

HTML



CSS



HTML & CSS



HTML & CSS



XYZ AGENCY Pôle web et prestations d'agence

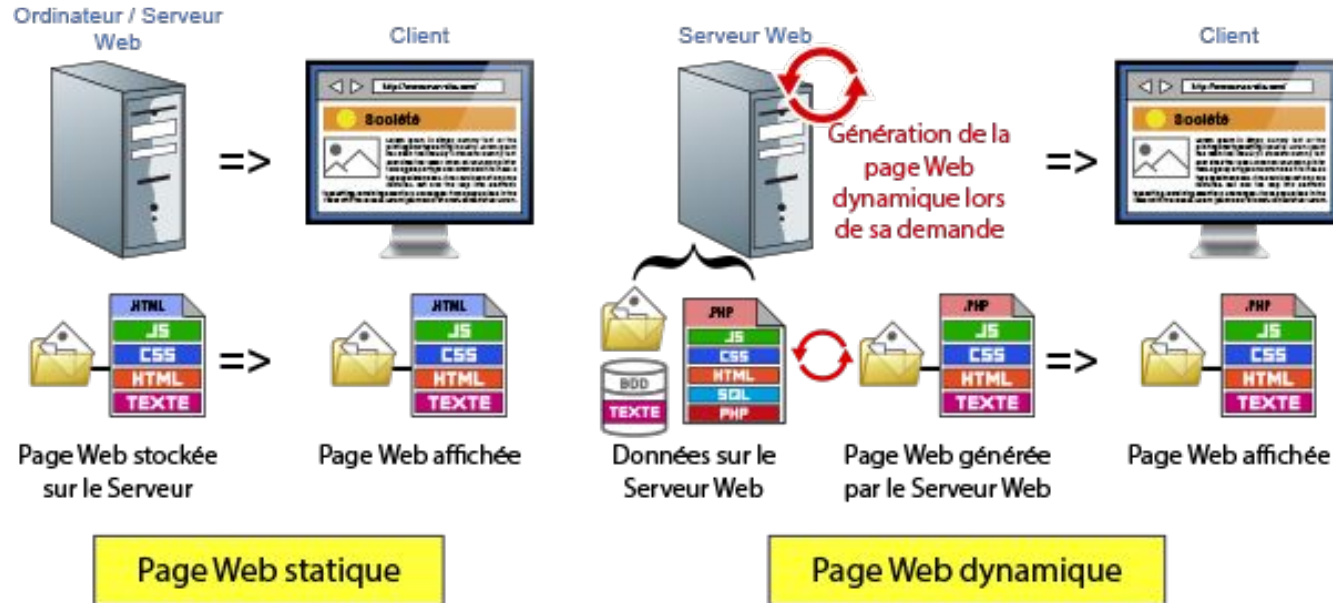
Nous nous occupons pour vous :
Conseils et mise en place de stratégie marketing
(publicité, A/B testing..)

[Nous découvrir](#)[Nos prestations](#)[Demander un devis](#)

Gestion des services

5

HTML & CSS



HTML & CSS

Client



Serveur



HTML & CSS

On parle de HTML5 et CSS3 mais pourquoi ?

Plus les clients HTTP évoluent et plus ils sont capables de comprendre plus d'attributs et d'éléments.

`<aside>` `<main>` `<nav>` `<header>` `<footer>` `<article>` `<video>` `<audio>`
`<section>`

Sont des éléments nouvellement créés.

L'attribut "download" (pour forcer le téléchargement d'un fichier) ou le `input[type="color"]` sont récents.

Plus d'infos ici :

<https://www.alsacreations.com/article/lire/1376-html5-section-article-nav-header-footer-aside.html>

HTML & CSS

Le W3C validator peut vous aider à avoir un code html plus propre et à mieux respecter les règles. Cependant, il faut prendre les résultats avec des pincettes.



HTML & CSS

Les limites :

- Gros sites
- Les constructeurs de thèmes
- Intégration de widgets

Error Attribute `height` not allowed on element `div` at this point.

From line 1147, column 4; to line 1150, column 1

```
inner">↵ <div class="googlemapsembed"↵ data-url="https://www.google.com/maps/embed?
pb=!1m14!1m8!1m3!1d10467.949985626818!2d-0.864407893701...
4f13.1!3m3!1m2!1s0x0%3A0x2c5c4bffa08c5ec1e!2sSDCOMM!5e0!3m2!1sfr!2sfr!4v1586441952341!5m2!1sfr!2sfr"↵ width="100%" height="300px"
```

Attributes for element `div`:

[Global attributes](#)

Si je mets pas "height", ma carte ne s'affichent pas.

HTML & CSS

Les limites :

- Gros sites
- Les constructeurs de thèmes
 - Intégration de widgets

Warning The `navigation` role is unnecessary for element `nav`.

From line 656, column 4; to line 656, column 112

```
</div><␣  
<nav ␣class="elementor-nav-menu--dropdown ␣elementor-nav-menu__container" ␣role="navigation" ␣aria-hidden="true"><ul ␣id
```

C'est un constructeur de thème, je n'y
peux rien...

HTML & CSS

HTML = HyperText Markup Language

CSS = Cascading Style Sheets

Il y a un abus de langage car le CSS est du HTML, c'est un attribut du CSS, mais cet attribut est tellement "volumineux" qu'il a été mis dans une feuille (de style) à part. Donc en fait, on va faire du HTML :)

HTML & CSS

ELEMENT

ATTRIBUT

HTML & CSS

Chaque élément HTML a une valeur d'affichage par défaut, en fonction du type d'élément dont il s'agit.

Il existe deux valeurs d'affichage : block et inline.

HTML & CSS

Un élément de type **bloc** commence toujours sur une nouvelle ligne et occupe toute la largeur disponible (s'étend à gauche et à droite dans la mesure du possible).

HTML & CSS

Voici des éléments de type bloc en HTML :

<address> <article> <aside> <blockquote> <canvas> <dd>
<div> <dl> <dt> <fieldset> <figcaption> <figure> <footer>
<form> <h1> <h6> <header> <hr> <main> <nav>
<noscript> <p> <pre> <section> <table> <tfoot>
<video>

HTML & CSS

Un élément de type **inline** ne démarre pas sur une nouvelle ligne et ne prend que la largeur nécessaire.

