

front-end°

Formation avancée html/css

novembre 2018

Stéphane Dailly

Manager front-end



Fort d'une expérience de 10 ans dans le métier, à l'agence depuis 7 ans

Je mets en place une architecture qui correspond à votre projet, votre plateforme, et dirige l'équipe technique qui réalise vos projets,

Mes dernières missions : Printemps, Céline, Eres, Loxam, Group Leduff, Maison Michel...

On démarre ?

C'est parti ?



HTML

Quelques rappels



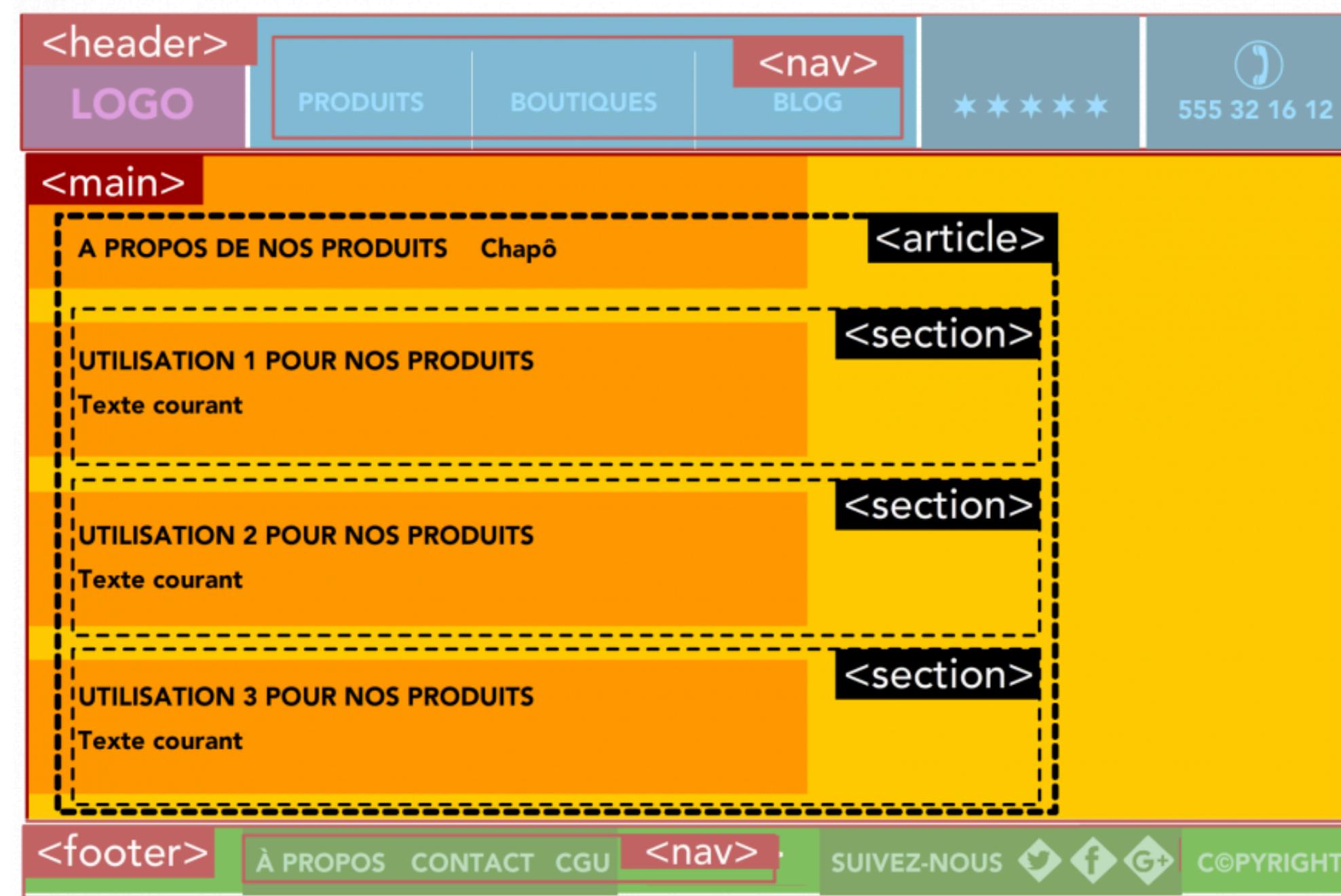
Balises de mise en page



Les balises de mise en page - HTML5

- **<main>** : Définit le contenu principal du document
- **<article>** : utilisé pour du contenu ayant son propre sens indépendamment du reste des autres éléments de la page, ce contenu est distribuable et réutilisable
- **<section>** : Utilisée pour regrouper des éléments différents, mais partageant la même thématique Cette balise est le plus souvent utilisé avec un header.
- **<aside>** : Cette balise permet de regrouper des informations non essentielles relatives au site Web. C'est un contenu complémentaire à l'article
- **<header>** : pour l'entête d'une section et/ou d'une page. Cette balise peut-être utilisée dans la balise <section> et dans la balise <article>.
- **<footer>** : pour le base de page dans une section et/ou d'une page. Cette balise peut-être utilisée dans la balise <section> et dans la balise <article>.

Les balises HTML5



Les balises HTML5

- **<datalist>** : pour une liste déroulante
- **<details>** : permet d'afficher un résumé ou une légende
- **<dialog>** : représente une boîte de dialogue ou un composant interactif
- **<nav>** : représente une section destinée à la navigation
- **<mark>** : permet de mettre en exergue du texte pertinent – sera *surligné*
- **<meter>** : représente une valeur scalaire dans un intervalle donné
- **<summary>** : permet au clic de révéler le contenu de **<details>**
- **<time>** : définit une unité de temps



CodePen

L'attribut data-*

Si vous avez besoin de stocker et récupérer une donnée depuis un élément html, vous pouvez cet attribut

```
<element data-gps="1.1212"> ... </element>
```

Quelques indications sur son utilisation :

- Doit être une String, peut être encoder sous forme de chaîne (tel JSON)
- Le nom ne doit pas contenir de point virgule
- Le nom peut contenir des tirets : data-gps-lat

Récupérer la donnée

Accéder en css

```
article::before { content: attr(data-parent); }
```

Accéder en javascript

```
var article = document.getElementById('monIdentifiant');  
article.dataset.columns
```

Il faut choisir la bonne balise !



Les images



Les figures

Les figures sont des éléments qui viennent enrichir le texte pour compléter les informations de la page.

SÉMANTIQUE

```
<figure>
    
    <figcaption>Le logiciel Bloc-Notes</figcaption>
</figure>
```

Remarque : les formats d'image conseillés sont : **jpeg, gif** ou **png**.

Les images responsives : srcset

L'attribut srcset permet de définir une image adaptée au terminal de consultation en ciblant la taille de l'écran, et également la densité de pixels.

- Le descripteur "x" sélectionne l'image en fonction de la densité de pixel (pixel ratio) de l'écran
- Le descripteur "w" sélectionne l'image en fonction de sa taille et de l'adéquation à la surface de l'écran

LE DESCRIPTEUR « X »

```

```

LE DESCRIPTEUR « W »

```

```

L'ATTRIBUT « SIZES »

L'attribut sizes permet d'indiquer au navigateur quelle doit être la largeur finale de l'image à l'affichage.

```

```

Les images responsives

```
<img  
srcset="  
https://via.placeholder.com/640x150 640w,  
https://via.placeholder.com/1200x150 1200w >  
sizes="(max-width: 1200px) 640px, 1200px"  
>
```

<https://codepen.io/sdailly/pen/LXEGjG>

Les images responsives

Utilisation de <picture>

<source>

Permet de définir la source de l'image (`srcset`) et sous quelle résolution (`media`) cette source doit s'afficher

```
<picture>
  <source
    srcset="https://via.placeholder.com/640x150"
    media="(max-width: 640px)" >
  
</picture>
```

<https://codepen.io/sdailly/pen/jezQNo>

Optimiser ses images !



CSS



```
#chucknorris {  
    color: #BADA55;  
}
```



Quelques rappels



Les sélecteurs



Les sélecteurs simples

- **Sélecteur de type :** cibler les balises qui correspondent au nom indiqué.

```
<p>...</p>
```

```
p { ... }
```

- **Sélecteur de classe :** cibler les balises dont l'attribut de classe correspond au nom indiqué.

```
<p class='maClasse'>...</p>
```

```
.maClasse { ... }
```

- **Sélecteur d'identifiant :** cibler la balise dont l'attribut d'id correspond au nom indiqué.

```
<p id='monUniqueID'>...</p>
```

```
#monUniqueID { ... }
```

- **Sélecteur universel :** cibler tous les nœuds du document.

```
<body>
  <div>
    <p>...</p>
  </div>
</body>
```

```
* { ... }
```

Sélecteur d'attribut : cibler la balise dont l'attribut correspond à l'égalité définie dans le ciblage.

- **[attr]** : ciblage direct de la présence de l'attribut
- **[attr=valeur]** : ciblage si l'attribut est égal à la valeur indiquée
- **[attr~=valeur]** : ciblage si l'attribut contient la valeur exacte indiquée
- **[attr|=valeur]** : ciblage si l'attribut est égal à la valeur indiquée ou si la valeur de l'attribut commence par la valeur indiquée suivie d'un '-'.
- **[attr^=valeur]** : ciblage si la valeur de l'attribut commence par la valeur indiquée
- **[attr\$=valeur]** : ciblage si la valeur de l'attribut finit par la valeur indiquée
- **[attr*=valeur]** : ciblage si l'attribut contient la valeur indiquée



Les combinateurs

- **Voisin direct** : cible un élément uniquement si celui-ci « suit » un élément donné et que les deux éléments sont les fils d'un même élément parent.

```
premier_element + element_cible {...}
```

- **Voisins généraux** : cible tous les éléments qui suivent le premier élément indiqué dont le parent est commun.

```
premier_element ~ element_cible {...}
```

- **Sélecteur enfant** : cible tous les éléments enfants directs du premier élément indiqué.

```
premier_element element_cible {...}
```

- **Sélecteur descendant** : cible tous les éléments enfants directs du premier élément indiqué.

```
premier_element > element_cible {...}
```



CodePen

Les pseudo-classes

- **:link** : cible les liens à l'intérieur d'éléments n'ayant pas été visités
- **:visited** : cible les liens ayant été visités
- **:hover** : cible un élément au survol de la souris
- **:active** : cible un élément au clic de l'utilisateur ou lors de la tabulation
- **:focus** : cible un élément qui a reçu le focus via la souris ou la tabulation
- **:checked** : cible un élément ayant été coché (radio, checkbox ou select)
- **:disabled** : cible un élément désactivé
- **:first-child** : cible le premier élément
- **:first-of-type** : cible le premier élément du type spécifié
- **:last-child** : cible le dernier élément
- **:last-of-type** : cible le dernier élément du type spécifié
- **:not ()** : cible tous les éléments qui ne correspondent pas à l'argument

Remarque : pour décorer les liens sans effet de bord problématique, on placera la règle **:hover** après les règles **:link** et **:visited** mais avant la règle **:active**.
L'ordre total est **:link – :visited – :hover – :active**.

- **:nth-child(an+b)** : cible les éléments fils d'un élément dont les positions correspondent au motif *an+b*
- **:nth-last-child(an+b)** : cible les éléments fils d'un élément dont les positions correspondent au motif *an+b en démarrant de la fin*
- **:nth-last-of-type(an+b)** : cible les éléments fils d'un élément dont les positions correspondent au motif *an+b et dont le type correspond en démarrant de la fin*
- **:nth-of-type(an+b)** : cible les éléments fils d'un élément dont les positions correspondent au motif *an+b et dont le type correspond*
- **:only-child** : cible un élément ayant un parent qui ne possède que lui comme enfant
- **:only-of-type** : idem mais selon un type spécifié
- **:optional** : cible un élément de champ qui ne possède pas l'attribut "**required**"
- **:required** : cible un élément de champ qui possède l'attribut "**required**"
- **:valid** : cible un élément de champ valide
- **:invalid** : cible un élément de champ invalide
- **:in-range** : cible un élément ayant des limites de valeur dont la valeur respecte l'intervalle
- **:out-of-range** : cible un élément ayant des limites de valeur dont la valeur est en dehors de l'intervalle
- **:root** : cible la racine de l'arbre DOM
- **:target** : cible l'élément dont l'id correspond à l'ancre dans l'url



CSS4

- `:any-link()` : Combine les définitions `a:link` et `a:visited` en une seule.
- `:local-link()` : Ne prend en compte que les liens internes au site web.
- `:not(a,b)` : La négation est déjà présente dans CSS3 mais elle a été améliorée pour lui permettre de prendre en compte plusieurs éléments.
- `:has(a)` : Cette pseudo-classe s'applique aux sélecteurs ayant pour enfant l'élément
- `:matches(ul, ol, .list) > [hidden]` and `ul > [hidden], ol > [hidden], .list > [hidden]`
- Le sélecteur de parents
 - `ul li! a.active {`
 - `color: red;`
 - `}`

Les pseudo-classes temporelles

- `:past, :current et :future` : Ces pseudo-classes permettent de distinguer des éléments CSS dans le temps. Elles peuvent être utilisées, par exemple, pour l'affichage de sous-titres de vidéo. Ces sélecteurs peuvent avoir plusieurs arguments.

