



NOUVEAUX OUTILS DE L'INTEGRATEUR

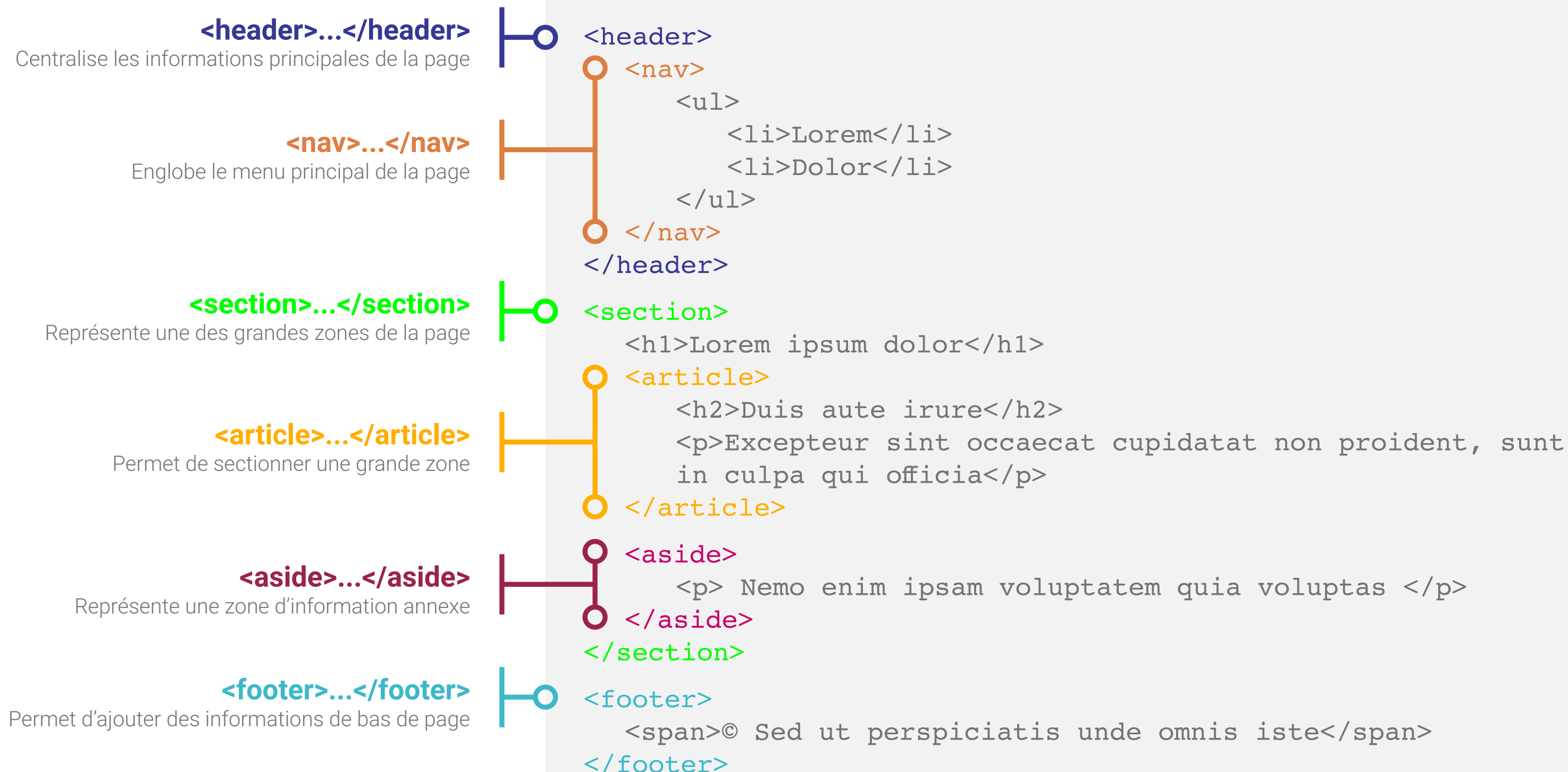
HTML & CSS AVANCÉS

NOUVEAUX OUTILS DE L'INTEGRATEUR

#Les nouvelles balises sémantiques HTML5	P.3
#Les techniques du Reset et du Normalize CSS	P.4
#Les sélecteurs d'éléments en CSS3	P.5
#Les sélecteurs CSS3 avancés	P. 6
#La priorité des sélecteurs CSS	P. 7
#Les pseudos-classes CSS	P. 8
#Les pseudos-éléments ::before et ::after	P.9
#Intégrer des polices avec @font-face	P.10
#Utiliser l'API Google Fonts pour les polices	P. 11

LES NOUVELLES BALISES SEMANTIQUES HTML5

Avec HTML5 des nouvelles balises sémantiques ont été élaborées, leurs fonctions principales et de mieux définir de façon native les zones d'une page Web. Une page Web optimisée doit aujourd'hui intégrer ces nouvelles balises pour s'inscrire dans une démarche de qualité Web.