HTML & CSS

Voici des éléments de type inline en HTML :

<a> <abbr> <acronym> <bdo> <big>
 <button>
<cite> <code> <dfn> <i> <input> <kbd>
<label> <map> <object> <output> <q> <samp> <script>
<select> <small> <sub> <sup>
<textarea> <time> <tt> <var>

HTML & CSS

On ne met **JAMAIS** d'élément block à l'intérieur d'élément inline :

~~<p>Du texte</p>~~

C'est l'inverse : <p>coucou,tu vasbien ?</p>

On peut aussi (et ça arrive souvent) mettre des éléments block dans d'autres éléments block :

```
<div>  
  <p>Du<span>texte</span>ici</p>  
</div>
```

HTML & CSS

`<p class="main">Je suis un texte situé entre 2 balises "p" </p>`

ELEMENT

ATTRIBUT

HTML & CSS

`Texte cliquable`

ELEMENT

ATTRIBUT

HTML & CSS

L'élément `<a>` est l'enfant de l'élément `<p>`

L'élément `<p>` est le parent de l'élément `<a>`

`<p>` Je vous propose de visiter ma page

``à propos``

Pour mieux me découvrir

`</p>`

ELEMENT

ATTRIBUT

HTML & CSS

L'élément `<a>` est le petit enfant de l'élément `<div>`

L'élément `<p>` est le parent de l'élément `<a>`

```
<div id="container">
```

```
  <p> Je vous propose de visiter ma page
```

```
    <a href="/a-propos.html" target="_blank">à propos</a>
```

```
    Pour mieux me découvrir
```

```
  </p>
```

```
</div>
```

ELEMENT

ATTRIBUT

HTML & CSS

L'élément `<h1>` est le l'enfant de l'élément `<div>`

L'élément `<div>` a 2 enfants `<h1>` et `<p>`

```
<div id="container">
```

```
<h1>Le titre principal de la page</h1>
```

```
<p> Je vous propose de visiter ma page
```

```
<a href="/a-propos.html" target="_blank">à propos</a>
```

```
  Pour mieux me découvrir
```

```
</p>
```

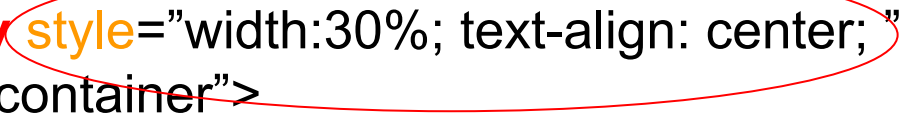
```
</div>
```

ELEMENT

ATTRIBUT

HTML & CSS

C'est ça, le
CSS



```
<div style="width:30%; text-align: center;" class="main"  
id="container">  
  <h1>Le titre principal de la page</h1>  
  <p> Je vous propose de visiter ma page  
    <a href="/a-propos.html" target="_blank">à propos</a>  
    Pour mieux me découvrir  
  </p>  
</div>
```

En fait, dire que l'on fait du HTML / CSS est un abus de langage car en réalité, on fait juste de l'HTML. Le CSS est juste un attribut style. Cet attribut est tellement volumineux, que pour une meilleure gestion / organisation de code, on le met dans une feuille à part.

ELEMENT

ATTRIBUT

HTML & CSS

```
<style>
  div{
    width:30%;
    text-align: center;
  }
</style>
```

```
<div class="main" id="container">
  <h1>Le titre principal de la page</h1>
  <p> Je vous propose de visiter ma page
    <a href="/a-propos.html" target="_blank">à propos</a>
    Pour mieux me découvrir
  </p>
</div>
```

ELEMENT

ATTRIBUT

HTML & CSS

Style.css

```
div{  
  width:30%;  
  text-align: center;  
}
```

Index.html

`<link rel="stylesheet" href="style.css">`

`<div class="main" id="container">`

`<h1>Le titre principal de la page</h1>`

`<p> Je vous propose de visiter ma page`

`à
propos`

Pour mieux me découvrir

`</p>`

`</div>`

HTML & CSS

C'est quoi les attributs "class" et "id" ?

Ce sont des sélecteurs.

La différence entre les 2 c'est qu'il peut il avoir plusieurs class du même nom dans un document et qu'il peut y avoir plusieurs class dans le même attribut :

```
<div class="main onLeft background" id="container">  
  <p class="main"> Je vous propose de visiter ma page  
    <a href="/a-propos.html" target="_blank">à propos</a>  
    Pour mieux me découvrir  
  </p>  
</div>
```

HTML & CSS

Cependant les id sont uniques ! et il ne peut y avoir qu'un id par attribut :

```
<div class="main onLeft background" id="container">  
  <p id="first" class="main">  
    Je vous propose de visiter ma page  
    <a href="/a-propos.html" target="_blank">à propos</a>  
    Pour mieux me découvrir  
  </p>  
</div>
```

HTML & CSS

OK, mais à quoi ça sert ?!?!

HTML & CSS

Préfixe “.” = class

Préfixe “#” = id

Style.css

```
.second{  
    color:#fff;  
}
```

```
#test{  
    color:grey;  
}
```

Index.html

```
<link rel="stylesheet" href="style.css">
```

```
<p class="first main" id="test">
```

Je vous propose de visiter ma page

Pour mieux me découvrir

```
</p>
```

```
<p class="second">
```

Autre texte ici

```
</p>
```

HTML & CSS

Préfixe “.” = class

Préfixe “#” = id

Style.css

```
.first.main{  
  color:#fff;  
}
```

Index.html

```
<link rel="stylesheet" href="style.css">
```

```
<p class="first main">
```

Je vous propose de visiter ma page
Pour mieux me découvrir

```
</p>
```

```
<p class="first">
```

Autre texte ici

```
</p>
```

HTML & CSS

Préfixe “.” = class

Préfixe “#” = id

Style.css

```
.first.main .coucou{  
    color:#fff;  
}
```

Index.html

```
<link rel="stylesheet" href="style.css">
```

```
<p class="first main">  
    <span class="coucou">Je vous propose  
de visiter ma page pour mieux me  
découvrir</span>  
</p>  
<p class="first main">  
<span>Autre texte</span>  
</p>
```

HTML & CSS

Préfixe “.” = class

Préfixe “#” = id

Style.css

```
.belette{  
    color:#fff;  
}
```

Index.html

```
<link rel="stylesheet" href="style.css">
```

```
<p class="first main">
```

```
<span class="belette">Je vous propose  
de visiter ma page pour mieux me  
découvrir</span>
```

```
</p>
```

```
<p class="first main">
```

```
<span>Autre texte</span>
```

```
</p>
```


HTML & CSS

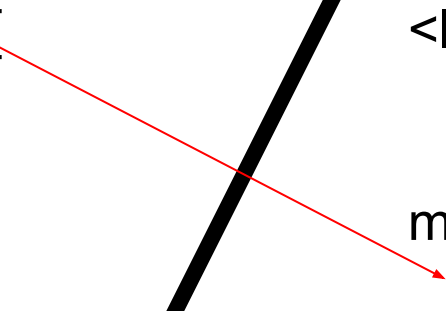
Style.css

```
p:nth-child(2){  
  color:#fff;  
}
```

Index.html

```
<link rel="stylesheet" href="style.css">
```

```
<p>Je vous propose de visiter ma page pour  
mieux me découvrir</p>  
<p>Autre texte</p>
```



HTML & CSS

Style.css

```
p[title]{  
    color:#fff;  
}
```

Index.html

```
<link rel="stylesheet" href="style.css">
```

```
<p title="paragraphe">Je vous propose de  
visiter ma page pour mieux me découvrir</p>  
<p>Autre texte</p>
```

HTML & CSS

Style.css

```
p[title="belette"]{  
    color:#fff;  
}
```

Index.html

```
<link rel="stylesheet" href="style.css">
```

```
<p title="paragraphe">Je vous propose de  
visiter ma page pour mieux me découvrir</p>  
<p title="belette">Autre texte</p>
```

HTML & CSS

Style.css

```
p[title^="para"]{  
    color:#fff;  
}
```

Index.html

```
<link rel="stylesheet" href="style.css">
```

```
<p title="paragraphe">Je vous propose de  
visiter ma page pour mieux me découvrir</p>
```

```
<p title="belette">Autre texte</p>
```

```
<p title="parachute">Autre texte</p>
```

HTML & CSS

Style.css

```
a[href^="http://mysite.com"],  
a[href^="http://www.mysite.com"] {  
    color: red;  
}
```

Index.html

```
<link rel="stylesheet" href="style.css">  
  <li><a href="/index.php">My site's page</a></li>  
  <li><a href="http://example.com">External link</a></li>  
  <li><a href="http://mysite.com">My site's base URL without  
www</a></li>
```

HTML & CSS

Bon, bah, on va pratiquer.....