Les pseudo-éléments

- **::before** : cible un élément factice avant le contenu ciblé
- **::after** : cible un élément factice après le contenu ciblé
- **::first-letter** : cible la première lettre de la première ligne d'un bloc, si elle n'est pas précédée par un quelconque autre contenu
- **::first-line** : cible la première ligne d'un élément



Remarque : les syntaxes **:before** et **:after** fonctionnent également.

Remarque : les pseudo-éléments **::before** et **::after** doivent obligatoirement être accompagnés de la propriété **content:..."**

Les poids des sélecteurs

- `* {...} /* a=0 b=0 c=0` spécificité = 0 */
- `li {...} /* a=0 b=0 c=1` spécificité = 1 */
- `ul li {...} /* a=0 b=0 c=2` spécificité = 2 */
- `ul ol + li {...} /* a=0 b=0 c=3` spécificité = 3 */
- `h1 + *[REL=up] {...} /* a=0 b=1 c=1` spécificité = 11 */
- `ul ol li.red {...} /* a=0 b=1 c=3` spécificité = 13 */
- `li.red.level {...} /* a=0 b=2 c=1` spécificité = 21 */
- `#x34y {...} /* a=1 b=0 c=0` spécificité = 100 */

Remarque : ceci est une astuce pour récupérer le poids d'un sélecteur par rapport à un autre, **il ne s'agit en aucun cas d'une règle officielle.**

Les sélecteurs de formulaires

Comme pour toutes les balises il est possible de les sélectionner pour y ajouter des styles grâce à leurs noms et autres :

- **textarea** {...} : va styliser toutes les zones de texte larges
- **input.save** {...} : va styliser tous les inputs avec la classe save
- **select#chooseGender** {...} : va styliser tous les menus déroulants avec l'identifiant chooseGender
- **button[type='submit']** {...} : va styliser tous les boutons de soumission de formulaire
- **input:focus** {...} : va styliser tous les champs ayant reçu un focus (sur lesquels le curseur est actif).
- **input[type='radio']:checked** {...} : va styliser tous les boutons radio cochés (valable pour les checkbox et les radio)
- **input:disabled** {...} : va styliser tous les inputs désactivés.

Remarque : On peut au contraire cibler les éléments actifs avec la mention **:enable**

<https://github.com/samuel-gomez/formation-front-end/>



EXERCICE

CIBLER LES BONS ÉLÉMENTS ET APPLIQUER DU STYLE

git checkout -f exo-styles



Enoncé

- Tout le code de l'exercice est à mettre dans le fichier **exo.css**
- Chaque sélecteurs doit être précédé de la classe CSS correspondante à la partie de l'exercice

```
.part-1 p {  
    color: blue;  
}
```

Box model



Les display

La propriété CSS **display** est la plus importante pour contrôler la mise en page. Les **display** les plus utilisés sont :

- **display:block;** : les **div**, **p**, **form**, **header**, **footer** et **section**
- **display:inline;** : les **span**, **small**
- **display:inline-block;** : permet d'utiliser l'espace du contenu pour les élément de type **block**
- **display:none;** : permet de masquer un élément
- **display:table;** : rendu comme pour les tableaux

<https://codepen.io/gsam59/pen/qMBdKG>

Ne pas abuser des balises universelles

```
<span class="important"></span>
<div class="titre"></div>
<div class="section"></div>
```

Insérer les display inline dans les block

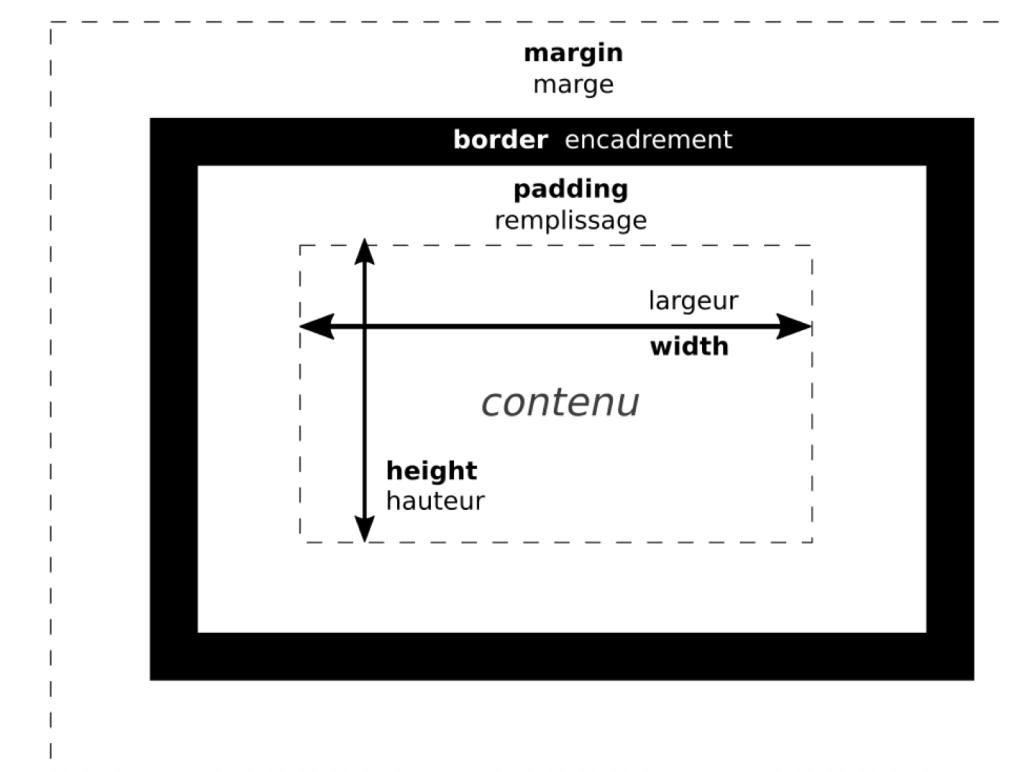


Les box-model

le terme ‘box-model’ est utilisé quand on parle de design et de mise en page.

Il comprend: les marges, les bordures, le remplissage et le contenu réel. L'image ci-dessous illustre le modèle de boîte:

```
* {  
    box-sizing : content-box;  
}
```



LES PROPRIÉTÉS

- **padding** : remplissage de la marge intérieure de la boite
- **border** : bordure de la boite
- **margin** : marge extérieure à la boite

Exemple:

```
margin: 10px 5px 8px 12px;  
margin : haut - droite - bas - gauche
```

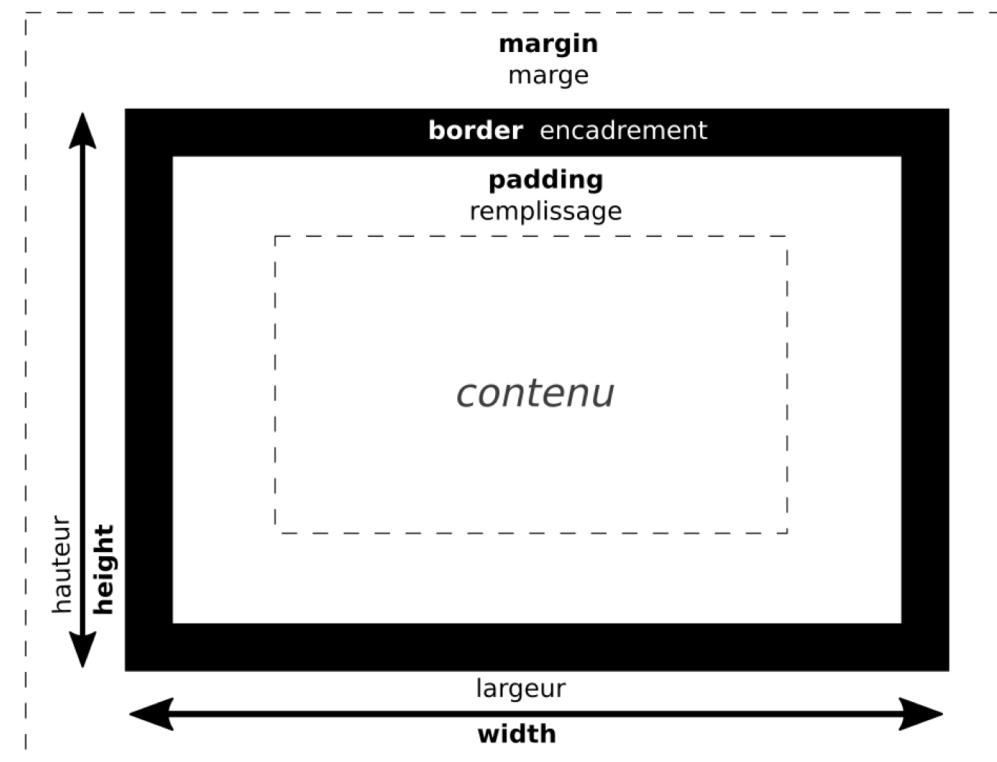
Remarque : Les marges ont un comportement spécifique nommé **margin-collapsing** (fusion des marges) : quand deux boîtes se touchent, la distance entre elles a la valeur de la plus grande des deux marges au contact et non leur somme.

```
* {  
    box-sizing : border-box;  
}
```

La valeur **border-box** indique au navigateur de prendre en compte la bordure et le remplissage dans la valeur définie pour la largeur et la hauteur.

```
* {  
    box-sizing : border-box;  
}
```

La valeur **border-box** indique au navigateur de prendre en compte la bordure et le remplissage dans la valeur définie pour la largeur et la hauteur.



CodePen



LIVE CODING DISPLAY



Conseils

Une bonne pratique est de ne pas définir une hauteur sur un élément html afin que celui-ci s'adapte automatiquement en fonction de son contenu

Favoriser les marges situées en bas

Préférer le min-height au height



Quelques liens pour aller plus loin



Developer Mozilla :

- <https://developer.mozilla.org/fr/docs/Web/CSS/display>

fonts



Les CSS fonts

INSERTION VIA HTML

```
<link href="https://fonts.googleapis.com/css?family=Roboto"  
rel="stylesheet">
```

INSERTION VIA CSS

```
@font-face { font-family: "font of all knowledge";  
src: url(fontofallknowledge.woff); }
```

```
body {  
    font-family: 'Roboto';  
}
```

Local fonts

Pour vérifier si une police est déjà présente sur l'ordinateur d'un utilisateur, vous pouvez également supprimer le téléchargement en remplaçant «url» par «local» dans la valeur.

```
@font-face {  
    font-family: "font of all knowledge";  
    src: local("font of all knowledge"),  
        local(fontofallknowledge),  
        url(fontofallknowledge.woff);  
    font-weight: 400;  
    font-style: normal;  
}
```

Légalement, vous ne pouvez pas utiliser n'importe quelle police, il faut tenir compte des termes du droit d'auteur et des conditions d'utilisation, sans parler des problèmes de compatibilité et d'optimisation.

position



Les positions

- **static** : comportement par défaut L'élément est alors positionné dans le flux avec sa position et vient s'imbriquer à la suite de l'élément précédent.
- **relative** : l'élément est positionné dans le flux normal du document puis décalé, par rapport à lui-même, selon les valeurs fournies par top, right, bottom et left.
- **absolute** : l'élément est retiré du flux et aucun espace n'est créé pour l'élément sur la page. Il est ensuite positionné par rapport à son parent le plus proche positionné de manière relative. La position finale de l'élément est déterminée par les valeurs de **top**, **right**, **bottom** et **left**.
- **fixed** : l'élément est retiré du flux normal et aucun espace n'est laissé pour l'élément. Il est positionné relativement au bloc englobant initial formé par la zone d'affichage (*viewport*),
- **sticky** : la position de l'élément est calculée en fonction du flux normal du document. Ensuite, il est décalé par rapport à son ancêtre de défilement le plus proche et par rapport à son bloc englobant selon les valeurs de **top**, **right**, **bottom** et **left**
Attention à la compatibilité.



<https://github.com/samuel-gomez/formation-front-end/>



EXERCICE

LES POSITIONS

git checkout -f exo-position

<https://codesandbox.io/s/z31844q6pl>



Home

Services

Portfolio

Blog

The team

Contact

Mon contenu de services

ridiculus mus. Duis ante orci, molestie vitae vehicula venenatis, tincidunt ac pede. Praesent vitae arcu tempor neque lacinia pretium. Maecenas aliquet accumsan leo. Aenean id metus id velit ullamcorper pulvinar. In convallis. Nam sed tellus id magna elementum tincidunt. Aliquam id dolor. Nunc tincidunt ante vitae massa. Morbi imperdiet, mauris ac auctor dictum, nisl ligula egestas nulla, et sollicitudin sem purus in lacus. Nam libero tempore, cum soluta nobis est eligendi optio cumque nihil impedit quo minus id quod maxime placeat facere possimus, omnis voluptas assumenda est, omnis dolor repellendus. Phasellus rhoncus. Morbi imperdiet, mauris ac auctor dictum, nisl ligula egestas nulla, et sollicitudin sem purus in lacus. Donec iaculis gravida nulla. Etiam quis quam.