LES TECHNIQUES DU RESET ET DU NORMALIZE CSS

Le Reset et le Normalize CSS sont des techniques qui visent à créer une configuration de base pour toutes les balises HTML d'une page Web. Elles partent du principe d'intégrer une feuille de style d'initialisation avant d'en créer une propre au site.

Le Reset CSS est une technique qui consiste à **réinitialiser à 0 la valeur de certains éléments HTML** afin d'éviter une partie des différences d'affichage des user-agents sur les divers navigateurs. **Le problème de cette technique**, c'est qu'elle annule certains styles par défaut des navigateurs qui sont réellement utiles, le cas le plus flagrant est celui des éléments de formulaire.

On conseille donc d'utiliser un Normalize CSS

car il vise plus précisément les différents éléments HTML et leur applique les styles nécessaires. Pour maîtriser parfaitement la technique du Normalize CSS **il est utile de créer son propre fichier normalize.css** basé sur des règles communes.

```
/* Exemple normalize.css */
html {
  font-family: sans-serif;
  font-size: 12px;
  color: black;
}

body, section, article {
  margin: 0;
}

a {
  text-decoration: none;
  background-color: transparent;
}

b, strong {
  font-weight: bold;
}
```

LES SELECTEURS D'ELEMENTS EN CSS3

En plus des sélecteurs classiques, CSS3 en propose des nouveaux pour aller plus loin dans la mise en page.

SÉLECTEUR	EXEMPLE	DESCRIPTION
element1~element2	p ~ ul	Tous les éléments précédé par un élément <p>
[attribute^=value]	a[href^="https"]	Tous les éléments <a> dont l'attribut href commence
[attribute\$=value]	a[href\$=".pdf"]	Tous les éléments <a> dont l'attribut href termine par ".pdf"
[attribute*=value]	a[href*="wf3"]	Tous les éléments <a> dont l'attribut href contient "wf3"
:checked	input:checked	Tous les éléments <input> dont l'attribut est checked
:disabled	input:disabled	Tous les éléments <input> dont l'attribut est disabled
:enabled	input:enabled	Tous les éléments <input> dont l'attribut est enabled
:empty	p:empty	Tous les éléments <p> qui ne pocèdent pas d'enfant
:first-child	p:first-child	Tous les éléments <p> qui sont les premiers enfants
:first-of-type	p:first-of-type	Tous les éléments <p> qui sont les premier enfants de leurs parents
:in-range	input:in-range	Tous les éléments <input> avec une valeur dans un intervalle spécifié
:invalid	input:invalid	Tous les éléments <input> d'un valeur non valide
:last-child	p:last-child	Tous les éléments <p> qui sont les derniers enfants de leurs parents
:last-of-type	p:last-of-type	Tous les éléments <p> qui sont les derniers enfants de leurs parents
:not(selector)	:not(p)	Tous les éléments que ne sont pas <p>

SÉLECTEUR	EXEMPLE	DESCRIPTION
:nth-child(n)	p:nth-child(2)	Tous les éléments <p> qui sont les deuxièmes
:nth-last-child(n)	p:nth-last-child(2)	Tous les éléments <p> qui sont les deuxièmes
:nth-last-of-type(n)	p:nth-last-of-type(2)	Tous les éléments <p> qui sont les deuxièmes enfants de leurs parents à partir du dernier enfant
:nth-of-type(n)	p:nth-of-type(2)	Tous les éléments <p> qui sont les les deuxièmes éléments <p> de leurs parents
:only-of-type	p:only-of-type	Tous les éléments <p> uniques de leurs parents
:only-child	p:only-child	Tous les éléments <p> qui sont les uniques enfants de leurs parents
:optional	input:optional	Tous les éléments <input> non-obligatoires
:out-of-range	input:out-of-range	Tous les éléments <input> avec une valeur hors d'un intervalle spécifié
:read-only	input:read-only	Tous les éléments <input> en lecture seule
:read-write	input:read-write	Tous les éléments <input> en lecture/écriture
:required	input:required	Tous les éléments <input> dont l'attribut <required> est spécifié
:root	:root	Le premier élément du DOM
::selection	::selection	Partie d'un élément sélectionnée par l'utilisateur
:valid	input:valid	Tous les éléments <input> valides



LES SELECTEURS CSS AVANCES

CSS permet d'associer aux sélecteurs classiques des sélecteurs associés qui permettent d'optimiser la programmation CSS car ils évitent l'ajout de nouvelles classes et id CSS.

Sélectionner les éléments enfants

Sélectionne tous les éléments <p> dont le parent est un élément <section>

```
section > p {  
  font-size: 12px;  
  color: black;  
}
```

Sélectionner les éléments suivants

Sélectionne la balise <p>...</p> qui est placée immédiatement après une balise </section>

```
section + p {  
  font-size: 12px;  
  color: black;  
}
```

Sélectionner les éléments précédent

Sélectionne la balise <p>...</p> qui est précédée par une balise </section>

```
section ~ p {  
  font-size: 12px;  
  color: black;  
}
```



LA PRIORITE DES SELECTEURS CSS

Il est possible de créer des styles généraux et des styles plus précis pour re définir des zones. L'ordre des priorités est le suivant : sélecteur de balise, sélecteur de classe et sélecteur d'id qui a la priorité sur les autres. Plus le sélecteur est précis, plus il est prioritaire.