HTML & CSS

Les fonts

HTML & CSS

fonts.google.com/specimen/Akaya+Telivigala?preview.text_type=custom

Akaya Telivigala [Download family](#)

[Select styles](#) [Glyphs](#) [About](#) [License](#) [Pairings](#)

Regular 400

Almost before we knew it, we had left the [Remove this style](#)

Glyphs

A	B	C	Č	Ć	D	Đ	E	F	G	H	I	J	K	L	M	N	O	P	Q
R	S	Š	T	U	V	W	X	Y	Z	Ž	a	b	c	č	ć	d	đ	e	f
g	h	i	j	k	l	m	n	o	p	q	r	s	š	t	u	v	w	x	y
z	ž	అ	ఆ	ఇ	ఈ	ఉ	ఊ	ఋ	ఎ	ఏ	ఐ	ఒ	ఓ	ఔ	అం	అం	అం	అం	క
ఖ	గ	ఘ	జ్	ఞ	చ	ఛ	జ	ఝ	ఞ	ట	ఠ	డ	ఢ	ణ	త	థ	ద	ధ	న

Selected family [Share](#) [X](#)

You can now share your selected families with others. [Dismiss](#)

Akaya Telivigala

Regular 400 [Remove](#)

[Add more styles](#) [Remove all](#)

To embed a font, copy the code into the <head> of your html

☒ <link> ☐ @import

```
<link rel="preconnect" href="https://fonts.gstatic.com">
<link href="https://fonts.googleapis.com/css2?family=Akaya+Telivigala&display=swap" rel="stylesheet">
```

CSS rules to specify families

[API docs](#) [Download all](#)

HTML & CSS

Style.css

```
a[href^="http://mysite.com"],  
a[href^="http://www.mysite.com"] {  
  Font-family: Akaya;  
}
```

Index.html

```
<link rel="stylesheet" href="style.css">
```

```
<link href="https://fonts.googleapis.com/css2?family=Akaya+Telivigala&display=swap"  
rel="stylesheet">
```

```
<li><a href="/index.php">My site's page</a></li>
```

```
<li><a href="http://example.com">External link</a></li>
```

```
<li><a href="http://mysite.com">My site's base URL without www</a></li>
```

HTML & CSS

Style.css

```
@import url('https://fonts.googleapis.com/css2?family=Akaya+Telivigala&display=swap');  
a[href^="http://mysite.com"], a[href^="http://www.mysite.com"] {  
  Font-family: Akaya;  
}
```

Index.html

```
<link rel="stylesheet" href="style.css">  
<li><a href="/index.php">My site's page</a></li>  
<li><a href="http://example.com">External link</a></li>  
<li><a href="http://mysite.com">My site's base URL without www</a></li>
```

HTML & CSS

Display et Position

HTML & CSS

Il existe plusieurs propriétés de display :

Inline, block, contents, flex, grid, inline-block, inline-flex, inline-grid, inline-table, list-item, run-in, table, table-caption, table-column-group, table-header-group, table-footer-group, table-row-group, table-cell, table-column, table-row, none, initial

Par défaut, un élément HTML a un de ces 4 display :
Inline, block, inline-block et none

D'ailleurs, si vous souhaitez tout tester :

https://www.w3schools.com/cssref/playit.asp?filename=playcss_display&preval=grid

HTML & CSS

Au fait, connaissez vous la particularité d'un élément
“inline-block” ?

Pouvez-vous me citer un élément qui est nativement
considéré en “inline-block” par les clients http ?

HTML & CSS

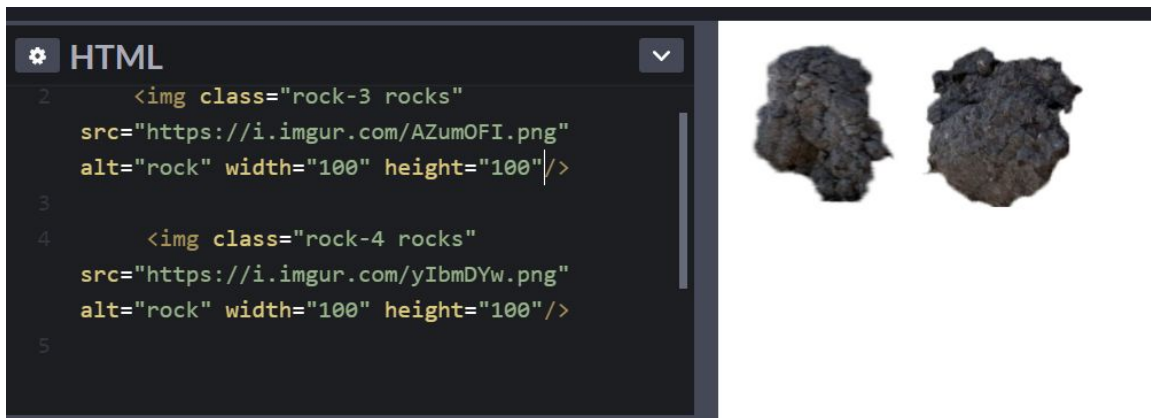
Par rapport à `display : inline`, la principale différence est que `display : inline-block` permet de définir une largeur et une hauteur sur l'élément.

Avec `display : inline-block`, les marges et les intercalaires supérieurs et inférieurs sont respectés, ce qui n'est pas le cas avec `display : inline`.

Par rapport à `display : block`, la principale différence est que `display : inline-block` n'ajoute pas de saut de ligne après l'élément, de sorte que l'élément peut être placé à côté d'autres éléments.

HTML & CSS

Les images sont des éléments “inline-block”.
Car elles peuvent être alignées et ont un attribut hauteur
(height) et largeur (width)



HTML & CSS

Il existe plusieurs propriétés de position :
static, relative, fixed, absolute, sticky

Vous connaissez toutes ces propriétés ?

HTML & CSS

Static

Les éléments HTML sont positionnés de manière statique par défaut.

Les éléments positionnés statiquement ne sont pas affectés par les propriétés top, bottom, left et right.

Un élément avec position : static ; n'est pas positionné de manière particulière ; il est toujours positionné selon le flux normal de la page.

HTML & CSS

Fixed

Un élément avec `position : fixed ;` est positionné par rapport à la fenêtre d'affichage, ce qui signifie qu'il reste toujours au même endroit, même si la page défile. Les propriétés `top`, `right`, `bottom` et `left` sont utilisées pour positionner l'élément.