Phasellus rhoncus. Aliquam erat volutpat. Fusce tellus odio, dapibus id fermentum quis, suscipit id erat. Nullam faucibus mi quis velit. Donec vitae arcu. Nam libero tempore, cum soluta nobis est eligendi optio cumque nihil impedit quo minus id quod maxime placeat facere possimus, omnis voluptas assumenda est, omnis dolor repellendus. Integer tempor. Fusce wisi. In enim a arcu imperdiet malesuada. Aliquam erat volutpat. Ut enim ad minima veniam, quis nostrum exercitationem ullam corporis suscipit laboriosam, nisi ut aliquid ex ea commodi consequatur? Duis sapien nunc, commodo et, interdum suscipit, sollicitudin et, dolor. Duis bibendum, lectus ut viverra rhoncus, dolor nunc faucibus libero, eget facilisis enim ipsum id lacus. Integer vulputate sem a nibh rutrum consequat. Duis pulvinar.

unités de mesure



Les unités de mesure

Les unités permettent d'exprimer les dimensions selon les cas d'usage :
em, px, %, rem.

	Recommandé	Usage occasionnel	Non recommandé
Écran	em, px, %	ex	pt, cm, mm, in, pc
Imprimante	em, cm, mm, in, pt, pc, %	px, ex	

Il faut donc veiller à utiliser les bonnes unités aussi selon les usages.

Unité de mesure EM

A quoi ça sert ?

Elle permet d'avoir des feuilles de style plus facilement adaptables d'un média à un autre

Le fonctionnement de la propriété CSS font-size avec les em est simple :

un élément HTML pour lequel nous avons une taille de texte de Xem aura un texte dont la hauteur représente X fois la hauteur du texte de son élément parent.

1.2em x 10px = 12px

1.2em x 16px = 19.2px

Attention aux cascades !

L'imbrication des éléments multiplient la valeur entre elles

```
.parent {  
    font-size: 16px;  
}  
  
.child {  
    font-size: 1.2em; // 19.2px  
}
```

<https://codepen.io/jeremychurch/pen/rCclh>

Unité de mesure REM

REM comme dans Root EM

Alors que *em* est relatif à la taille de la police de son parent direct ou le plus proche, *rem* est uniquement relatif à la balise html (racine) et son font-size.

Si une feuille de style est construite de manière modulaire, *rem* ne devrait pas être utilisé très souvent, mais il peut parfois être pratique.

Fini les cascades !

<https://codepen.io/jeremychurch/pen/dlyqw>

```
html {  
  font-size: 62.5%;  
}  
Body {  
  font-size: 1.8em; // 18px  
}  
  
.child {  
  font-size: 1.2rem; // 12px  
}
```

Unité de mesure VW VH

C'est unités permettent d'avoir des tailles variables selon la résolution d'écran

vh signifie "hauteur de la fenêtre",

vw pour viewport width,

vmin sélectionne la plus petite valeur entre vw et vh.

1VH représente 1% de la fenêtre courante

Les unités de fenêtres sont dynamiques plutôt que statiques, donc si vous redimensionnez la fenêtre du navigateur, la valeur calculée des unités changera également.





LIVE CODING

UNITÉ DE MESURE





QUIZZ

UNITÉS DE MESURE



Je souhaite utiliser les em pour mes tailles de police mais je n'ai défini aucune taille de référence au préalable.

Qu'elle sera la taille par défaut de 1em ?

- o Impossible de savoir il s'agit d'une unité relative
- o 14px
- o La taille de police par défaut de mon navigateur, soit bien souvent 16px

**J'ai un article de 400px de large sur smartphone et
1000px sur ordinateur.**

A l'intérieur se trouve une section de 25% de largeur.

**Qu'elle sera la largeur de ma section en px sur smartphones et
ordinateurs ?**

- o 90px et 220px
- o 100px et 250px
- o 40px et 200px

Quels sont les valeurs de child et child-child en pixel ?

- 25px et 50px
- 25px et 75px
- 25px et 20px

```
.parent {  
    font-size: 10px;  
}  
  
.child {  
    font-size: 2.5em;  
}  
  
.child-child {  
    font-size: 2rem;  
}
```

Quelques liens pour aller plus loin



Images :

- <https://www.alsacreations.com/article/lire/1621-responsive-images-srcset.html>

Liens :

- <https://www.alsacreations.com/article/lire/1400-attribut-rel-relations.html>

Formulaires :

- <https://developer.mozilla.org/fr/docs/Web/CSS/outline>
- <https://developer.mozilla.org/fr/docs/Web/CSS/-moz-appearance>
- <https://developer.mozilla.org/fr/docs/Web/CSS/resize>

Agencement des éléments



Centrer le contenu



Centrage horizontal

Cas 1 : un parent contenant des éléments de type inline ou inline-*.

```
.parent {  
    text-align: center;  
}
```



[CodePen](#)

Cas 2 : un parent contenant un élément de type block.

```
div {  
    margin: 0 auto;  
    width: 100px;  
}
```



[CodePen](#)

Cas 3 : un parent contenant plusieurs éléments de type block.

```
.parent {text-align:center}  
div {  
    display: inline-block;  
    width: 100px;  
}
```



[CodePen](#)

Centrage vertical

Cas 1a : un parent contenant un éléments de type **inline** ou **inline-***.

- Avec **line-height**:



Remarque : Ne fonctionne que si le parent possède une hauteur définie.

Cas 1b : un parent contenant plusieurs lignes d'éléments de type **inline** ou **inline-***.

- Avec **table** :



- Avec **flexbox** :



- Avec **vertical-align** :



Cas 2a : un parent contenant un éléments block dont la hauteur est connue.

- Avec **position** :



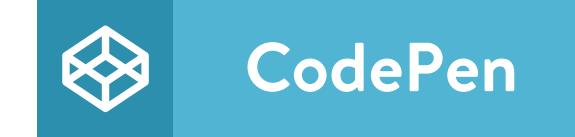
Cas 2b : un parent contenant un élément block dont la hauteur est inconnue.

- Avec **position** et **translate** :



Cas 2c : un parent contenant un élément block avec flexbox.

- Avec **flexbox**



Centrage horizontal et vertical

Cas 1a : un parent avec une largeur et hauteur fixes.

- Avec **margin** :



Cas 1b : un élément avec une largeur et une hauteur inconnues.

- Avec **transform** :



Cas 1c : c'est « magique ».

- Avec **flexbox** :



Les flottants



Float

Même si l'utilisation des float CSS est de plus en plus rare, il reste encore beaucoup de sites ou applications qui l'utilisent encore.

```
.foo {  
    float: left | right | inherit | none;  
}
```

1. Un élément flotté est “invisible” pour son parent
2. Un élément flotté à gauche se positionnera toujours le plus en haut à gauche dans son élément parent
3. Les éléments définis avant repoussent l’élément flotté vers le bas
4. L’ordre de déclaration des éléments flottés compte



Clearfix

Lorsqu'on utilise les **float CSS** pour agencer les éléments d'une page, on se trouve rapidement confronté à la problématique principale des flottants : la hauteur du parent ne s'adapte pas à son contenu puisqu'il n'est pas visible par celui-ci.

Il faut donc annuler ce défaut, une astuce a depuis été trouvée pour y remédier :

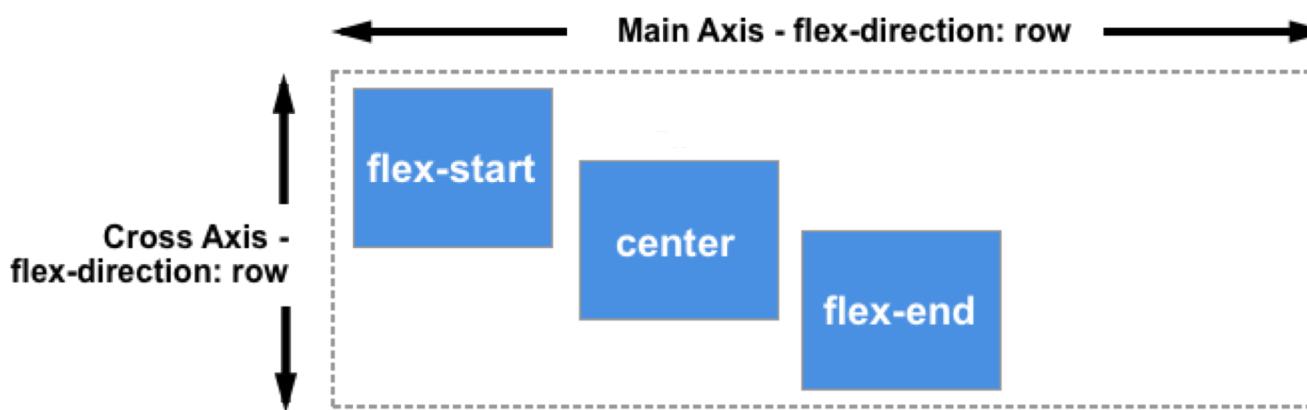
```
.clearfix::after {  
    content:"";  
    display:table;  
    clear:both;  
}
```

Flexbox



l'axe principal

- **flex-direction**
 - **row** – par défaut
 - **column**
 - **row-reverse**
 - **column-reverse**

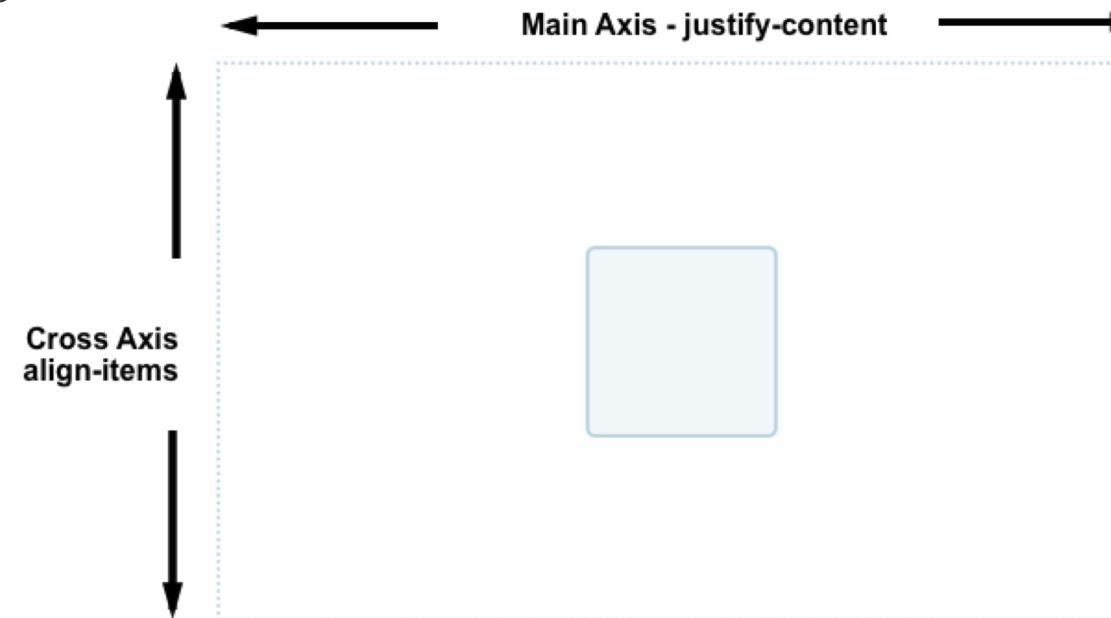


Le retour à la ligne se fait avec :

Flex-wrap : wrap | nowrap

L'alignement

- **justify-content** : contrôle l'alignement de tous les objet(s) sur l'axe principal
- **align-items** : étend/aligner le ou les objet(s) sur l'axe secondaire.
stretch par défaut,
- **align-self** : contrôle l'alignement d'un objet flexible donné le long de l'axe secondaire.
- **align-content** : contrôle l'espace entre les lignes flexibles sur l'axe secondaire



L'alignement

- `justify-content`



- `align-items`
- `align-self`



- `align-content`



Flexibilité

- **flex-shrink** : indique le facteur de rétrécissement d'un élément flexible.
- **flex-grow** : indique le facteur de grossissement d'un élément flexible. Elle indique la quantité d'espace que l'élément devrait consommer dans un conteneur flexible relativement à la taille des autres éléments du même conteneur.
- **flex-basis** : détermine la base de flexibilité utilisée comme taille initiale principale pour un élément flexible. Cette propriété détermine la taille de la boîte de contenu sauf si une autre boîte est visée par **box-sizing**.
- **Order:** permet de changer l'ordre des items



Les propriétés raccourcies

- **flex-flow** : **flex-direction** | **flex-wrap**;
- **flex** : **flex-shrink** | **flex-grow** | **flex-basis**;

<https://github.com/samuel-gomez/formation-front-end/>



EXERCICE

FLEXBOX

<https://flexboxfroggy.com/#fr>



CSS grid



Css grid

Le principe

Piloté la disposition de la grille (définir des colonnes et des lignes) depuis un élément afin de définir des mises en page

```
grid {  
  display: grid;  
  grid-template-columns: 20% 20% auto;  
  grid-template-rows: 100px 200px 400px;  
}
```

