



Réaliser une interface utilisateur web statique et adaptable

Soupramanien BOUVANESVARY

Sommaire

- ▶ [Comprendre le rôle d'HTML](#)
- ▶ [Apprendre la syntaxe d'HTML5](#)
- ▶ [Ecrire du HTML sémantique](#)
- ▶ [Comprendre la relation entre HTML et CSS](#)
- ▶ [Apprendre la syntaxe et les sélecteurs de CSS3](#)
- ▶ [Méthodes pour appliquer des styles en HTML](#)
- ▶ [Utiliser CSS3 pour formater et mettre en forme une page web](#)
- ▶ [Responsive Web Design](#)
- ▶ [Tableau HTML](#)
- ▶ [Formulaire HTML](#)
- ▶ [Animations CSS3](#)
- ▶ [Boîtes flexibles CSS](#)
- ▶ [Référencement](#)
- ▶ [Optimisation pour le web mobile](#)



Comprendre le rôle d'HTML



Qu'est-ce qu'HTML ?

- ▶ HyperText Markup Language
- ▶ Inventé par Tim Berners-Lee
- ▶ N'est pas un langage de programmation comme Java ou C
- ▶ Est un langage à balise
- ▶ Spécifie la structure d'une page Web
- ▶ A des éléments (balises) pré-définis qu'on peut utiliser pour marquer le contenu
 - ▶ <p>, <h1>, <a>, , etc...

Qu'est-ce qu'HTML ?



H T M L

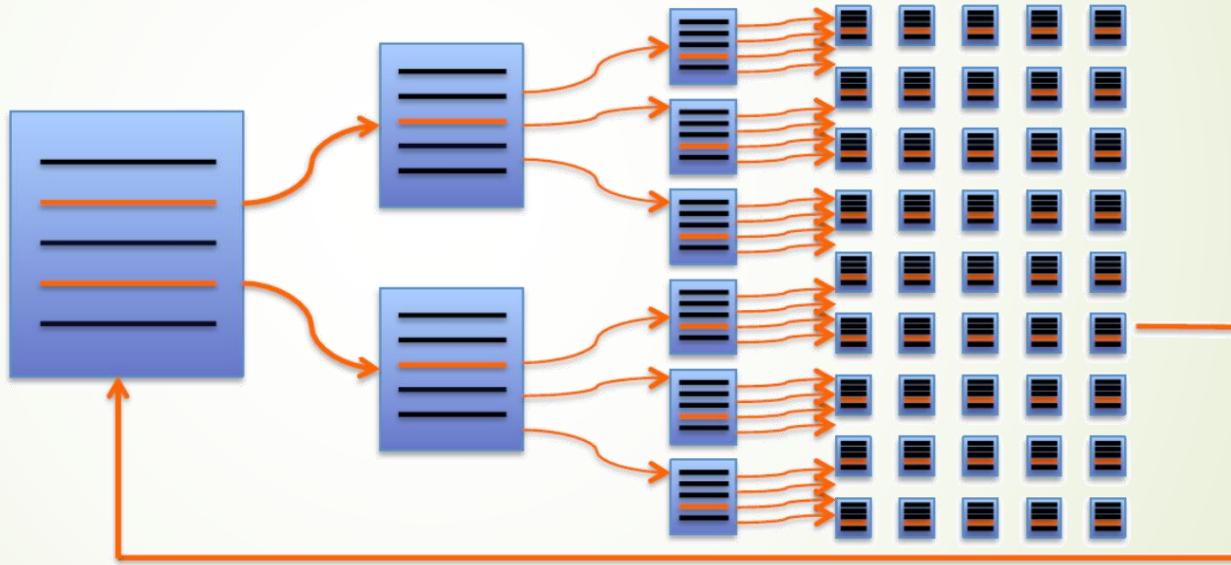
HyperText Markup Language

The word "HTML" is written in large, bold, orange letters at the top. Below it, the full name "HyperText Markup Language" is written in a larger font, where "HyperText" and "Language" are in blue and "Markup" is in orange. Orange lines connect the letters of "HTML" to the corresponding words in the full name.

Langage à balises de description de documents hypertextes

Qu'est-ce qu'HTML ?

Hypertext Markup Language



Hyper : Non linéaire, liens entre noeuds

Text : Composé de texte

Qu'est-ce qu'HTML ?

Hypertext **Markup** Language

```
<!doctype html>
<html>
<head>
  <title>Why I Love This Course</title>
</head>
<body> [ . . . ]
</body>
</html>
```

content

Markup : Marqué, balisé

Qu'est-ce qu'HTML ?

Hypertext Markup **Language**

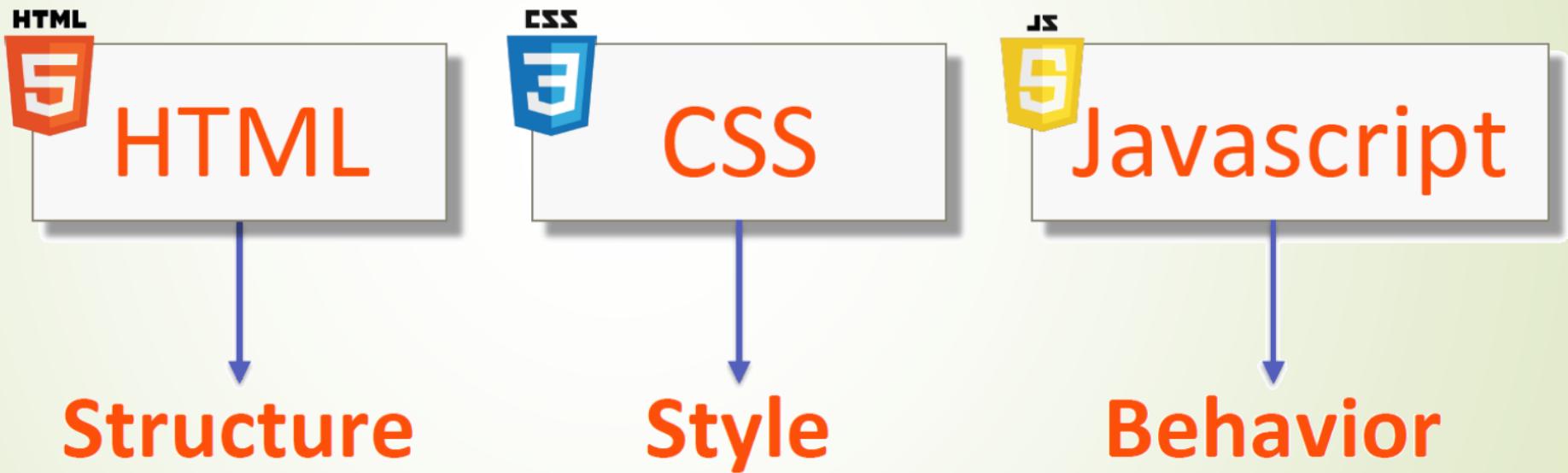
```
<h1>  
  <div>Hello World!</h1>  
</div>
```

```
<h1>  
  <div>Hello World!</div>  
</h1>
```