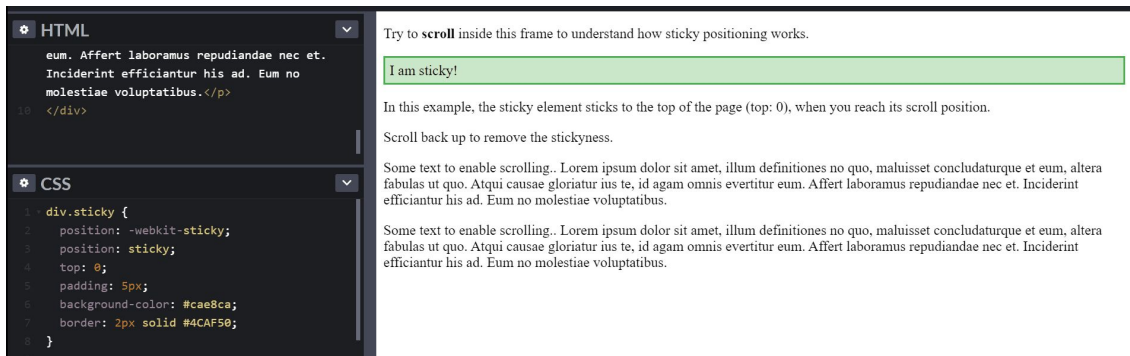


HTML & CSS

Sticky

Un élément avec position : sticky ; est positionné en fonction de la position de défilement de l'utilisateur.

Un élément sticky bascule entre relatif et fixe, en fonction de la position de défilement. Il est positionné de manière relative jusqu'à ce qu'une position de décalage donnée soit atteinte dans la fenêtre d'affichage - il est alors "collé" en place (comme position:fixed).

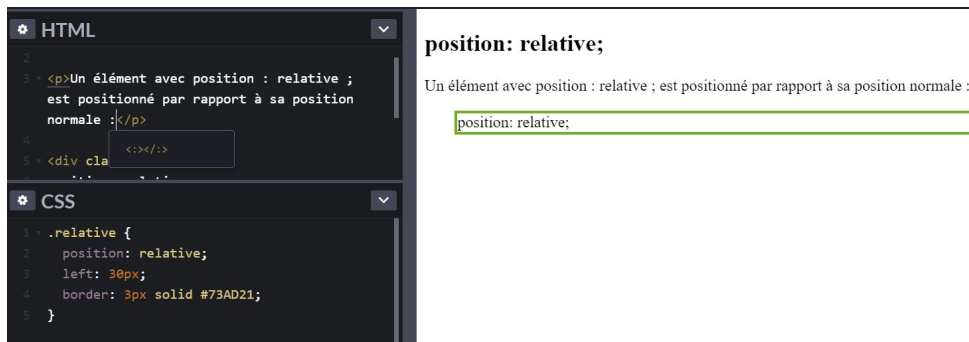


HTML & CSS

Relative

Un élément avec position : relative ; est positionné par rapport à sa position normale.

Le réglage des propriétés top, right, bottom et left d'un élément positionné de manière relative entraînera son éloignement de sa position normale. Les autres contenus ne seront pas ajustés pour s'insérer dans un quelconque espace laissé par l'élément.



The screenshot shows a code editor with two panels: HTML and CSS. The HTML panel contains the following code:

```
1 <p>Un élément avec position : relative ;  
2   est positionné par rapport à sa position  
3   normale :</p>  
4  
5 <div class="relative">  
6   <div class="content">  
7     <p>Contenu de l'élément relatif</p>  
8   </div>  
9 </div>
```

The CSS panel contains the following code:

```
1 .relative {  
2   position: relative;  
3   left: 30px;  
4   border: 3px solid #73AD21;  
5 }
```

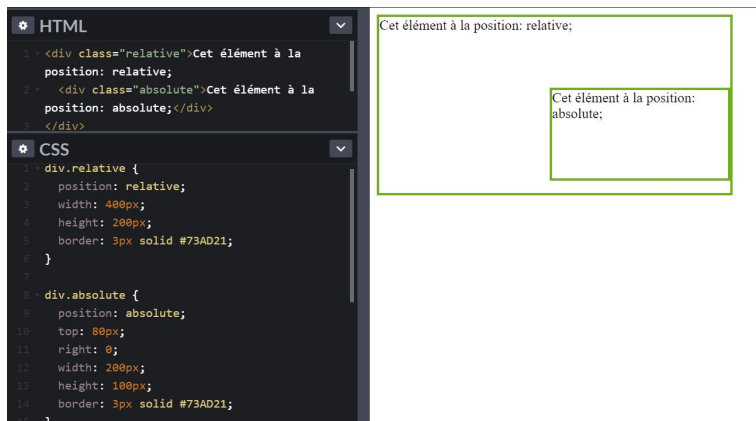
Below the code panels, the text "position: relative;" is displayed, followed by a description: "Un élément avec position : relative ; est positionné par rapport à sa position normale :". Below this text, the CSS rule "position: relative;" is shown again, highlighted with a green border.

HTML & CSS

Absolute

Un élément avec position : absolute ; est positionné par rapport au parent positionné le plus proche (au lieu d'être positionné par rapport à la fenêtre d'affichage, comme fixed).

Toutefois, si un élément positionné de manière absolue n'a pas de parent, il utilise le corps du document et se déplace avec le défilement de la page.

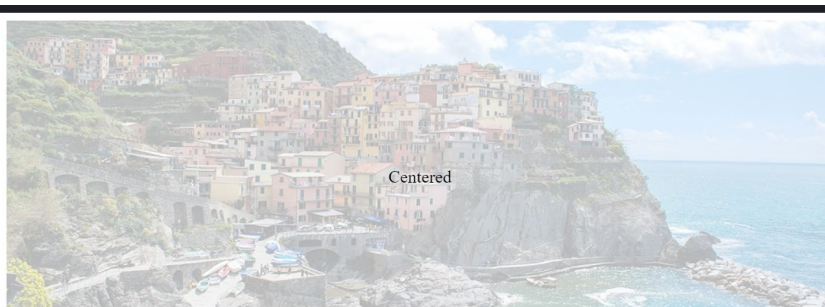


HTML & CSS

Absolute

```
<div style="position: relative;">  
    
  <div style="position: absolute;" top: 50%; width: 100%; text-align:  
center;font-size: 18px;>Centered</div>  
</div>
```

```
HTML  
1 <div class="container">  
2     
3   <div class="center">Centered</div>  
4 </div>  
5  
CSS  
1 .container {  
2   position: relative;  
3 }  
4  
5 .center {  
6   position: absolute;  
7   top: 50%;  
8   width: 100%;  
9   text-align: center;  
10  font-size: 18px;  
11 }
```



HTML & CSS

A vous !