<p>Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Curabitur sagittis hendrerit ante. Morbi imperdiet, mauris ac auctor dictum, nisl ligula egestas nulla, et sollicitudin sem purus in lacus. Praesent in mauris eu tortor porttito</p>	<p>Fusce dui leo, imperdiet in, aliquam sit amet, liquam quaerat voluptatem. Fusce tellus odio, dapibus id fermentum quis.</p>	<p>Phasellus rhoncus. Aliquam erat volutpat. Fuscea nibh rutrum consequat. Duis pulvinar.</p>
<p>Integer pellentesque quam vel velit. Maecenas lMauris dictum facilisis augue. Maecenas aliquet accumsan leo. Phasellus faucibus molestie nisl. Integer pellentesque quam vel velit. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Etiam quis quam. Quis autem vel eum iure reprehenderit qui in ea voluptate velit esse quam nihil molestiae consequatur, vel illum qui dolorem eum fugiat quo voluptas nulla pariatur? Mauris tincidunt sem sed arcu. Nullam feugiat, turpis at pulvinar vulputate, erat libero tristique tellus, nec bibendum odio risus sit amet ante. Vestibulum fermentum tortor id mi.</p>	<p>Aenean vel massa quis mauris vehicula lacinia. Donec vitae arcu. sapien wisi sed libero. Aliquam ornare wisi eu metus.</p>	<p>Aenean vel massa quis mauris vehicula lacinia. Donec vitae arcu. sapien wisi sed libero. Aliquam ornare wisi eu metus.</p>

Css grid

Espacement entre ligne et colonne

```
.grid {  
    display: grid;  
    grid-template-columns: 20% 20% auto;  
    grid-column-gap: 20px;  
}
```

```
.grid {  
    grid-gap: 20px;  
}
```

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Curabitur sagittis hendrerit ante. Morbi imperdiet, mauris ac auctor dictum, nisl ligula egestas nulla, et sollicitudin sem purus in lacus. Praesent in mauris eu tortor porttito

Integer pellentesque quam vel velit. Maecenas I Mauris dictum facilisis augue. Maecenas aliquet accumsan leo. Phasellus faucibus molestie nisl. Integer pellentesque quam vel velit. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Etiam quis quam. Quis autem vel eum iure reprehenderit qui in ea voluptate velit esse quam nihil molestiae consequatur, vel illum qui dolorem eum fugiat quo voluptas nulla pariatur? Mauris tincidunt sem sed arcu. Nullam feugiat, turpis at pulvinar vulputate, erat libero tristique tellus, nec bibendum odio risus sit amet ante. Vestibulum fermentum tortor id mi.

Fusce dui leo, imperdiet in, aliquam sit amet, liquam querat voluptatem. Fusce tellus odio, dapibus id fermentum quis.

Aenean vel massa quis mauris vehicula lacinia. Donec vitae arcu. sapien wisi sed libero. Aliquam ornare wisi eu metus.

Phasellus rhoncus. Aliquam erat volutpat. Fuscea nibh rutrum consequat. Duis pulvinar.

Aenean vel massa quis mauris vehicula lacinia. Donec vitae arcu. sapien wisi sed libero. Aliquam ornare wisi eu metus.

Unité fractionnée

Possible de définir des zones plutôt que des tailles

```
.grid {  
    display: grid;  
    grid-template-columns: 1fr 1fr 1fr;  
    grid-column-gap: 20px;  
}
```

grid-template-columns: 1fr 1fr 1fr;
grid-column-gap: 20px;

Unité fractionnée

La première colonne occupe 1/3 de la largeur totale, la deuxième 1/3 et la troisième 1/3.

Content :

Fusce dui leo, imperdiet in, aliquam sit amet, liquam quaerat voluptatem. Fusce tellus odio, dapibus id fermentum quis.

Aenean vel massa quis mauris vehicula lacinia. Donec vitae arcu. sapien wisi sed libero. Aliquam ornare wisi eu metus.

Phasellus rhoncus. Aliquam erat volutpat. Fuscea nibh rutrum consequat. Duis pulvinar.

Aenean vel massa quis mauris vehicula lacinia. Donec vitae arcu. sapien wisi sed libero. Aliquam ornare wisi eu metus.

Fonction raccourcie

Vous pouvez répéter le nombre de colonne facilement

```
.grid {  
    display: grid;  
    grid-template-columns: repeat(4, 1fr);  
    grid-template-gap: 20px;  
}
```

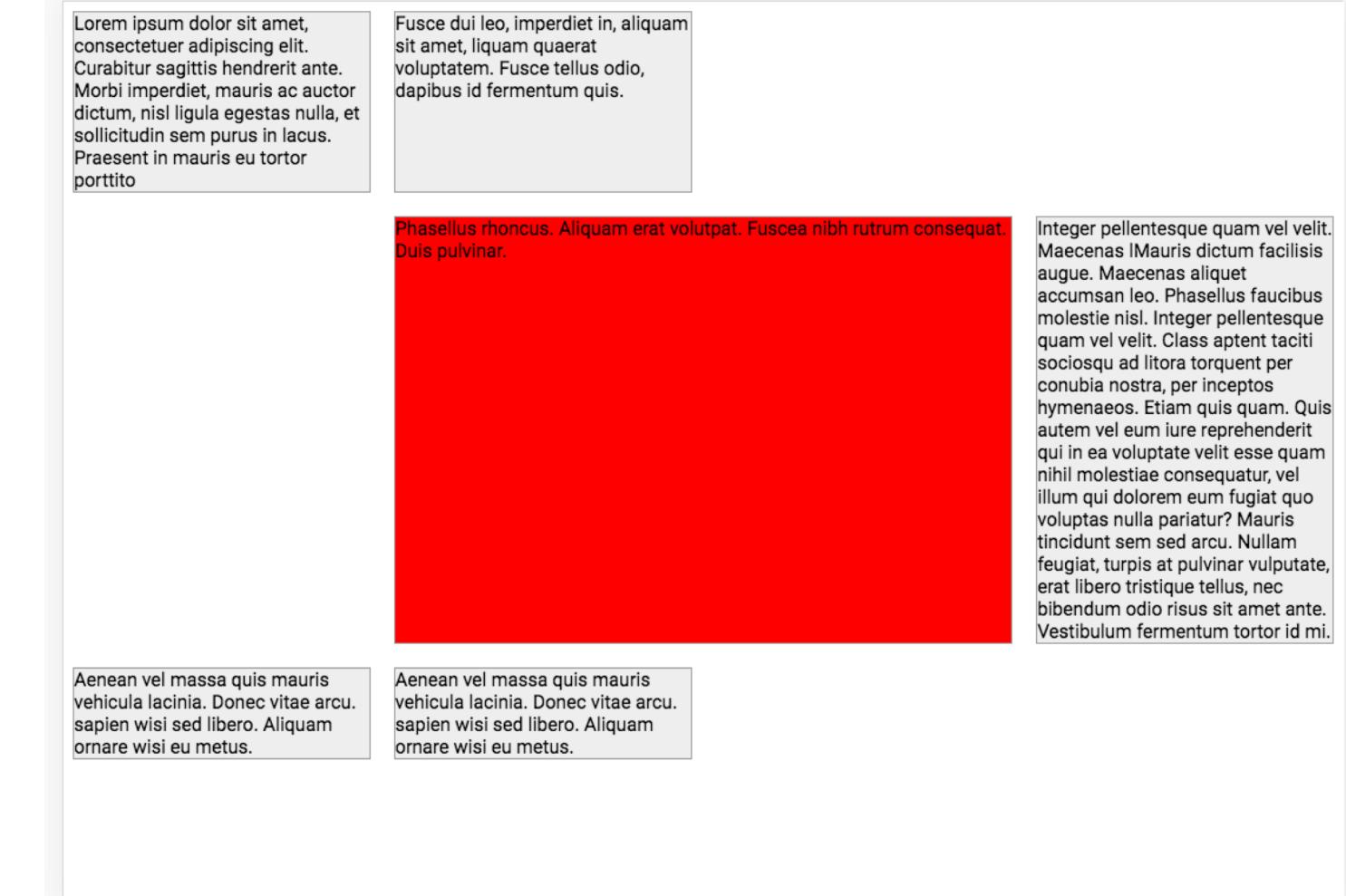
Positionnement

Il est possible de positionner un élément dans la grille, ci-dessous :
Child commence à la **2^e** colonne et s'étend jusque la **4^e** colonne

```
.grid {  
  display: grid;  
  grid-template-columns: repeat(4, 1fr);  
}
```

```
.child {  
  grid-column-start: 2;  
  grid-column-end: span 2;  
}
```

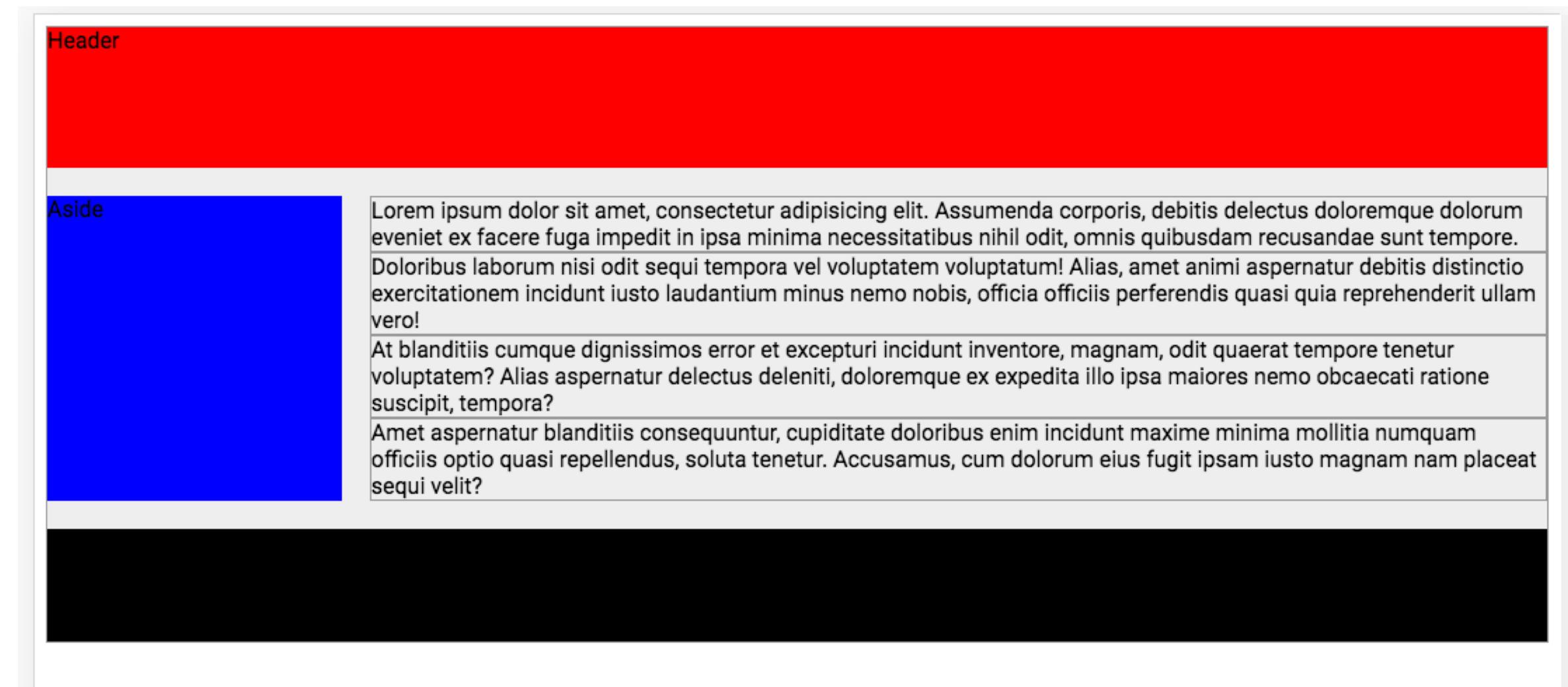
```
.child {  
  grid-column: 2 / span 2;  
}
```



Zones

Vous pouvez définir des zones pour structurer votre grille

```
grid {  
  display: grid;  
  grid-template-columns: 2fr 8fr;  
  grid-template-rows: 100px 1fr 80px;  
  grid-gap: 20px;  
  grid-template-areas:  
    "header header"  
    "aside body"  
    "footer footer"  
}  
  
header {  
  background: red;  
  grid-area: header;  
}  
  
aside {  
  background: blue;  
  grid-area: aside;  
}
```



