet.



```
html{  
    font-size: 62.5%;  
}  
body {  
    font-size: 16px;  
    font-size: 1.6rem;  
    line-height: 1.5;  
}  
h1    {  
    font-size: 48px;  
    font-size: 4.8rem;  
}
```

## 7. Une typographie réactive (*responsive*)

Une typographie responsive est un ensemble de règles visant à maintenir une hiérarchie visuelle cohérente tout en adaptant cette hiérarchie aux contraintes du contexte.

### (1) Redimensionner la police.

Redimensionner en fonction de la taille de l'écran.

Plus l'écran est large, plus loin se tiendra l'utilisateur, donc plus la taille des polices doit être grosse.

```
h1 {  
    /* = 32px */  
    font-size: 3.2rem;  
}  
  
@media (min-width: 600px) {  
    h1 {  
        /* = 40px */  
        font-size: 4rem;  
    }  
}  
  
@media (min-width: 800px) {  
    h1 {  
        /* = 48px */  
        font-size: 4.8rem;  
    }  
}
```

## (2) Contrôler l'interlignage.

L'interlignage peut être plus serré lorsque la taille des polices diminue.  
Le minimum à respecter est de 1.5.

L'interlignage correspond à la propriété css **line-height**.

**Lorsqu'il est utilisé sans unité de mesure tel que recommandé,  
le line-height est un facteur multiplicateur du font-size.**

Par exemple, un line-height de 1.375 appliqué sur un font-size de 16px obtient un interlignage de 22px.

$$1.375 \times 16\text{px} = 22\text{px}$$

```
body {  
    /* 16px pour toutes les largeurs d'écran */  
    font-size: 16px;  
    font-size: 1.6rem;  
  
    line-height: 1.375; /* 1.375 x 16 = 22 */  
  
}  
@media (min-width:600px) {  
    body {  
        line-height: 1.5;  
    }  
}  
@media (min-width:800px) {  
    body {  
        line-height: 1.75;  
    }  
}
```

### (3) Contrôler la longueur des lignes.

Maintenir la lisibilité en contrôlant la longueur des lignes.

Longueur max.: 80 caractères

Longueur min.: 3-4 mots  
ou 35 caractères (en français et en anglais)

## 8. Polices liées, @font-face et formats

Lorsqu'on sert des polices de caractères autre que celles communes aux systèmes d'opération, il faut s'assurer qu'elles sont téléchargeables dans différents formats. L'utilisation de @font-face s'impose.

Lorsqu'on utilise les *Google Fonts*, c'est cette API qui fera le travail pour nous en générant les instructions @font-face.

Tester : <https://fonts.googleapis.com/css2?family=News+Cycle&family=Roboto+Condensed:wght@300;700&display=swap>

Elle proposera d'abord le format le plus récent *woff2*, puis, s'il n'est pas supporté elle proposera les autres formats : *woff* ou *truetype*.

```
@font-face {  
    font-family: NewsCycle;  
    src:url(../polices/News_Cycle/NewsCycle-Regular.woff2) format('woff2'),  
    url(../polices/News_Cycle/NewsCycle-Regular.woff) format('woff'),  
    url(../polices/News_Cycle/NewsCycle-Regular.ttf) format('truetype');  
}
```