HTML & CSS

Les pseudo-classes

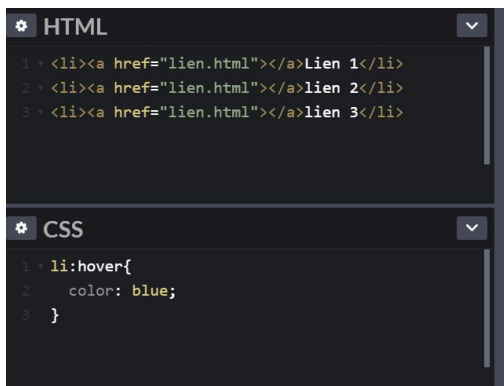
HTML & CSS

:pseudo-classes

Elles permettent de changer le style en offrant plus de précision dans les sélecteurs ou en fonction de certaines actions. On les reconnaît à leur syntaxe avec “:” après un sélecteur (ex : a:visited)

HTML & CSS

La pseudo-classe “:hover” change le style d’un élément à son survol.




```
HTML
1 <li><a href="lien.html"></a>Lien 1</li>
2 <li><a href="lien.html"></a>lien 2</li>
3 <li><a href="lien.html"></a>lien 3</li>

CSS
1 li:hover{
2     color: blue;
3 }
```

- Lien 1
- lien 2
- lien 3

HTML & CSS

Dans ce cas, la pseudo-classe `:nth-child()` nous permet de sélectionner un élément “p” en particulier (le deuxième dans l'exemple) quand il n'y a pas de sélecteur.



```
HTML
1 <p>Un texte pas comme les autres</p>
2 <p>Un texte pas comme les autres</p>
3 <p>Un texte pas comme les autres</p>
4 <p>Un texte pas comme les autres</p>
5 <p>Un texte pas comme les autres</p>
6 <p>Un texte pas comme les autres</p>
7 <p>Un texte pas comme les autres</p>

CSS
1 p:nth-child(2){
2   color:blue;
3 }
```

Un texte pas comme les autres

Un texte pas comme les autres

Un texte pas comme les autres

Un texte pas comme les autres

Un texte pas comme les autres

Un texte pas comme les autres

Un texte pas comme les autres

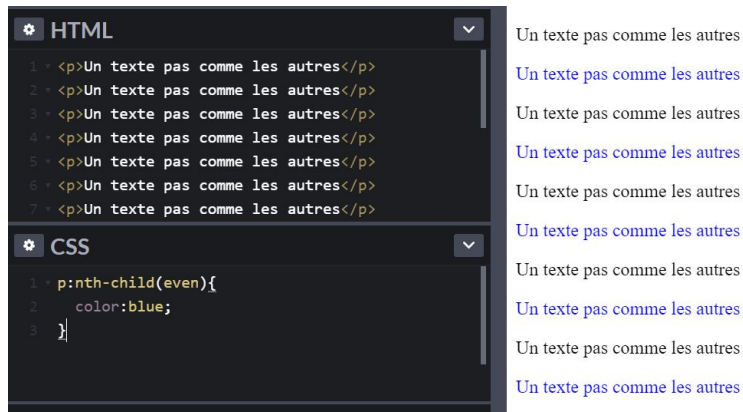
Un texte pas comme les autres

Un texte pas comme les autres

Un texte pas comme les autres

HTML & CSS

Lorsque l'on change le numéro de l'élément à sélectionner par "even" (`:nth-child(even)`), il sélectionne tous les élément "p" pairs.



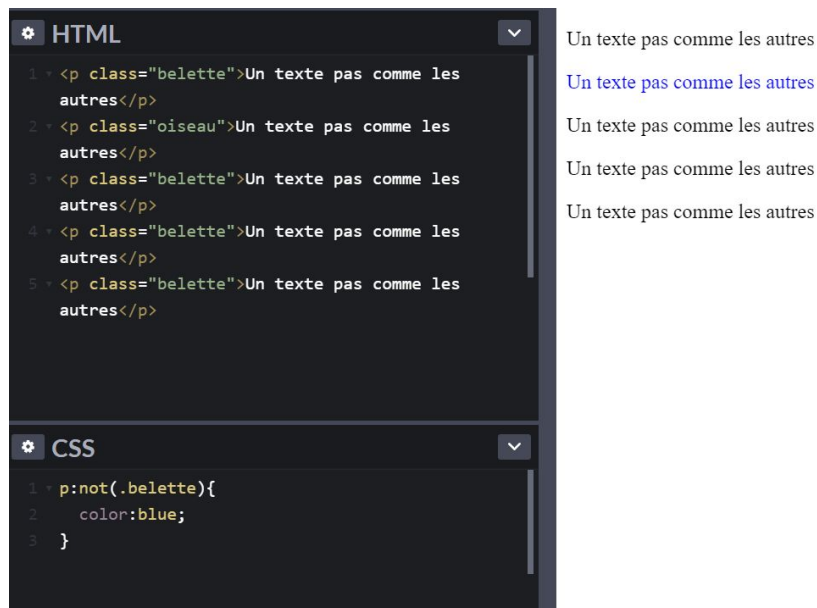
HTML & CSS

Lorsque l'on change le numéro de l'élément à sélectionner par "even" (:nth-child(odd)), il sélectionne tous les élément "p" impairs.



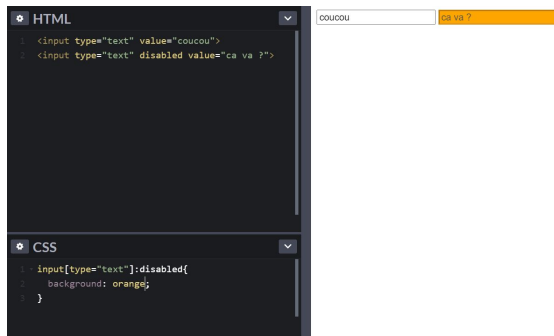
HTML & CSS

La pseudo-classe `:not()` sélectionne les attributs n'ayant pas un certain attribut. Dans l'exemple, on sélectionne les Éléments “p” n'ayant pas la classe “belette”



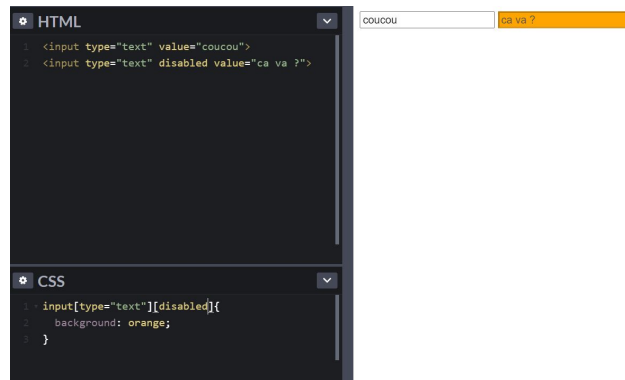
HTML & CSS

Certaines pseudo-classes sont “discutables” car ne servent pas à grand chose... Par exemple, `:disabled` permet de sélectionner un champs “input” avec l’attribut “disabled” (qu’on ne peut pas sélectionner). Alors que le CSS prévoit déjà la possibilité de sélectionner un élément par un attribut (ex : `[disabled]`). Donc dans le code comme le reste, tout n’est pas bon à prendre.



```
HTML
1 <input type="text" value="coucou">
2 <input type="text" disabled value="ca va ?">

CSS
1 input[type="text"]:disabled{
2   background: orange;
3 }
```