Alignment

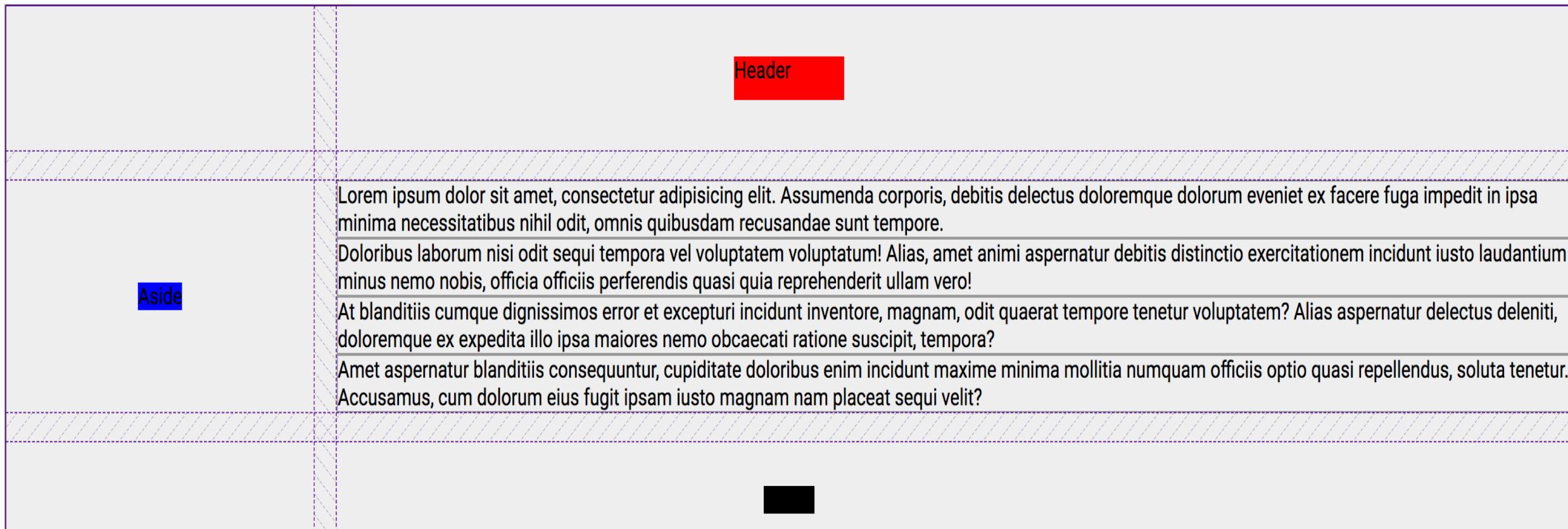
Grid css comporte les mêmes propriété d'alignment que le flex

Il est possible d'aligner les éléments depuis le parent

```
grid {  
  display: grid;  
  grid-template-columns: 2fr 8fr;  
  grid-template-rows: 100px 1fr 80px;  
  grid-gap: 20px;  
  grid-template-areas:  
    "header header"  
    "aside body"  
    "footer footer";  
  justify-items: center;  
  align-items: center;  
}  
  
header {  
  width: 100px;  
  height: 30px;  
  background: red;  
  grid-area: header;  
}  
  
aside {  
  background: blue;  
  grid-area: aside;  
}
```

Justify-items gère l'axe X

Align-items gère l'axe Y



<https://codepen.io/sdailly/pen/yQyYEp>

Alignment - self

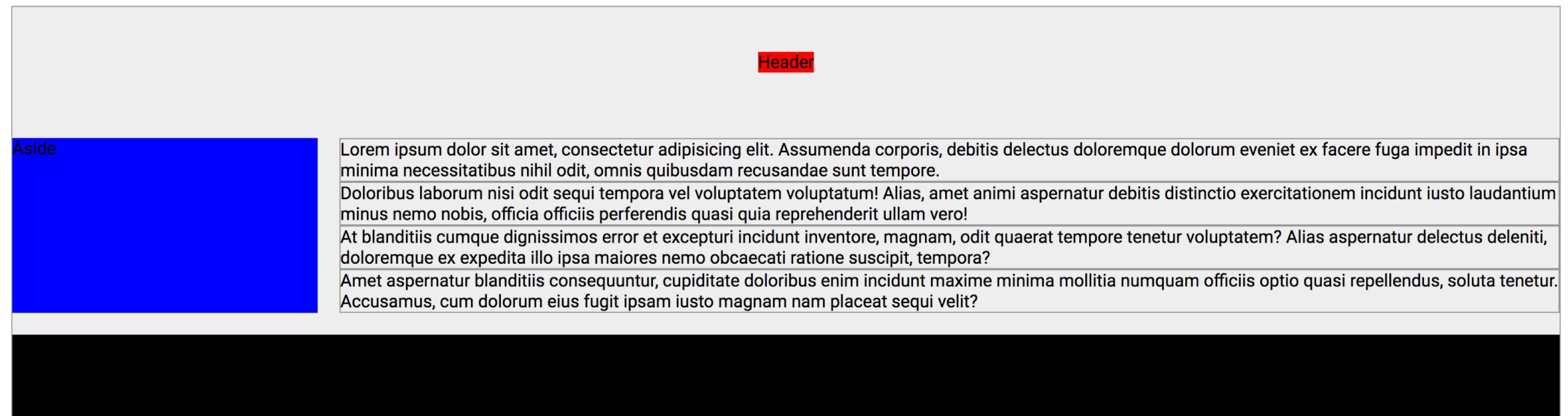
Grid css comporte les mêmes propriété d'alignement que le flex

Il est possible d'aligner un élément spécifique

```
grid {  
  display: grid;  
  grid-template-columns: 2fr 8fr;  
  grid-template-rows: 100px 1fr 80px;  
  grid-gap: 20px;  
  grid-template-areas:  
    "header header"  
    "aside body"  
    "footer footer";}  
  
header {  
  width: 100px;  
  height: 30px;  
  background: red;  
  grid-area: header;  
  justify-self: center;  
  align-self: center;  
}  
  
aside {  
  background: blue;  
  grid-area: aside;  
}
```

Justify-self gère l'axe X de l'élément

Align-self gère l'axe Y de l'élément



<https://codepen.io/sdailly/pen/qQEOKv>

<https://github.com/samuel-gomez/formation-front-end/>



EXERCICE

CSS GRID

<http://cssgridgarden.com/#fr>

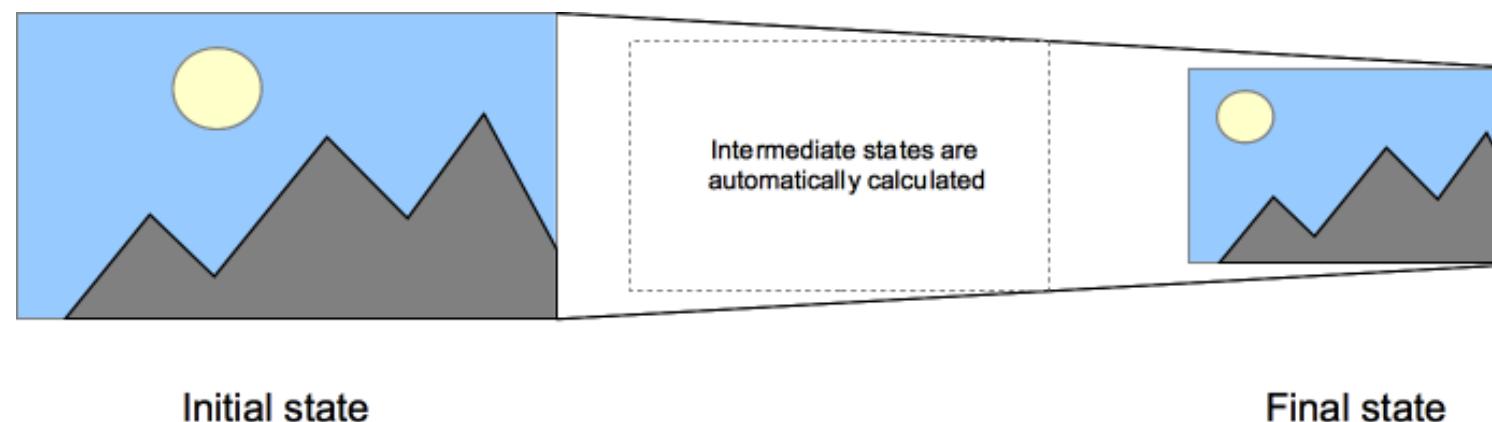


Les transitions



Les transitions

Les **transitions CSS** permettent de contrôler la vitesse d'animation lorsque les propriétés CSS sont modifiées. Plutôt que le changement soit immédiat, on peut l'étaler sur une certaine période



Les transitions CSS vous permettent de choisir :

- les propriétés à animer
- le délai du démarrage de l'animation
- la durée de l'animation
- la façon dont la transition s'exécutera

Les transitions

```
div {  
  transition-property: opacity, left, top, height;  
  transition-delay: 3s;  
  transition-duration: 3s, 5s, 3s, 5s;  
  transition-timing-function: ease | linear  
}
```



CodePen

Les transitions CSS vous permettent de choisir :

- les propriétés à animer
- le délai du démarrage de l'animation
- la durée de l'animation
- la façon dont la transition s'exécutera

Les transformations



Les transformations 2D

Les transformations 2D s'opèrent sur deux axes : X et Y

- **translate** (ou **translateX/translateY**) : permet de changer la position de l'élément
- **rotate** (ou **rotateX/rotateY**) : permet d'effectuer une rotation de l'élément
- **scale** (ou **scaleX/scaleY**) : permet de modifier la taille de l'élément
- **skew** (ou **skewX/skewY**) : permet d'effectuer une distorsion de l'élément
- **matrix** : méga propriété qui combine toutes les propriétés de transformation



Les transformations 3D

Les transformations 3D s'opèrent sur trois axes : X, Y et Z. Les propriétés CSS restent identiques à celles de la 2D.

- `translate3d` / `translateX` / `translateY` / `translateZ`
- `rotate3D` / `rotateX` / `rotateY` / `rotateZ`
- `scale3D` / `scaleX` / `scaleY` / `scaleZ`



Responsive



MOBILE



TABLET

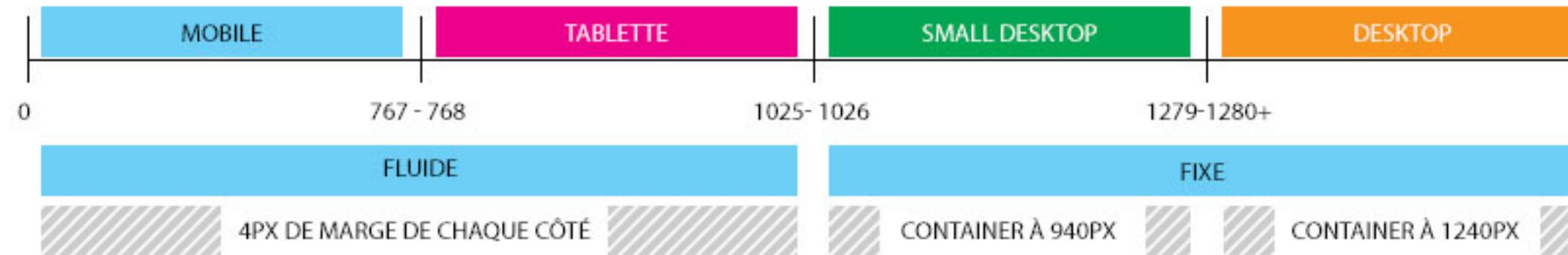


DESKTOP



Résolutions & breakpoints

Pour une navigation facilitée il est essentiel d'opter pour un affichage dédié à chaque device. On définit alors des points de rupture (breakpoints) et on adaptera la mise en page à chaque cassure.



On définit alors **3 breakpoints** et **4 définitions de devices** grâce à des variables CSS et JS.



Viewport

La balise meta va imposer la largeur de viewport et le niveau de zoom au navigateur

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

HTML

Valeurs utilisables :

- **Width**
largeur de fenêtre viewport
- **Height**
Hauteur de fenêtre viewport
- **Initial-scale**
niveau de zoom initial
- **Minimum-scale**
niveau de zoom minimum
- **Maximum-scale**
niveau de zoom maximum
- **User-scalable**
Permettre à l'utilisateur de zoomer ou non

Médias Queries

une media query est une expression dont la valeur est toujours vraie ou fausse.

Type de média

Un type de média définit une catégorie générale d'appareil.

Le type de média est optionnel dans une requête média, sauf si celle-ci utilise les opérateurs logiques *not* ou *only*.

Par défaut, le type de média utilisé est *all*.

all

Correspond pour tous les appareils.

print

Correspond aux matériaux paginés et aux documents consultés en aperçu avant impression.

screen

Correspond aux appareils dotés d'un écran.

speech

Correspond aux outils de synthèse vocale.

Les opérateurs logiques peuvent être :

and : "et"

only : "uniquement"

not : "non".