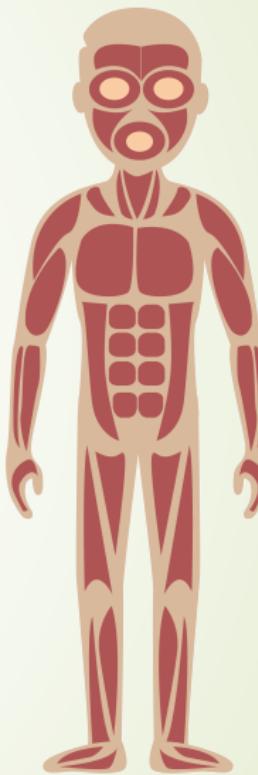
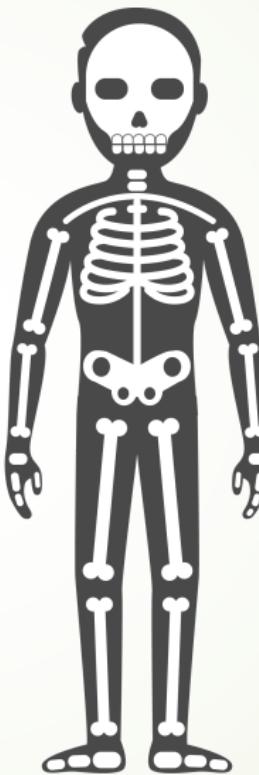
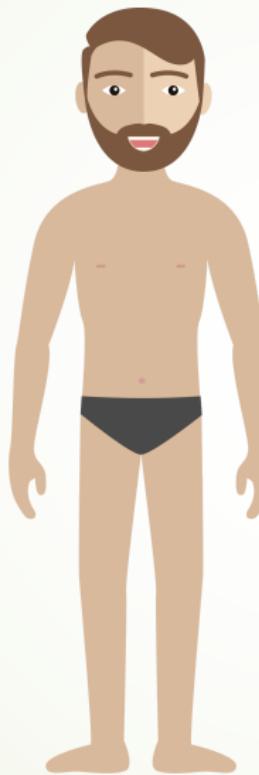