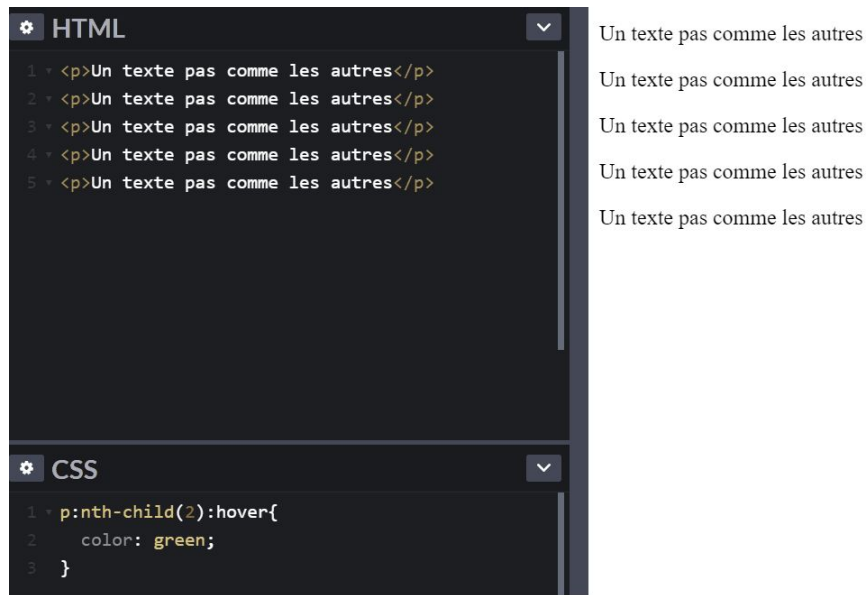


```
HTML
1 <input type="text" value="coucou">
2 <input type="text" disabled value="ca va ?">

CSS
1 input[type="text"][disabled]{
2   background: orange;
3 }
```

HTML & CSS

Il est possible de cumuler des pseudo-classes (pas toutes, faut tester)



HTML & CSS

Attention, toutes les pseudo-classes ne fonctionnent pas sur tous les clients http (c'est un navigateur pour ceux qui auraient oublié ^^), soit car ils sont trop récents ou qu'ils sont en phase de test.

Vous avez tout ici :

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

HTML & CSS

Les pseudo-éléments

HTML & CSS

::pseudo-element

Elles permettent de changer le style en offrant plus de précision dans les sélecteurs ou en fonction de certaines actions. On les reconnaît à leur syntaxe avec “::” après un élément.

HTML & CSS

Le pseudo-élément `::after` permet d'insérer du texte (et un style) via le CSS, après le texte d'origine.

```
HTML
1 <p class="texte-ennuyeux">Voici un peu de vieux texte
  ennuyeux ordinaire.</p>
2 <p>Voici un peu de texte normal qui n'est ni ennuyeux, ni
  intéressant.</p>
3 <p class="texte-interessant">Contribuer à MDN est facile
  et amusant.
4 Cliquez simplement sur le bouton Modifier pour ajouter
  de nouveaux exemples vivants, ou améliorer ceux existants.
  </p>

CSS
1 .texte-interessant::after {
2   content: "<- maintenant, cela *est* intéressant !";
3   color: green;
4 }
5
6 .texte-ennuyeux::after {
7   content: "<- ENNUYEUX !";
8   color: red;
9 }
```

Voici un peu de vieux texte ennuyeux ordinaire.<- ENNUYEUX !

Voici un peu de texte normal qui n'est ni ennuyeux, ni intéressant.

Contribuer à MDN est facile et amusant. Cliquez simplement sur le bouton Modifier pour ajouter de nouveaux exemples vivants, ou améliorer ceux existants.<- maintenant, cela *est* intéressant !

HTML & CSS

Bon, je vous cache pas que le pseudo-élément `::before`, c'est pareil que `::after` mais pour caler les infos avant...

```
HTML
1 <p class="texte-ennuyeux">Voici un peu de vieux texte
  ennuyeux ordinaire.</p>
2 <p>Voici un peu de texte normal qui n'est ni ennuyeux, ni
  intéressant.</p>
3 <p class="texte-interessant">Contribuer à MDN est facile et
  amusant.

CSS
1 .texte-interessant::before {
2   content: "<- maintenant, cela *est* intéressant !";
3   color: green;
4 }
5
6 .texte-ennuyeux::before {
7   content: "<- ENNUYEUX !";
8   color: red;
9 }
```

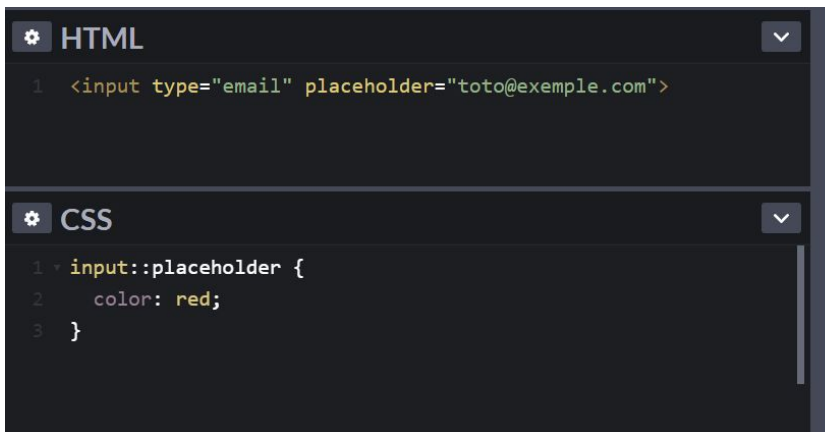
<- ENNUYEUX !Voici un peu de vieux texte ennuyeux ordinaire.

Voici un peu de texte normal qui n'est ni ennuyeux, ni intéressant.

<- maintenant, cela *est* intéressant !Contribuer à MDN est facile et amusant. Cliquez simplement sur le bouton Modifier pour ajouter de nouveaux exemples vivants, ou améliorer ceux existants.

HTML & CSS

Comme les pseudo-classes, certains pseudo-éléments servent pas à grand chose. Un simple “input[placeholder]” fonctionne mais bon....



```
HTML
1 <input type="email" placeholder="toto@exemple.com">

CSS
1 input::placeholder {
2   color: red;
3 }
```

toto@exemple.com

HTML & CSS

Les média queries

HTML & CSS

Les média queries gèrent le responsive et l'affichage d'une page web selon le support (print, screen, all, handheld, projection,...) pour optimiser le rendu. Les media queries s'écrivent dans le CSS.