Médias Queries

Min-width

Largeur minimale du viewport

Max-width

Largeur maximale du viewport

orientation

L'orientation la zone d'affichage

Aspect-ratio

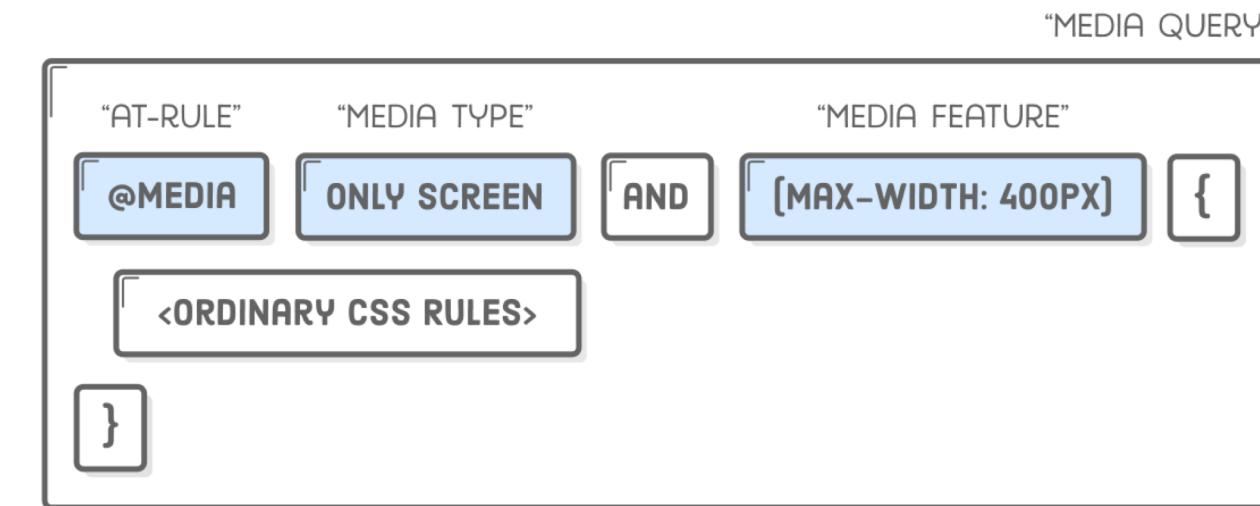
Le rapport largeur/hauteur de la zone d'affichage

Hover – css4

La capacité du mécanisme de saisie principal à survoler les éléments

Pointer – css4

La présence d'un appareil de pointage comme mécanisme de saisie principal et sa précision



```
@media (min-width: 30em) and (orientation: landscape) { ... }
```

```
@media (min-height: 30em), screen and (orientation: portrait) { ... }
```

Mobile first

Avant on concevait la version desktop avant de la décliner sur les différents supports. Aujourd’hui on conseille de commencer par la version mobile afin d’avoir un code plus simple et éviter d’écraser certaines règles écrites pour le desktop.

DESKTOP FIRST

```
.main {  
    background: #ffffff;  
    float: left;  
    width: 75%;  
}  
.sidebar {  
    background: #ff0000;  
    float: left;  
    width: 25%;  
}  
@media only screen and  
(max-width: 640px) {  
    .main, .sidebar {  
        float: none;  
        width: 100%;  
    }  
}
```

MOBILE FIRST

```
.main {  
    background: #ffffff;  
}  
.sidebar {  
    background: #ff0000;  
}  
@media only screen and  
(min-width: 640px) {  
    .main {  
        width: 75%;  
        float:left;  
    }  
    .sidebar{  
        width: 25%;  
        float: left;  
    }  
}
```

Adaptive VS. Responsive

La différence entre l'adaptif et le responsif est la fluidité.

- Une approche **responsive** signifie proposer une expérience optimale à travers tous les formats de devices.
- Une approche **adaptive** privilégiera des zones de rupture franches et un container réadapté en fonction du device sélectionné.

Aujourd'hui on a tendance à combiner les deux approches pour plus de facilité.



<https://github.com/samuel-gomez/formation-front-end/>



EXERCICE

EXO DEL ARTE

Git checkout -f exo-delarte



TOUS NOS RESTAURANTS



CARTE & MENUS



LA QUALITÉ



RECRUTEMENT



FRANCHISE



Commander en ligne



Mon restaurant

Choisir mon restaurant préféré



Mon compte

Me Connecter Ou Créer Un Compte



PROGRAMME DE FIDÉLITÉ

Découvrir ou Activer une carte



CSS / **Les sélecteurs**

The screenshot shows the top navigation bar of the Del'Arte website. On the left, there are links for "Choisir mon restaurant préféré" and "Modifier". In the center is the Del'Arte logo, which includes the text "RISTORANTE DEL'ARTE" and a stylized globe graphic. On the right, there are links for "Mon compte" (with a user icon) and "PROGRAMME DE FIDÉLITÉ" (with a green square icon). Below the navigation bar, a horizontal menu bar contains links for "TOUS NOS RESTAURANTS", "CARTE & MENUS", "LA QUALITÉ", "RECRUTEMENT", "FRANCHISE", and "Commander en ligne" (with a wavy line icon).

CSS4

RANGE SYNTAX

Before

```
@media (min-width: 20em) and  
        (max-width: 40em)  
{ ... }
```

After

```
@media (20em <= width <= 40em)  
{ ... }
```

CSS4

BOOLEAN LOGIC SYNTAX

Before

```
@media (min-width: 20em), (min-height: 40em) {  
  @media not all and (pointer: none) {  
    ...  
  }  
}
```

After

```
@media ((min-width: 20em) or (min-height: 40em))  
and  
(not (pointer: none)) {  
  ...
```

cela devrait tout faire beaucoup
plus lisible

CSS4

MEDIA FEATURES TO THE RESCUE

	pointer	hover	update	overflow-block	scripting
laptop	fine	hover	fast	scroll	enabled
phone	coarse	none	fast	scroll	enabled
Wii	coarse	hover	fast	scroll	enabled
TV	none	none	fast	scroll	enabled
print	none	none	none	paged	initial-only

CSS4

MEDIA FEATURES	
Level 3	width <length>
	height <length>
	aspect-ratio <ratio>
	orientation portrait landscape
	resolution <resolution> infinite
	scan interlace progressive
	grid <mq-boolean>
	color <integer>
	color-index <integer>
	monochrome <integer>
Level 5	color-gamut srgb p3 rec2020
	update none slow fast
	overflow-block none scroll optional-paged paged
	overflow-inline none scroll
	pointer none coarse fine
	hover none hover
	any-pointer none coarse fine
	any-hover none hover
	light-level dim normal washed
	inverted-colors none inverted
	prefers-reduced-motion no-preference reduce
	scripting none initial-only enabled

Quelques liens pour aller plus loin



Media Queries :

- https://developer.mozilla.org/fr/docs/Web/CSS/Requ%C3%AAtes_m%C3%A9dia/Utiliser_les_Media_queries

Responsive :

- <https://www.utilab.com/fr/responsive-ou-adaptive-design/>
- <https://www.alsacreations.com/article/lire/1615-cest-quoi-le-responsive-web-design.html>
- <https://usabilitygeek.com/adaptive-vs-responsive-design/>

S'exercer :

- <https://apprendre-html.3wa.fr/projets/>
- <https://www.sassmeister.com/>
- <http://www.flexboxdefense.com/>
- <https://flukeout.github.io/>

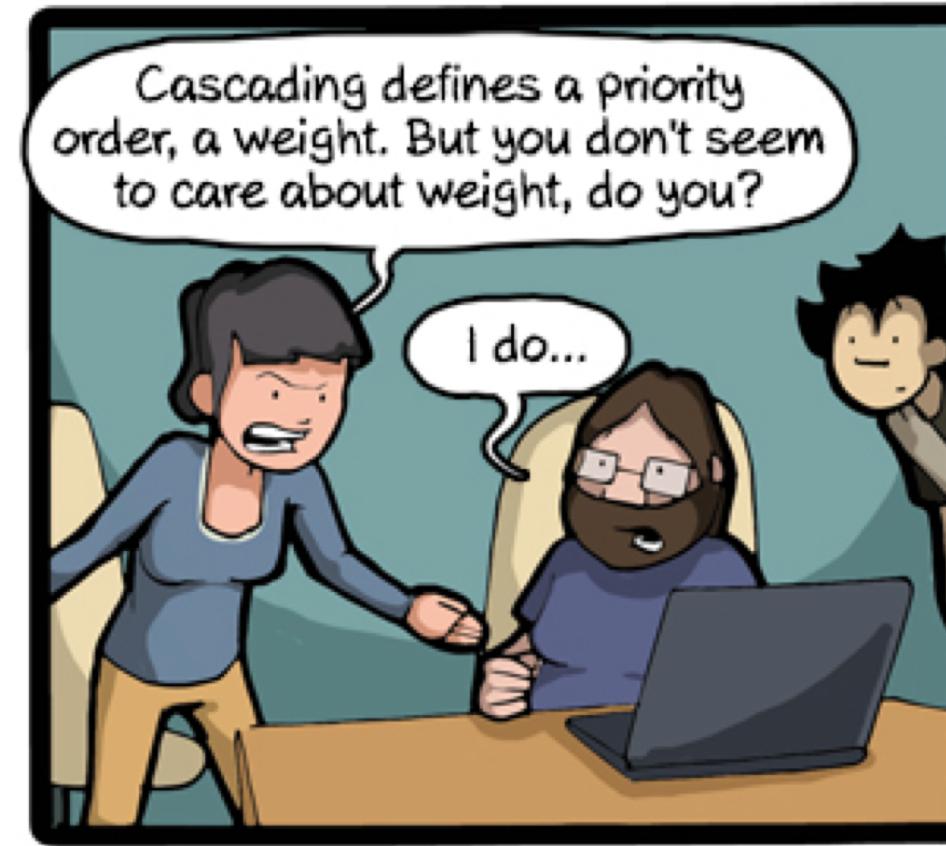
Méthodologie



BEM



Améliorer votre code avec BEM



Définition

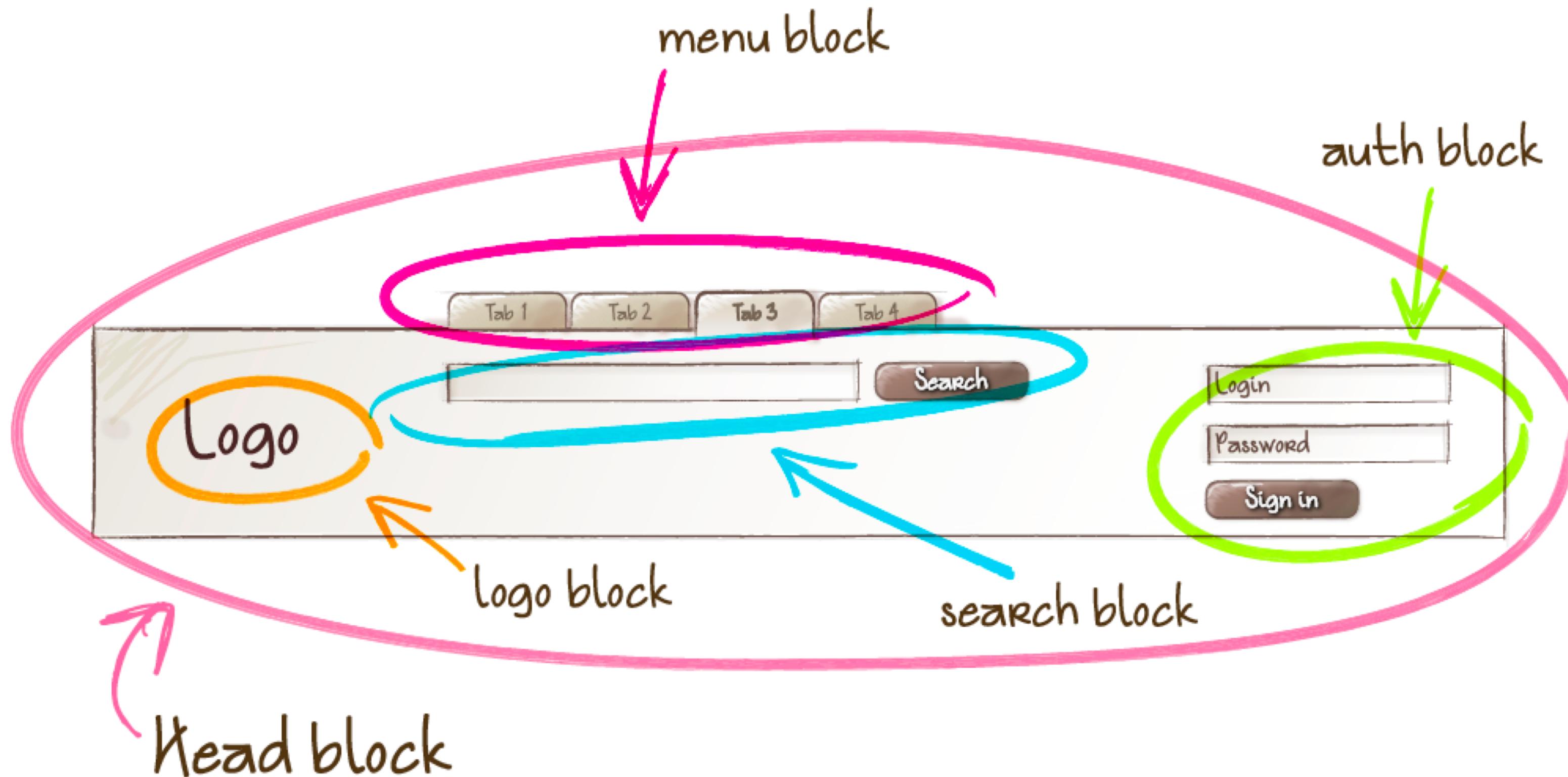
C'est plutôt une façon de nommer les classes en CSS, une convention en d'autres termes voire une méthodologie.

BEM apporte une solution à un problème majeur en informatique : le nommage.

BEM est le sigle pour *Block, Element, Modifier*

Son intérêt ?