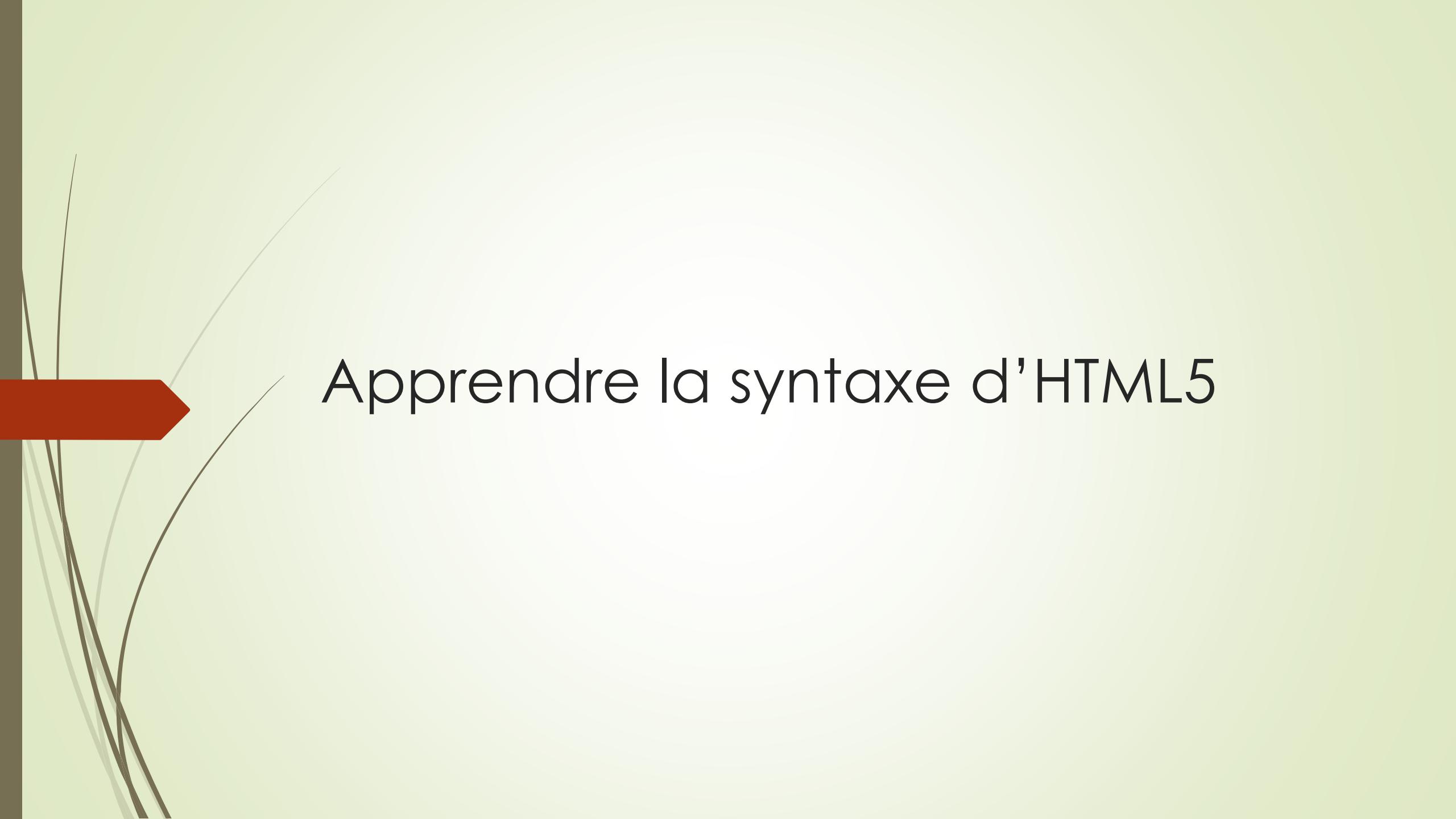
Language : Langage

Technologies qui pilotent le web



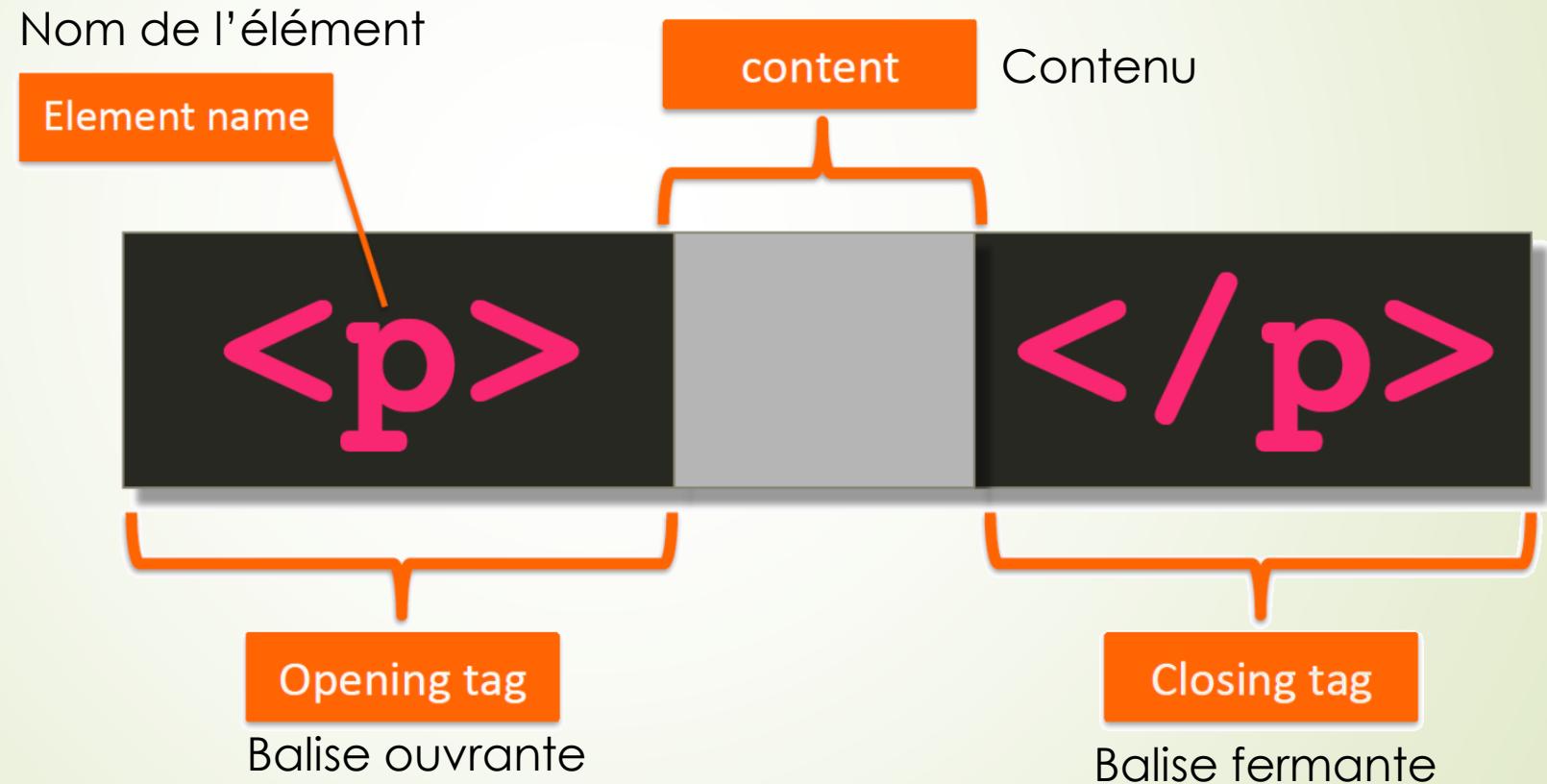
Technologies qui pilotent le web





Apprendre la syntaxe d'HTML5

Anatomie d'une balise HTML



Anatomie d'une balise HTML

Saut de ligne

Line Break

A diagram illustrating the structure of the `
` tag. It consists of a black rectangular box containing the pink text "`
`". A horizontal bracket below the box spans its width, and an orange callout box labeled "Only opening tag" is positioned at the bottom center of this bracket. Another orange callout box labeled "Line Break" is positioned above the top center of the black box.