Selecteur de balise HTML

Toutes les balises <p> du document font 15px

```
p {  
  font-size: 15px;  
}
```

Selecteur imbriqué

Les balises <p> situées dans une section font 20px

```
section p {  
  font-size: 20px;  
}
```

Selecteur multiple

Les balises <p> situées dans une section ayant la classe .uneClasse ou l'id #unId font 25px

```
.uneClasse p, #unId p {  
  font-size: 25px;  
}
```

Selecteur par .classe

Les balises <p> dont la classe est .unTexte feront 30px

```
.unTexte {  
  font-size: 30px;  
}
```

Selecteur par #id

Les balises <p> dont la id est .unTitre feront 35px même si elles ont la classe .unTexte

```
#unTitre {  
  font-size: 35px;  
}
```



LES PSEUDOS-CLASSES CSS

Une pseudo-classe CSS est un mot-clé ajouté au sélecteur pour indiquer un état particulier de l'élément qui doit être sélectionné. Par exemple, `:hover`, appliquera le style quand l'utilisateur survolera l'élément visé par le sélecteur.

:hover

Permet de spécifier l'apparence d'un élément au moment où l'utilisateur passe la souris dessus



```
a:hover {  
  color: red;  
  text-decoration: none;  
}
```

:active

Permet de définir l'apparence d'un élément au moment exact où l'utilisateur clique dessus



```
a:active {  
  color: white;  
  background-color: red;  
}
```

:focus

Permet de spécifier l'aspect d'un élément lorsque celui-ci est sélectionné par l'utilisateur



```
input:focus {  
  text-shadow: 2px 2px red;  
}
```

:first-child

Représente un élément qui est le premier enfant d'un élément parent



```
span:first-child {  
  text-transform: uppercase;  
}
```




LES PSEUDOS-ELEMENTS ::BEFORE ET ::AFTER

Les pseudos-éléments créent une balise qui sera le premier enfant de l'élément sélectionné ils sont généralement utilisés pour ajouter du contenu esthétique à un élément en utilisant la propriété CSS content. C'est une balise de type en-ligne par défaut.

::before

Ajoute un guillemet ouvert avant la balise blockquote>



```
blockquote::before {  
  content: "«\00A0";  
  color: blue;  
  font-size: 45px;  
}
```

::after

Ajoute un guillemet fermé après la balise blockquote>



```
blockquote::after {  
  content: '\00A0»';  
  color: red;  
  font-size: 45px;  
}
```

Dans une optique de qualité Web, il est préférable d'éviter d'utiliser la propriété content pour ajouter de contenu textuel. A utiliser avec parcimonie.



INTEGRER DES POLICES AVEC @FONT-FACE

La commande @font-face permet d'intégrer dans une page Web des fichiers de police afin de les utiliser dans la mise en page. Ces fichiers doivent être placés sur le serveur pour être appelé via le CSS. Les navigateurs ne supportant pas tous les mêmes fichiers de police il faut donc spécifier plusieurs formats.

Initialisation

Commande @font-face { ... } pour appeler le service

Nom de la police

Nom personnalisé de la police de caractère

Les fichiers de police

Les polices sont des fichiers externes

Intenet Explorer

Mode d'écriture spécifique pour IE

Appliquer la police

La police est ensuite appliquée avec la propriété font-family (sans-serif est la police de remplacement)

```
@font-face{  
  
    font-family : "maFonte";  
  
    /* Fichers des polices */  
    src : url('maFonte.woff') format('woff'),  
          url('maFonte.svg#abcd') format('svg'),  
          url('maFonte.ttf') format('truetype');  
  
    /* Fichier pour IE */  
    src : url('maFonte.eot');  
}  
  
h1 {  
    font-family: 'maFonte', sans-serif;  
}
```

Cette technique nécessite de savoir convertir des fichiers de police dans les différents formats



UTILISER L'API GOOGLE FONTS POUR LES POLICES

L'utilisation de l'API Google Fonts pour intégrer les polices est une solution à privilégier car elle permet de ne pas charger le serveur avec des fichiers de police et parce qu'elle offre un très grand choix de polices et de déclinaisons.

Importer la police

Les paramètres de la fontes sont envoyer dans l'adresse, chaque fonte a des paramètres spécifiques.



```
@import url(http://fonts.googleapis.com/css?  
family=Gloria+Hallelujah);
```

Appliquer la police

La police est ensuite appliquée avec la propriété font-family (cursive est la police de remplacement)



```
h1 {  
  font-family: 'Gloria Hallelujah', cursive;  
  color: red;  
  font-size: 70px;  
}
```

Que ce soit en utilisant @font-face ou Gopgle Font, il est primordial de définir une police de remplacement dans le cas ou la police importée ne fonctionnerait pas sur certains navigateurs.