- Prépare et structure votre code
- Isoler le style des composants réutilisables
- Maintenir un css propre
- Approche/Convention commune

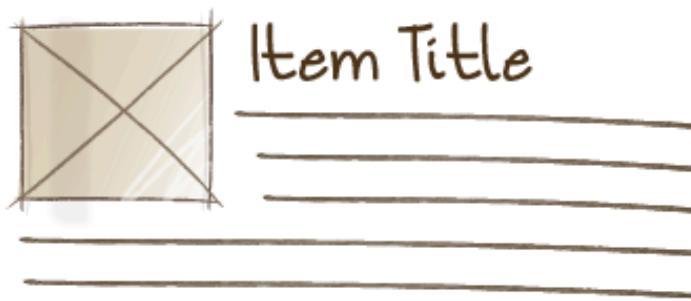


Block Element Modifier

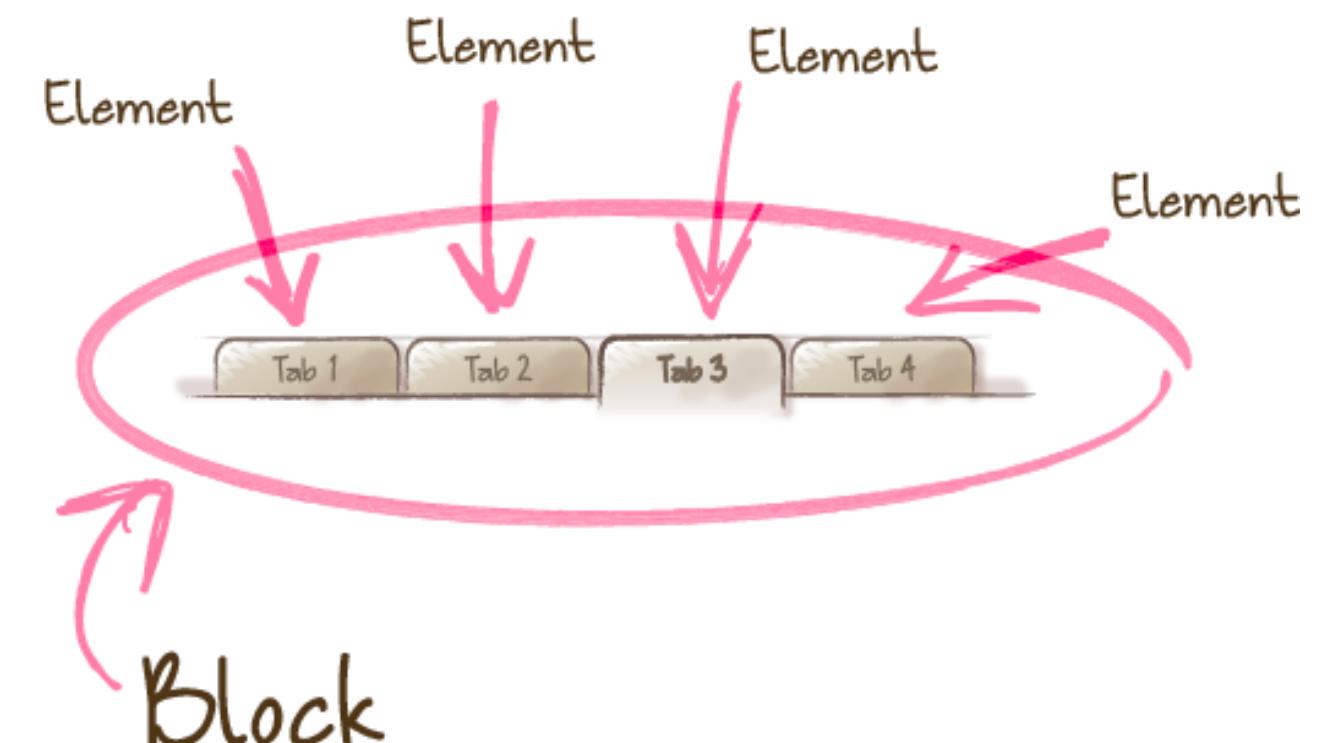
Composant de page indépendant du point de vue logique et fonctionnel, équivalent à un composant de Web Components.

Block est le conteneur, l'ensemble, ce qui entoure le tout

Les blocs étant indépendants, ils peuvent être réutilisés

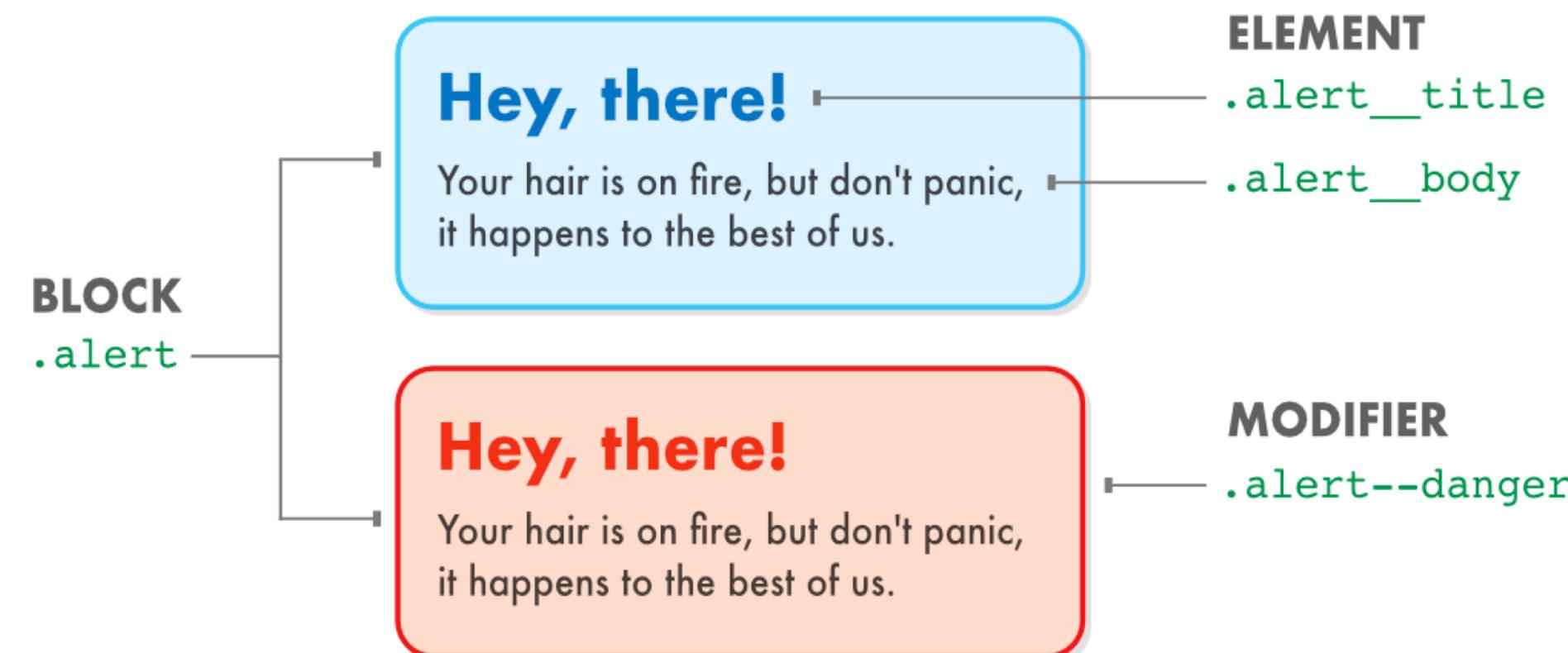


Element est une partie constructive de notre Block et qui ne peut être utilisé dans un autre contexte



Block Element Modifier

Le **modifier** définit l'apparence et le comportement d'un bloc ou d'un élément .



Convention de nommage

La même entité BEM a toujours le même nom dans toutes les technologies (CSS, JavaScript et HTML).

Règle de nommage

block-name__elem-name_modifier

block-name__elem-name_modifier

- Les noms sont écrits en lettres latines minuscules.
- Les mots sont séparés par un trait d'union (-).
- Le nom de l'élément est séparé du nom du bloc par un double trait de soulignement (__).
- Le nom du modificateur est séparé du nom du bloc ou de l'élément par un soulignement simple (_).
- La valeur du modificateur est séparée du nom du modificateur par un simple trait de soulignement (_).
- Pour les modificateurs booléens, la valeur n'est pas incluse dans le nom.

Variantes

Il existe plusieurs variantes de la méthodologie BEM, avec leurs propres règles de nommage

.block__element_modifier

.block__element--modifier

.Block-element--modifier

Le plus import est de se fixer une convention de nommage identique dans toute notre application

Préfix

Afin d'éviter tout conflits avec d'autres framework utilisant la convention BEM ou pour définir une action javascript.

Il peut être intéressant d'utiliser un préfix sur toutes les classes

```
.block__element_modifier
```

```
.prefix-block__element--modifier
```

```
.af-Header
```

```
.af-Carousel
```

```
<div class=af-Carousel js-af-Carousel></div>
```

Kebab Case

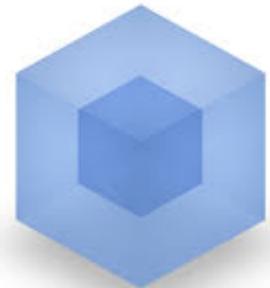
If the name of your component is too long

Bad 😞	Good 😊
.af-headerTop	.af-header-top
.af-tmpMetier	.af-tmp-metier
.af-subActivity	.af-sub-activity

Automatisation des tâches



Task runner



GULP / WEBPACK

- Automatisation des tâches récurrentes
- Compilation des fichiers (inclusions, transformations, minifications, autoprefixer)
- Détection des erreurs de compilation/ build des fichiers (avec watcher)
- Déploiement d'un dossier **dist** regroupant les fichiers destinés au serveur



CSS3 / BEM / CONVENTIONS



SASS / LESS

- Variables, modulaire, nested, mixins, fonctions



POST-CSS

- Regroupement des **media-queries** dans les fichiers finaux : **css-mqpacker**
- Variable des chemins css : **postcss-assets**
- Optimisation (suppression des commentaires/ doublons/ selecteurs vides, props non utilisées, autoprefixer, espaces...etc) : **cssnano-postcss**

<https://github.com/samuel-gomez/formation-front-end/>



EXERCICE

EXO PRINTEMPS

Git checkout -f exo-printemps2



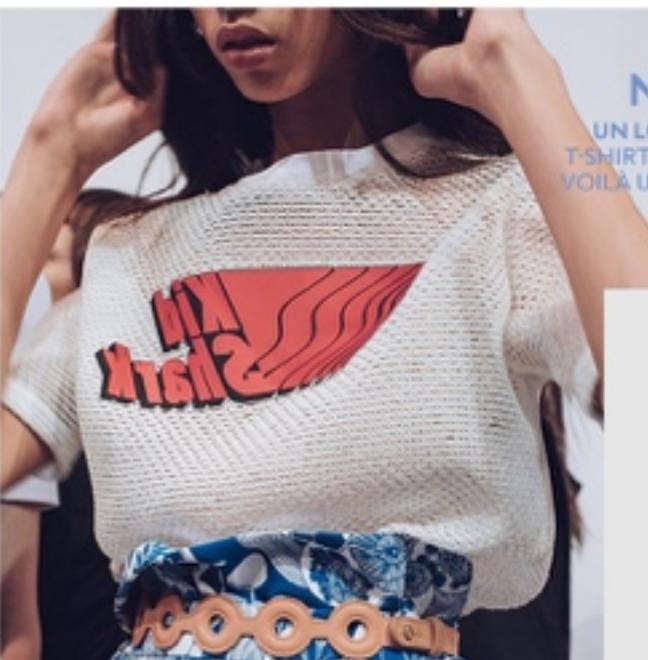


JULIEN DOSSENA, DIRECTEUR ARTISTIQUE DE PACO RABANNE, A REMIS LA CÉLÈBRE MAISON FRANÇAISE EN ORBITE. ON ADORE SON APPROCHE RADICALE ET NOVATRICE, PLEINE D'ÉNERGIE ET DE DÉTERMINATION, UNE VISION VERY PRINTEMPS !
(RE)DÉCOUVERTE D'UNE GRIFFE MYTHIQUE.

PRINTEMPS AIME...

SA LIBERTÉ DE MOUVEMENT ET D'ATTITUDE

Passion, générosité et enthousiasme – le tout teinté d'une gentillesse palpable : il y a tout cela dans la voix de Julien Dossena. C'est réjouissant et surtout très contagieux. Au cours de notre rencontre, le créateur prononcera de nombreuses fois le mot "accessible", employé non pour signifier un côté populaire – après tout, Paco Rabanne est "une maison de couture parisienne", avec tout l'aspect luxueux et sélectif que cela implique – mais plutôt la volonté d'offrir aux femmes une liberté non feinte et de suivre le rythme de leurs vies trépidantes. Les devancer d'un petit pas pour leur permettre d'avancer plus vite. Idem, chez Paco, elles ne marchent qu'à plat et les faire se sentir fortes, presque invincibles. "À notre époque, je pense qu'il n'y a pas de place pour la fragilité", explique le directeur artistique, formé notamment chez Balenciaga auprès de Nicolas Ghesquière.