Only opening tag

Uniquement la balise ouvrante

Règle horizontale

Horizontal Rule

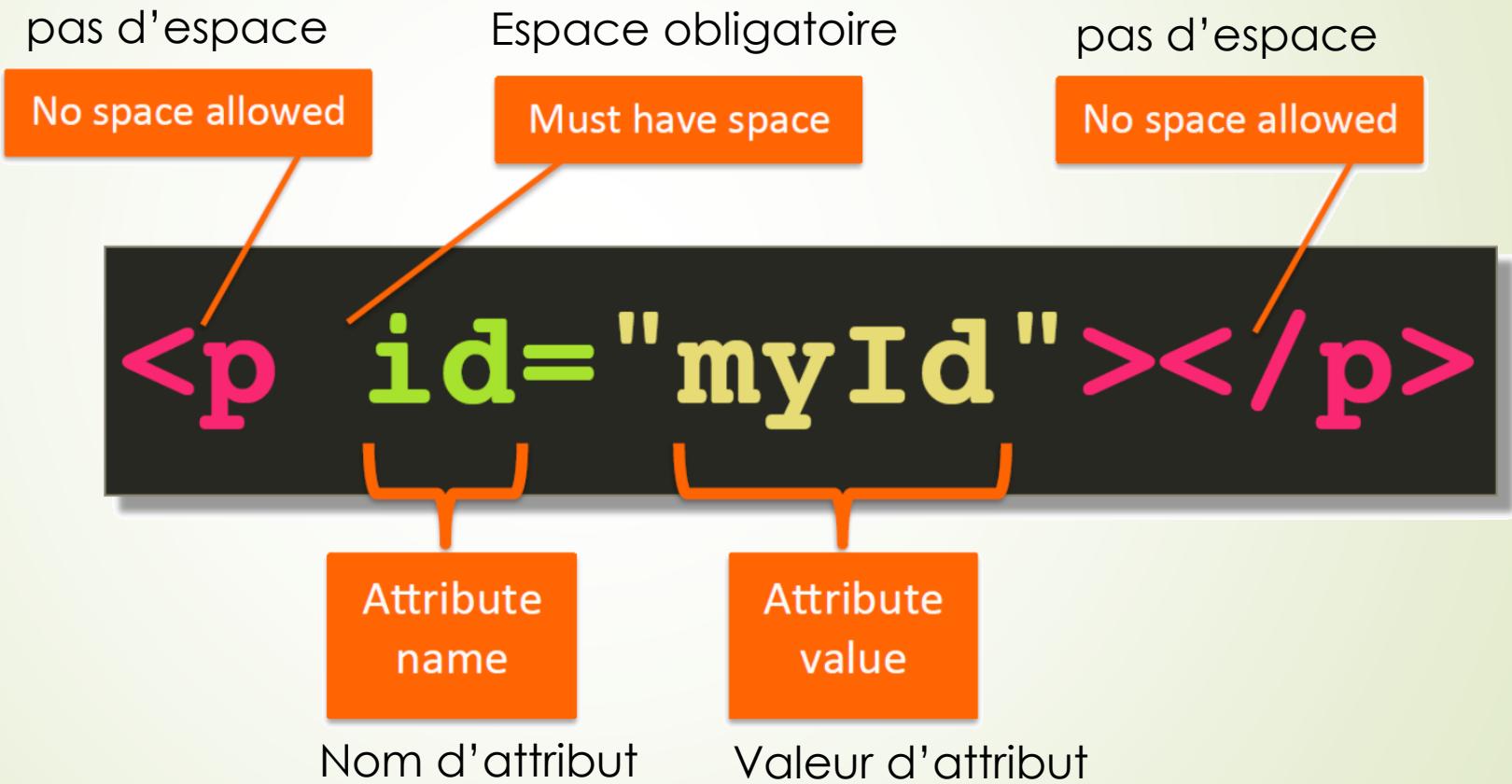
<hr>

A diagram illustrating the structure of the `<hr>` tag. It consists of a black rectangular box containing the pink text "`<hr>`". A horizontal bracket below the box spans its width, and an orange callout box labeled "Only opening tag" is positioned at the bottom center of this bracket. Another orange callout box labeled "Horizontal Rule" is positioned above the top center of the black box.

Only opening tag

Uniquement la balise ouvrante

Anatomie d'une balise HTML



Anatomie d'une balise HTML

```
<p onclick="alert('hi')"></p>
```

Guillemets double externe

Outer double quotes

Guillemets simple interne

Inner single
quotes

Anatomie d'une balise HTML

```
<p/>
```



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



Anatomie d'un document HTML

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>ma première page</title>
  </head>
  <body>
    <p>C'est ma première page web</p>
    <span>Hello</span>
    <span>World !</span>
  </body>
</html>
```

Balise <Head>

```
<head>
  <meta charset="utf-8">
  <title>My test page</title>
</head>
```

- ▶ Contenu n'est pas affiché sur la page
- ▶ Contient metadata (métadonnée) du document HTML
 - ▶ Une **métadonnée** est une donnée qui décrit une donnée. Ici, la description du document HTML
- ▶ Élément <title> est utilisé pour ajouter un titre au document
 - ▶ affiché dans la barre de titre du navigateur ou dans l'onglet de la page
 - ▶ Est utilisé pour décrire la page quand on l'ajoute dans favori.

Metadata : balise <meta>

- ▶ Une **métadonnée** est une donnée qui décrit une donnée
 - ▶ <meta charset="utf-8">
 - ▶ déclare l'encodage utilisé par la page
 - ▶ Plusieurs éléments <meta> incluent les attributs "name" et "content"
 - ▶ **Name** définit le nom d'un métadonnée au niveau du document
 - ▶ **Content** fournit la valeur associée avec l'attribut http-equiv ou l'attribut name suivant le contexte utilisé