HTML & CSS

Dans cet exemple, on demande à la page web de supprimer le menu, footer, et la barre latéral (aside) ainsi que d'augmenter la police à 120% et de mettre tout le texte en noir, si on souhaite imprimer la page.

```
@media print {  
  #menu, #footer, aside {  
    display:none;  
  }  
  body {  
    font-size:120%;  
    color:black;  
  }  
}
```

HTML & CSS

Dans cet exemple, à la classe “bloc” de changer de comportement si elle est affichée sur un support à écran (screen) et si la largeur maximale de la page est de 640px (donc un petit appareil)

```
@media screen and (max-width: 640px) {  
  .bloc {  
    display: block;  
    clear: both;  
  }  
}
```

HTML & CSS

Il est également possible de spécifier un media query directement dans la balise “link” via l’attribut “media”. Dans ce cas, dans la feuille de style “smallscreen.css”, on mettra du css “normal” sans “@ media...” car avec l’attribut “media” impose à ce CSS de s’activer seulement sous certaines conditions.

```
<link rel="stylesheet" media="screen and (max-width: 640px)" href="smallscreen.css" type="text/css" />
```

HTML & CSS

En fait vous avez 2 choix, soit vous intégrez vos médias queries dans votre CSS général et il faudra utiliser “@media....{....}” (si vous en avez peu) OU vous mettez tout dans une feuille de style à part (si vous avez beaucoup de code) et vous spécifiez les conditions via “media”.

```
@media screen and (max-width: 640px) {  
  .bloc {  
    display: block;  
    clear: both;  
  }  
}
```

```
<link rel="stylesheet" media="screen and (max-width: 640px)" href="smallscreen.css"  
type="text/css" />
```

HTML & CSS

Exemple concret, je demande au CSS via une media query de faire disparaître la classe “hide” si la largeur de la fenêtre est de minimum 800px



HTML & CSS

Il est également possible de cumuler plusieurs media-queries. Dans cet exemple, quand l'écran est entre 800px et 850px, on change la police de la classe "hide", on change la propriété et les couleurs des classes "block".



```
HTML
2 <p class="block hide">Je vais disparaître</p>
3 <p class="block">Je reste</p>
4 <p class="block">Je reste</p>

CSS
1 @import url('https://fonts.googleapis.com/css2?family=Fascinate&display=swap');
2
3 p.block:not(.hide){
4   color: orange;
5 }
6 @media screen AND (min-width: 800px) AND (max-width: 850px){
7   .hide{
8     font-family: 'Fascinate';
9   }
10  .block{
11    display: inline;
12  }
13  p.block:not(.hide){
14    color: green;
15  }
16 }
```

Je reste

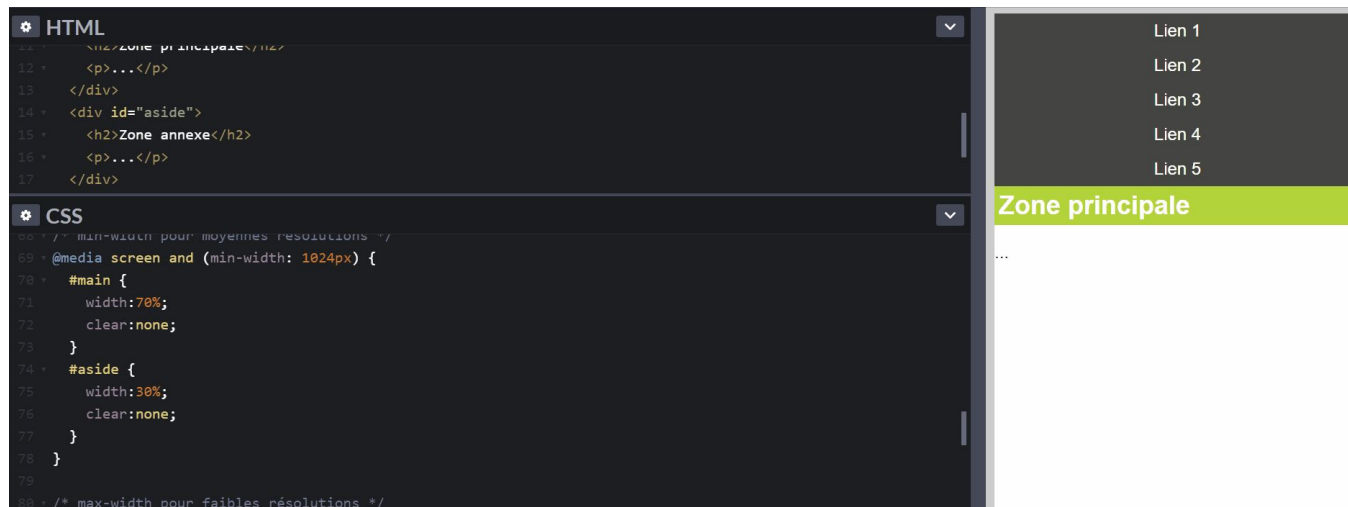
Je vais disparaître

Je reste

Je reste

HTML & CSS

Sinon, à part changer des couleurs et des polices, les media-queries gèrent le responsive. Sachez aussi que dans la vraie vie, on utilise des frameworks pour gérer le responsive. C'est rare de le gérer manuellement.



The screenshot displays a web browser interface. On the left, the developer tools are open, showing the HTML and CSS code. The HTML code includes a sidebar with five links and a main content area. The CSS code uses media queries to define the layout for different screen widths.

```
HTML
11 <div>
12 <p>...</p>
13 </div>
14 <div id="aside">
15 <h2>Zone annexe</h2>
16 <p>...</p>
17 </div>

CSS
68 /* min-width pour moyennes resolutions */
69 @media screen and (min-width: 1024px) {
70   #main {
71     width:70%;
72     clear:none;
73   }
74   #aside {
75     width:30%;
76     clear:none;
77   }
78 }
79
80 /* max-width pour faibles résolutions */
```

The rendered page on the right shows a dark sidebar with five links: Lien 1, Lien 2, Lien 3, Lien 4, and Lien 5. Below the links is a light green bar labeled "Zone principale".

HTML & CSS

Vous avez aussi d'autres fonctionnalités, comme la possibilité d'activer un style en fonction de l'orientation d'un écran. Ce qui revient à faire:

```
@media screen AND (max-width: 610px){/*pour le portrait*/  
@media screen AND (min-width: 611px){/*pour le paysage*/
```

```
<link rel="stylesheet" media="(orientation:portrait)" href="portrait.css">  
<link rel="stylesheet" media="(orientation:landscape)" href="paysage.css">
```

Plus d'infos:

https://developer.mozilla.org/fr/docs/Web/CSS/Media_Queries/Using_media_queries
<https://www.alsacreations.com/article/lire/930-css3-media-queries.html>

Les transitions et animations

HTML & CSS

Le concept de transition est assez simple. Dans l'exemple, on connaît déjà :hover (il y a juste la propriété "rotate" en plus). On peut aussi voir une nouvelle propriété "transition" dans la classe "box". Cette propriété va définir le passage entre ".box" et ".box:hover". Il fait passer toutes les nouvelles propriété dans ":hover" en 2 secondes.

```
HTML
<div class="box"></div>

CSS

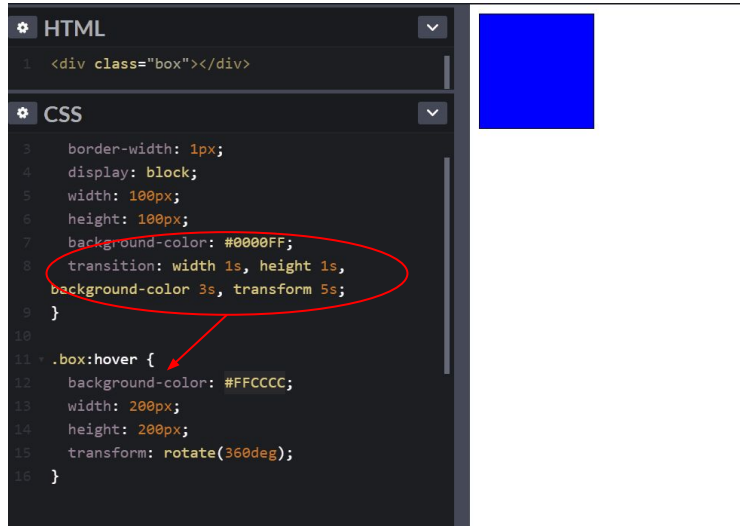
border-width: 1px;
display: block;
width: 100px;
height: 100px;
background-color: #0000FF;
transition: width 2s, height 2s,
background-color 2s, transform 2s;
}

.box:hover {
  background-color: #FFCCCC;
  width: 200px;
  height: 200px;
  transform: rotate(360deg);
}
```