PRINTEMPS AIME

PACO RABANNE EN MODE DÉCOLLAGE



JULIEN DOSSENA, DIRECTEUR ARTISTIQUE DE PACO RABANNE, A REMIS LA CÉLÈBRE MAISON FRANÇAISE EN ORBITE. ON ADORE SON APPROCHE RADICALE ET NOVATRICE, PLEINE D'ÉNERGIE ET DE DÉTERMINATION, UNE VISION VERY PRINTEMPS ! (RE)DÉCOUVERTE D'UNE GRIFFE MYTHIQUE.

PRINTEMPS AIME...

SA LIBERTÉ DE MOUVEMENT ET D'ATTITUDE

Passion, générosité et enthousiasme – le tout teinté d'une gentillesse palpable : il y a tout cela dans la voix de Julien Dossena. C'est réjouissant et surtout très contagieux. Au cours de notre rencontre, le créateur prononcera de nombreuses fois le mot "accessible", employé non pour signifier un côté populaire – après tout, Paco Rabanne est "une maison de couture parisienne", avec tout l'aspect luxueux et sélectif que cela implique – mais plutôt la volonté d'offrir aux femmes une liberté non feinte et de suivre le rythme de leurs vies trépidantes. Les devancer d'un petit pas pour leur permettre d'avancer plus vite (d'ailleurs, chez Paco, elles ne marchent qu'à plat) et les faire se sentir fortes, presque invincibles. "À notre époque, je pense qu'il n'y a pas de place pour la fragilité", explique le directeur artistique, formé notamment chez Balenciaga auprès de Nicolas Ghesquière.

NOUS AIMONS

UN LOOK POINTU, MAIS FACILE
T-SHIRT RÉSILLE ET JUPE IMPRIMÉ... VOILÀ UNE SILHOUETTE SIMPLE ET EFFICACE.



Processeur SASS



Les variables

Les variables sont très utilisées au sein des frameworks CSS, elles permettent d'améliorer la maintenance et ajoute la possibilité de surcharger les valeurs initiales.

Sass propose plusieurs types de variables :

- Nombre
- Chaîne de caractères
- Couleur
- Valeur avec unité (px, %, ...)
- Tableau de valeur (ou liste)
- Booléen
- Utilisant une fonction SASS

Les variables ont une portée globale, si un fichier est importé via un `@import` dans un fichier parent. Il pourra accéder aux valeurs de toutes les variables déclarées dans le fichier parent.

```
$font-size-base: 1.6rem;  
$font-name-base: Arial;  
$breakpoints: (  
    'm': (min-width: 50em),  
    'l': (min-width: 62.5em),  
    'xl': (min-width: 90em),  
);  
$medium: map-get($breakpoints, 'm');
```

```
body {  
    font-size: $font-size-base;  
    font-family: $font-name-base;  
}
```



CodePen

Le nesting

Le nesting est très pratique notamment lorsque l'on utilise une méthodologie telle que BEM.

SCSS

```
.block {  
  &__element {  
    &--modifier {  
      color: white;  
    }  
  }  
  &__autre {  
    width: 100%;  
  }  
}
```

CSS

```
.block__element-modifier {  
  color: white;  
}  
  
.block__autre {  
  width: 100%;  
}
```

Les extends

Les @extend peuvent être utiles pour récupérer le style d'un autre élément.

Cette fonctionnalité est à manier avec précaution car lorsque l'on étant un élément qui possède beaucoup de propriétés, il y a un gonflement du fichier de sortie.

```
.Btn {  
  padding: 1em 2em;  
  background: #eee;  
  color: #333;  
  border-radius: 10px;  
}  
  
.Btn--primary {  
  @extend .Btn;  
  background: #00964c;  
}
```

```
.Btn, .Btn--primary {  
  padding: 1em 2em;  
  background: #eee;  
  color: #333;  
  border-radius: 10px;  
}  
  
.Btn--primary {  
  background: #00964c;  
}
```

```
.message {  
  padding: .5em;  
}  
  
.important {  
  font-weight: bold;  
}  
  
.message-error {  
  @extend .message, .important;  
}
```

```
.message, .message-error {  
  padding: .5em;  
}  
  
.important, .message-error {  
  font-weight: bold;  
}
```



CodePen

En savoir plus sur @extend

Les mixins

Les **mixins** sont une des fonctionnalités les plus intéressantes dans SASS.

Ce sont de petits morceaux de code paramétrable et réutilisable qui renvoient une ou plusieurs propriétés. Elles peuvent également recevoir des paramètres.

On peut prévoir une valeur par défaut si le paramètre n'est pas renseigné lors de l'appel.

```
@mixin BtnState($bg: #fff, $bgHover: #000) {
  background: $bg;
  &:hover {
    background: $bgHover;
  }
}

.Btn {
  @include BtnState();
  display: block;
  padding: 1em 2em;
}

.Btn--secondary {
  @extend .Btn;
  @include BtnState(#eee);
}

.Btn, .Btn--secondary {
  background: #fff;
  display: block;
  padding: 1em 2em;
}
.Btn:hover, .Btn--secondary:hover {
  background: #000;
}

.Btn--secondary {
  background: #eee;
}
.Btn--secondary:hover {
  background: #000;
}
```



CodePen

Les fonctions

Les fonctions sont le résultat retourné une fois compilé. Elles s'utilisent plutôt dans des propriétés et ou des variables.

```
$font-base-size: 16;

@function remCalc($size, $base: $font-base-size) {
  $remSize: $size / $base;
  @return #{$remSize}rem;
}

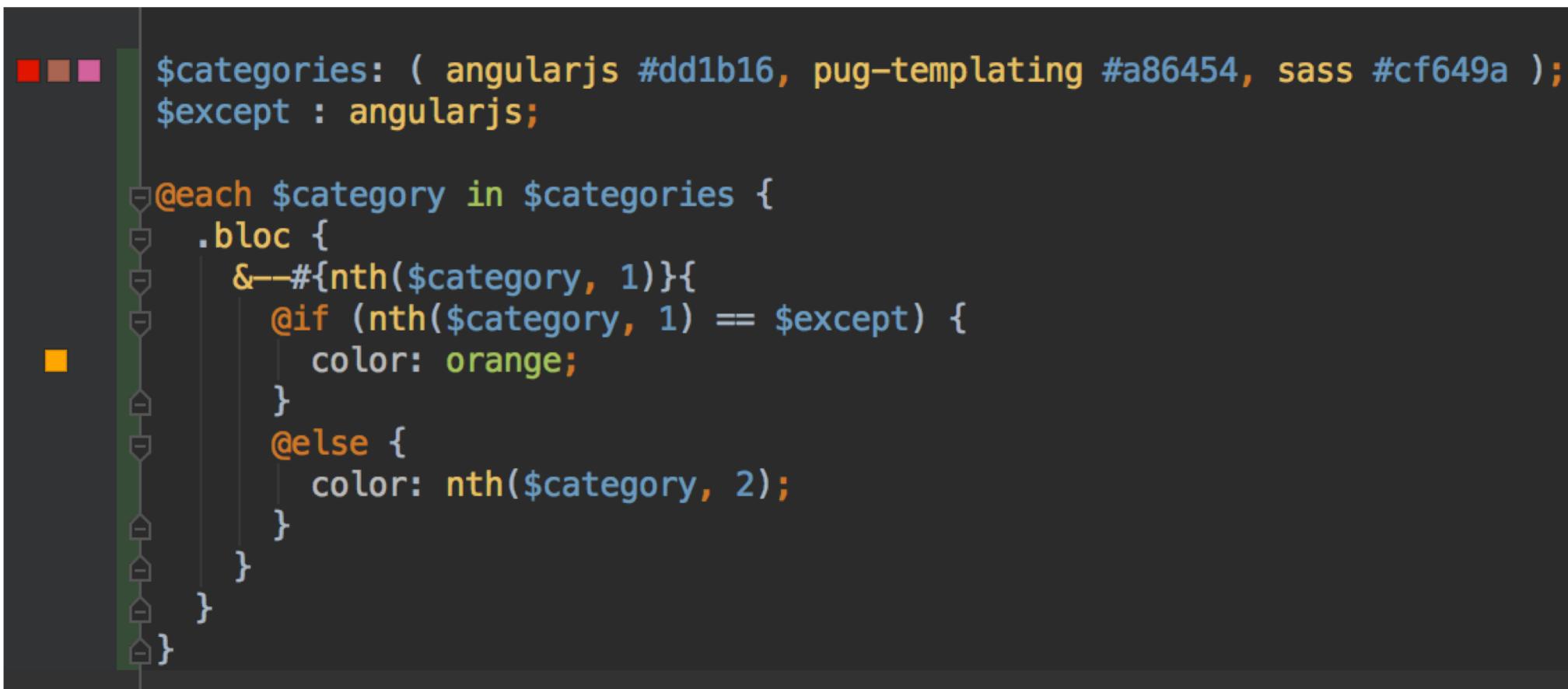
a {
  font-size: remCalc(20);
}
```

Des fonctions de base sont déjà proposées pour travailler sur les couleurs, les nombres, les chaînes de caractères, ...



Les boucles et conditions

En SASS, il est possible de créer liste de variables et donc des boucles pour les parcourir :



```
$categories: ( angularjs #dd1b16, pug-templating #a86454, sass #cf649a );
$except : angularjs;

@each $category in $categories {
  .bloc {
    &--#{nth($category, 1)}{
      @if (nth($category, 1) == $except) {
        color: orange;
      }
      @else {
        color: nth($category, 2);
      }
    }
  }
}
```



CodePen

Les directives

Comme pour les autres langages de développements, SASS offre des outils de traçage :

- **@error** : affiche des erreurs
- **@debug**
- **@warn** : affiche des warnings

```
@function color-variation($color) {  
  @if map-has-key($colors, $color) {  
    @return map-get($colors, $color);  
  }  
  @error "Invalid color name: `#${color}`.";  
}  
.element {  
  color: color-variation(brand-orange);  
}
```



CodePen

Map functions

Fonction	Description	exemple
map-get(\$map, \$key)		map-get(("red": #ff0000, "green": #00ff00), "green") Result: #00ff00
map-merge(\$map1, \$map2)		map-merge(("red": #ff0000, "green": #00ff00), ("blue": #0000ff)) Result: ("red": #ff0000, "green": #00ff00, "blue": #0000ff)
map-keys(\$map)		map-keys(("red": #ff0000, "green": #00ff00)) Result: ("red", "green")

Map functions

map-values(\$map)		map-values(("red": #ff0000, "green": #00ff00)) Result: (#ff0000, #00ff00)
map-has-key(\$map, \$key)		map-has-key(("red": #ff0000, "green": #00ff00), blue) Result: false

<https://github.com/samuel-gomez/formation-front-end/>



TP FINAL

INTÉGRER LE SITE BONDUELLE ☺

git checkout -f exo-bonduelle





Bonduelle en bref

2 777M € DE CHIFFRE
D'AFFAIRES

128 000 HA
CULTIVÉS DANS LE MONDE

54 SITES INDUSTRIELS

2 CIRCUITS DE DISTRIBUTION: LE RETAIL
ET LE FOOD SERVICE

3 100 AGRICULTEURS

500 VARIÉTÉS DE LÉGUMES

11 000 COLLABORATEURS

4 TECHNOLOGIES: CONSERVE, FRAIS,
SURGELÉ ET TRAITEUR

Quelques liens pour aller plus loin



Fonctions :

- <http://sass-lang.com/documentation/Sass/Script/Functions.html>

Extend :

- <https://la-cascade.io/sass-tout-sur-extend>

Mixins :

- <https://engageinteractive.co.uk/blog/top-10-scss-mixins>

Sass :

- <http://sass-lang.com/>

Aller (encore) plus loin



Quelques outils

- La console du navigateur ;-)
- <http://scg.ar-ch.org/>
- <https://codepen.io/>
- <http://www.colorzilla.com/gradient-editor/>
- <http://apps.eky.hk/css-triangle-generator/>

S'exercer

- <https://apprendre-html.3wa.fr/projets/>
- <https://www.sassmeister.com/>
- <http://www.flexboxdefense.com/>
- <https://flukeout.github.io/#>
- <https://flexboxfroggy.com/#fr>
- <http://cssgridgarden.com/#fr>
- <https://codepen.io/gsam59/>

Approfondir

- <https://css-tricks.com/snippets/css/fluid-typography/>
- <https://css-tricks.com/perfect-full-page-background-image/>
- <https://css-tricks.com/snippets/css/using-font-face/>
- <https://css-tricks.com/snippets/css/less-mixin-for-rem-font-sizing/>
- <https://css-tricks.com/combining-the-powers-of-sem-and-bio-for-improving-css/>
- <https://css-tricks.com/perfect-full-page-background-image/>
- <https://www.creativebloq.com/how-to/create-a-responsive-layout-with-css-grid>
- <https://www.smashingmagazine.com/2018/06/bem-for-beginners/>
- <https://www.alsacreations.com/article/lire/930-css3-media-queries.html>
- https://medium.com/@js_tut/css-tutorial-1-box-model-6ff80a9bbfda
- <https://cssgrid.io/>
- <https://hackernoon.com/all-you-need-to-know-about-css-in-js-984a72d48ebc>



Merci !

Samuel Gomez

Lead front-end

samuel.gomez@axa.fr



made by  with happiness