```
<meta name="author" content="Soupramanien BOUVANESVARY">
<meta name="description" content="The MDN Web Docs Learning Area aims to
provide complete beginners to the Web with all they need to know to get
started with developing web sites and applications.">
```

- ▶ Pour plus d'info : [consulter cette page](#)

Autres éléments dans <head>

- ▶ Ajout d'icon personnalisé à votre site

```
<link rel="shortcut icon" href="favicon.ico" type="image/x-icon">
```

- ▶ Ici, on ajoute favicon (favorites icon)

- ▶ Liaison de JavaScript et CSS à HTML

```
<link rel="stylesheet" href="my-css-file.css">
```

```
<script src="my-js-file.js"></script>
```

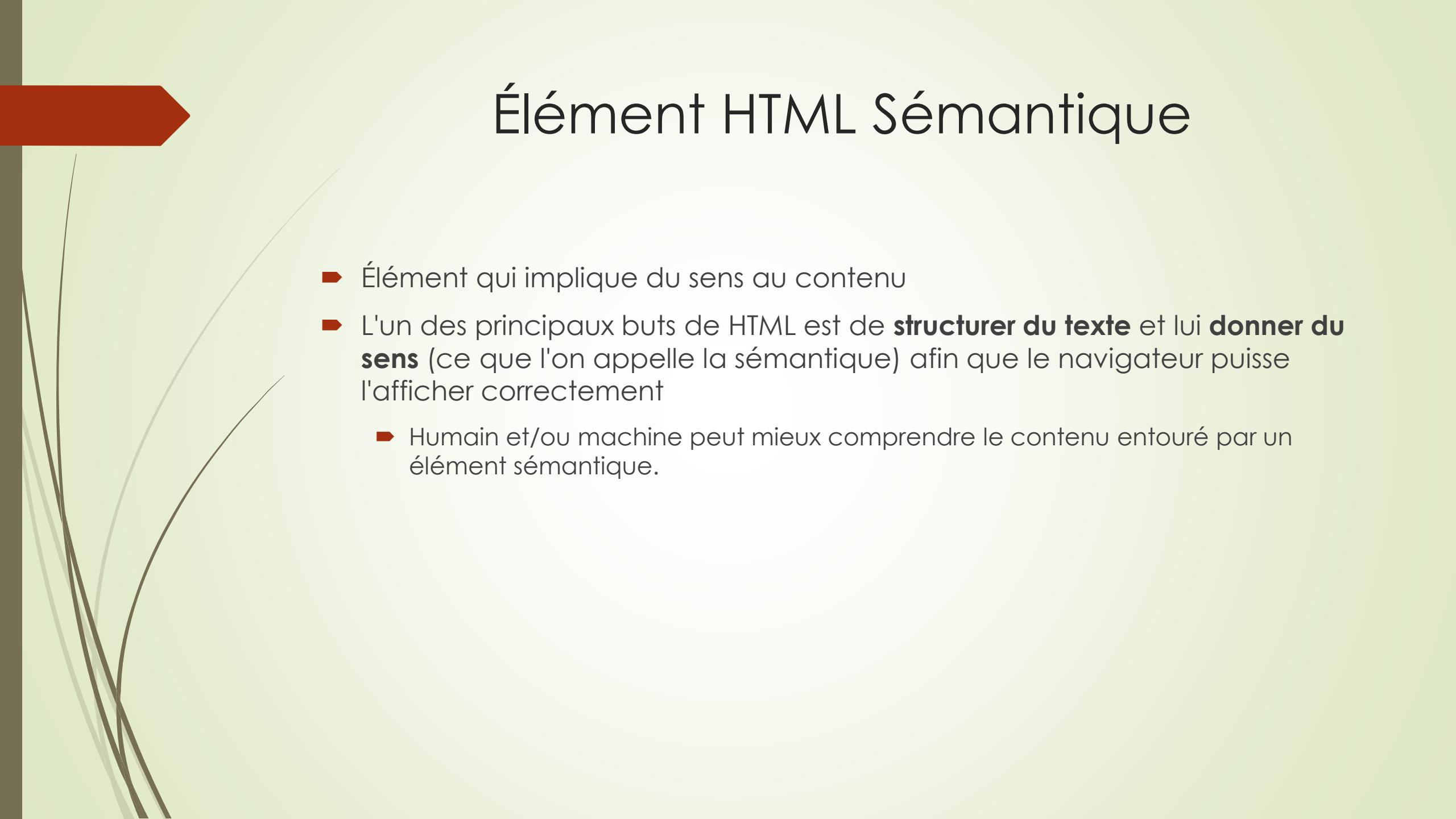
Entité HTML : inclure les caractères spéciaux dans HTML

- ▶ <, > et & sont des caractères réservés
- ▶ On doit toujours échapper ces caractères dans notre documents HTML

Character	Description	Entity Name	Entity Number
©	Copyright symbol	©	©
®	Registered trademark	®	®
€	Euro	€	€
&	Ampersand	&	&
<	Less than	<	<
>	Greater than	>	>
	Nonbreaking space	 	



Ecrire du HTML sémantique



Élément HTML Sémantique

- ▶ Élément qui implique du sens au contenu
- ▶ L'un des principaux buts de HTML est de **structurer du texte** et lui **donner du sens** (ce que l'on appelle la sémantique) afin que le navigateur puisse l'afficher correctement
 - ▶ Humain et/ou machine peut mieux comprendre le contenu entouré par un élément sémantique.

Racine de sectionnement <body>

- ▶ L'élément <body> représente le **contenu principal** du document HTML.
- ▶ Il ne peut y avoir qu'un élément <body> par document.



Sectionnement du contenu

- ▶ Organiser le contenu d'une page en différentes **sections** permet d'avoir une structure logique au sein d'un document.
- ▶ Grâce à ces éléments
 - ▶ on peut créer un plan pour la page
 - ▶ ajouter des titres pour identifier les sections et
 - ▶ également gérer un en-tête et un bas de page.

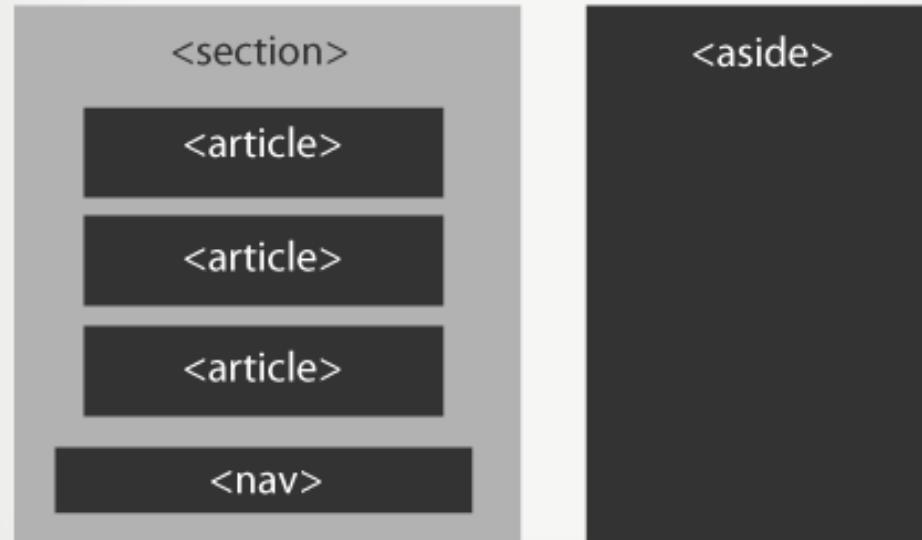
Sectionnement du contenu

Nom	Détails
<code><article></code>	Portion de contenu indépendante, se suffisant en termes de compréhension (pouvant être syndiquée dans un flux RSS). Exemples : article de presse, fiche cinéma, réponse de forum, commentaire d'article
<code><aside></code>	Contenu indirectement lié au contenu principal, non nécessaire à sa compréhension : définitions, digressions, sidebar, compléments...