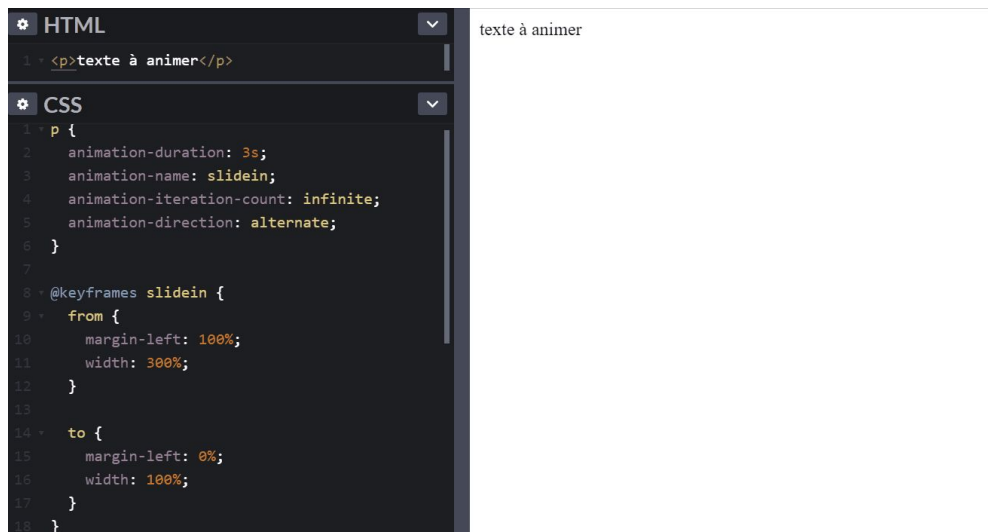
HTML & CSS

On voit la différence si on change les vitesses de transition. On retrouve bien les propriétés du :hover dans “transition”



HTML & CSS

En plus des “transitions”, le CSS prévoit la possibilité de faire des animations. Bon là, ça sert pas à grand chose...



HTML & CSS

“Animation-duration” définit le temps de l’animation

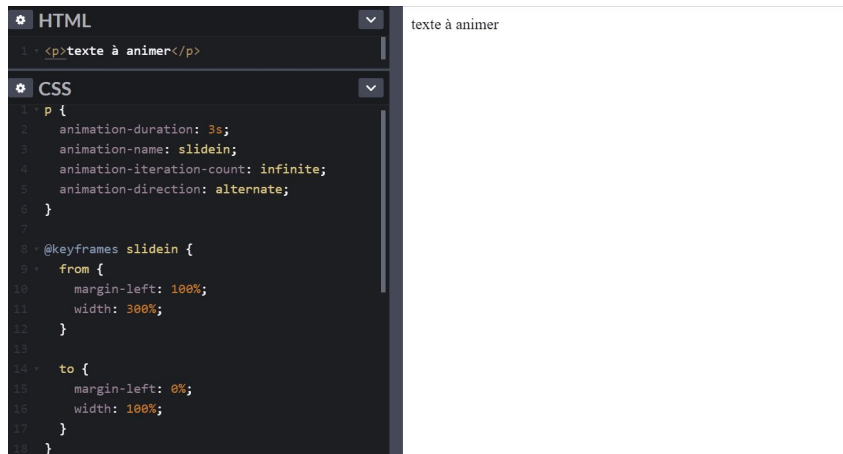
“Animation-name” c’est le nom de votre animation

“Animation-iteration-count” c’est le nombre de fois que va être joué e l’animation (si vous mettez “infinite”, elle s’arrêtera pas)

“Animation-direction” c’est si vous souhaitez un “aller-retour” (avec “alternate”)

“Animation-delay” c’est le temps que mettra l’animation avant de commencer.

“@keyframes nomDeL’animation{...}” Vous définissez ici le comportement de l’animation.



HTML & CSS

Il y a aussi la possibilité d'utiliser un raccourci pour tout écrire ;). Après 2 ou 3 tests, il s'avère que l'ordre est pas trop important.

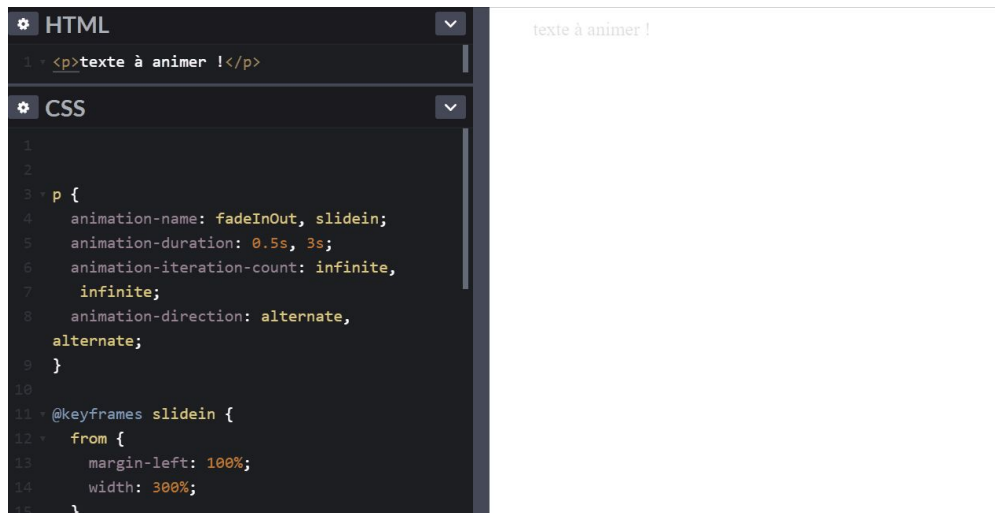
```
p {  
  animation-duration: 3s;  
  animation-name: slidein;  
  animation-iteration-count: infinite;  
  animation-direction: alternate;  
  animation-delay: 4s;  
}  
  
p {  
  animation: 3s infinite alternate slidein  
  5s;  
}
```

HTML & CSS

Il est également possible de cumuler plusieurs animations en même temps.

Animations : https://developer.mozilla.org/fr/docs/Web/CSS/CSS_Animations/Using_CSS_animations

Transitions : https://developer.mozilla.org/fr/docs/Web/CSS/CSS_Transitions/Using_CSS_transitions



HTML & CSS

Bon, sinon on peut aussi faire ça en CSS mais on va pas le faire....hein ?