<code><section></code>	Élément générique pour une section de contenu ou d'application web, utilisé à défaut d'un autre élément de section plus sémantique tel que <code><article></code> , <code><nav></code> , <code><header></code> , <code><footer></code> , <code><aside></code> . À ne pas confondre avec <code><div></code> qui n'a aucune valeur sémantique. En général, <code><section></code> contient un titre qui la définit.
<code><nav></code>	Regroupe la sélection des principaux liens pour naviguer dans le site, l'application ou le document.
<code><header></code>	S'applique au document entier, mais aussi à toute section au sens large, <code><article></code> y compris.
<code><footer></code>	S'applique au document entier, mais aussi à toute section au sens large, <code><article></code> y compris.



`<header>`

`<nav>`



`<section>`

`<article>`

`<article>`

`<article>`

`<aside>`

`<footer>`

Éléments <h1> à <h6>

- ▶ Les éléments <h1> à <h6> représentent six **niveaux de titres** dans un document
- ▶ <h1> est le plus important et <h6> est le moins important
- ▶ Un élément de titre décrit brièvement le sujet de la section qu'il introduit
 - ▶ L'information d'un titre peut être utilisée par les agents utilisateurs, par exemple, pour construire automatiquement une table des matières d'un document.
 - ▶ Les titres ne doivent pas être utilisé afin de réduire ou d'augmenter la taille de la police d'un texte : il faut pour cela utiliser la propriété CSS font-size à la place.
 - ▶ On évitera de sauter des niveaux de titre : on commence toujours par <h1> puis <h2> et ainsi de suite.
 - ▶ Il est préférable d'éviter d'utiliser plus d'un élément <h1> sur une même page.
- ▶ L'élément <hgroup> permet de regrouper le titre principal d'une section avec son (ou ses) sous-titre(s) afin d'obtenir un titre sur plusieurs niveaux sémantiques.



Contenu textuel

- ▶ Le contenu HTML textuel permet d'organiser des blocs ou des sections de contenu entre la balise ouvrante `<body>` et la balise fermante `</body>`.
- ▶ Ces éléments sont cruciaux pour **l'accessibilité** et le **référencement** car ils permettent d'identifier le sens du contenu.

Éléments de groupement de contenu

► Paragraphe

- ▶ L'élément HTML `<p>` représente un **paragraphe de texte**.
- ▶ Les paragraphes sont généralement représentés comme des blocs et séparés par un espace vertical, leur première ligne est également parfois indentée.
- ▶ Les paragraphes sont des éléments blocs.

```
<p>Premier paragraphe du texte. J'aime les  
licornes beaucoup beaucoup beaucoup.</p>
```

```
<p>Deuxième paragraphe du texte. Et si j'en  
avais une apprivoisée je serais très  
contente.</p>
```

Éléments de groupement de contenu

► Règle horizontale

- ▶ L'élément HTML `<hr>` représente un **changement thématique** entre des éléments de paragraphe
- ▶ Exemples
 - ▶ un changement de décor dans un récit
 - ▶ un changement de sujet au sein d'une section

```
<p>paragraph 1</p>
<hr />
<p>paragraph 2</p>
```

Éléments de groupement de contenu

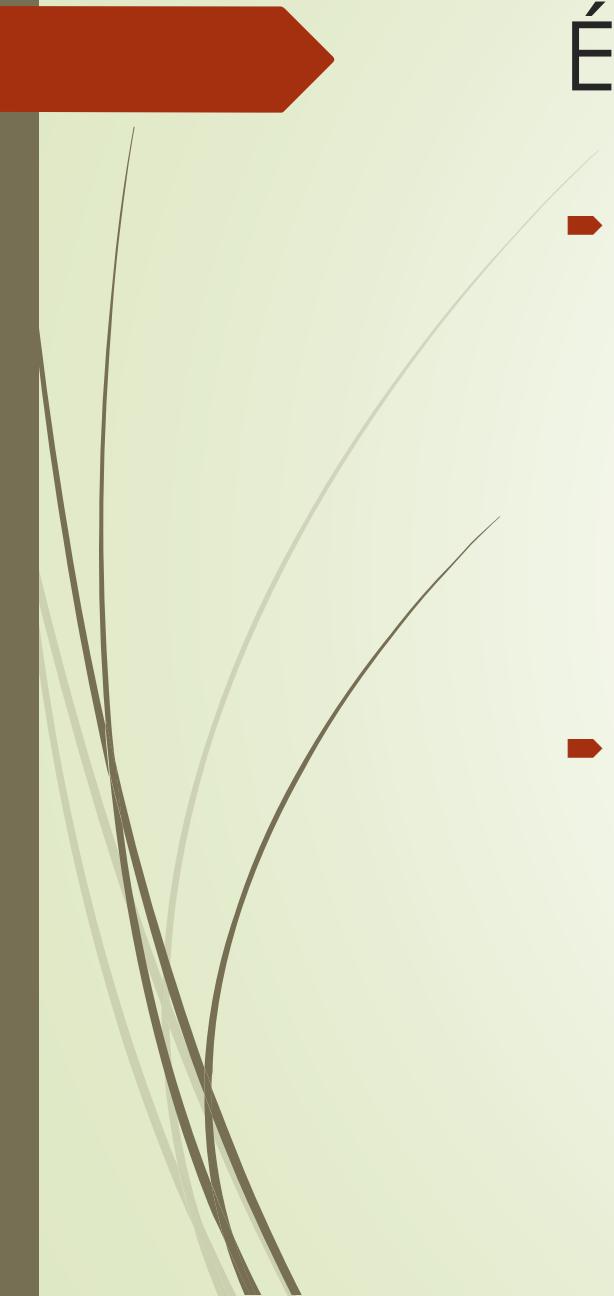
► Texte préformaté

- ▶ L'élément HTML `<pre>` représente du texte préformaté, généralement écrit avec une **police à chasse fixe**.
- ▶ Le texte est affiché tel quel, les espaces utilisés dans le document HTML seront retranscrits
- ▶ Il est nécessaire d'échapper les caractères '`<`' en '`<`' afin de s'assurer que le code écrit entre les éléments ne soit pas interprété de façon involontaire.

```
<pre>
body {
    color:red;
}
</pre>
```



```
body {
    color:red;
}
```



Éléments de groupement de contenu

► **main**

- ▶ L'élément HTML <main> représente le **contenu majoritaire** du <body> du document.
- ▶ Le contenu principal de la zone est constitué de contenu directement en relation, ou qui étend le sujet principal du document ou de la fonctionnalité principale d'une application.
- ▶ Un document ne peut pas avoir plus d'un seul élément <main> sans attribut hidden.

► **division**

- ▶ L'élément HTML <div> (qui signifie **division du document**) est un conteneur générique qui permet d'organiser le contenu sans représenter rien de particulier.
- ▶ Il peut être utilisé afin de grouper d'autres éléments pour leur appliquer un style (en utilisant les attributs class ou id) ou parce qu'ils partagent des attributs aux valeurs communes, tel que lang.
- ▶ Il doit uniquement être utilisé lorsqu'aucun autre élément sémantique (par exemple <article> ou <nav>) n'est approprié.



Éléments de groupement de contenu

► **blockquote / cite**

- ▶ L'élément <blockquote> (qui signifie **bloc de citation**) indique que le texte contenu dans l'élément est une citation longue.
- ▶ Le texte est généralement affiché avec une indentation
- ▶ Une URL indiquant la source de la citation peut être donnée grâce à l'attribut « cite » tandis qu'un texte représentant la source peut être donné via l'élément <cite>.



```
<blockquote  
cite="https://www.huxley.net/bnw/four.html">  
<p>Words can be like X-rays, if you use them  
properly—they'll go through anything. You read  
and you're pierced.</p>  
<footer>—Aldous Huxley, <cite>Brave New  
World</cite></footer>  
</blockquote>
```



“Words can be like
X-rays, if you use
them properly—
they’ll go through
anything. You read
and you’re pierced.”

—Aldous Huxley, *Brave
New World*

```
blockquote {  
    margin: 0;  
}  
  
blockquote p {  
    padding: 15px;  
    background: #eee;  
    border-radius:  
5px;  
}  
  
blockquote p::before {  
    content: '\201C';  
}  
  
blockquote p::after {  
    content: '\201D';  
}
```

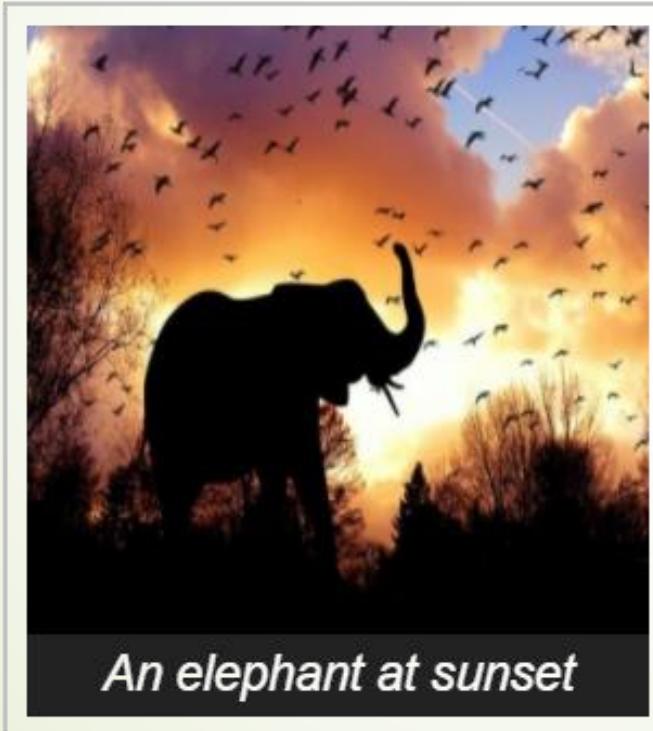


Éléments de groupement de contenu

► **figure**

- ▶ L'élément HTML <figure> représente une figure (un **schéma**), qui peut être accompagné d'une **légende** grâce à l'élément <figcaption>.
- ▶ Il est normalement référencé de manière unique.
- ▶ C'est souvent une image, une illustration, un diagramme, un fragment de code ou un schéma auquel il est fait référence dans le texte principal mais qui peut être utilisé sur une autre page ou dans une annexe sans que cela affecte le contenu principal.

```
<figure>
  
  <figcaption>An elephant at sunset</figcaption>
</figure>
```



```
figure {
  border: thin #c0c0c0 solid;
  display: flex;
  flex-flow: column;
  padding: 5px;
  max-width: 220px;
  margin: auto;
}

img {
  max-width: 220px;
  max-height: 150px;
}

figcaption {
  background-color: #222;
  color: #fff;
  font: italic smaller sans-serif;
  padding: 3px;
  text-align: center;
}
```



Éléments de groupement de contenu

► **details / summary**

- ▶ L'élément HTML `<details>` est utilisé comme un outil permettant de révéler une information. Un résumé ou un intitulé peuvent être fournis grâce à un élément `<summary>`.
- ▶ La plupart du temps, le contrôle utilisé pour cet élément est un triangle qui est tourné ou tordu afin d'indiquer si l'élément est révélé ou non. Si le premier élément fils de l'élément `<details>` est un élément `<summary>`, c'est le contenu de ce dernier qui est utilisé comme intitulé pour le contenu à révéler (l'intitulé est donc toujours visible).

```
<details>
  <summary>Details</summary>
  Something small enough to escape
  casual notice.
</details>
```



▼ Details

Something small enough to escape
casual notice.

Éléments de groupement de contenu

► Listes de définitions

- ▶ **<dl>** Liste de définitions : Associe un « terme à définir » (**<dt>**) et des « définitions » (**<dd>**) s'y rapportant. Contenu autorisé : succession de **<dt>** et **<dd>**.
- ▶ **<dt>** Terme/titre de définition : Enfant d'une liste **<dl>**, balisant le terme à définir. Contenu autorisé : contenus textuels y compris des balises de niveau intra-paragraphe chargées de contenu.
- ▶ **<dd>** Contenu de la définition : Enfant d'une liste **<dl>**, recueillant la définition du terme **<dt>** qui le précède immédiatement.

```
<dl>
  <dt>Beast of Bodmin</dt>
  <dd>A large feline inhabiting Bodmin Moor.</dd>
  <dt>Morgawr</dt>
  <dd>A sea serpent.</dd>
  <dt>Owlman</dt>
  <dd>A giant owl-like creature.</dd>
</dl>
```

The diagram illustrates the grouping of content using the `<dl>` element. On the left, a grey box contains the HTML code for a definition list. On the right, three terms are listed with their respective definitions, connected by arrows indicating the mapping from the list items to the terms. A large red arrow points from the entire `<dl>` block to the first term, **Beast of Bodmin**. Below it, two smaller red arrows point from the `<dt>` and `<dd>` pairs to the second term, **Morgawr**, and the third term, **Owlman** respectively.

Beast of Bodmin
A large feline inhabiting Bodmin Moor.

Morgawr
A sea serpent.

Owlman
A giant owl-like creature.

Éléments de groupement de contenu

► Liste ordonnée

- ▶ L'élément HTML `` représente une liste ordonnée.
- ▶ Les éléments d'une telle liste sont généralement affichés avec un **indicateur ordinal** pouvant prendre la forme de nombres, de lettres, de chiffres romains ou de points.
- ▶ La mise en forme de la numérotation n'est pas utilisée dans la description HTML mais dans la feuille de style CSS associée grâce à la propriété `list-style-type`.
- ▶ Il doit être utilisé pour regrouper plusieurs éléments qui ont une relation d'ordre.

```
<ol>
  <li>Premier élément</li>
  <li>Deuxième élément</li>
  <li>Troisième élément</li>
</ol>
```



1. Premier élément
2. Deuxième élément
3. Troisième élément

Éléments de groupement de contenu

► Liste sans ordre

- ▶ L'élément HTML `` représente une liste d'éléments **sans ordre particulier**.
- ▶ Il est souvent représenté par une liste à puces.
- ▶ Il doit être utilisé pour regrouper plusieurs éléments qui n'ont pas de relation d'ordre.

```
<ul>
  <li>1 artichaut</li>
  <li>De l'essuie-tout</li>
  <li>200g de chocolat</li>
</ul>
```



- 1 artichaut
- De l'essuie-tout
- 200g de chocolat

Éléments de groupement de contenu

```
<h2>Book Topics</h2>
<ul>
    <li>HTML</li>
    <li>CSS</li>
    <li>JavaScript</li>
</ul>
<h2>HTML Chapters</h2>
<ol start="4">
    <li>Structural Elements</li>
    <li>Text Elements</li>
    <li>Table Elements</li>
    <li>Embedded Elements</li>
    <li>Form Elements</li>
</ol>
```

Book Topics

- HTML
- CSS
- JavaScript

HTML Chapters

4. Structural Elements
5. Text Elements
6. Table Elements
7. Embedded Elements
8. Form Elements

L'élément **** dispose les attributs suivants :

- **reversed** : Cet attribut booléen précise que les objets listés par l'élément sont affichés avec un ordre inversé
- **start** : La valeur de cet attribut, un nombre entier, définit l'indice à partir duquel les objets de la liste sont numérotés.

Sémantique du texte en ligne

- ▶ Les éléments pour le texte en ligne peuvent être utilisés pour définir la signification, la structure ou la mise en forme d'un terme, d'une ligne ou d'un fragment de texte.

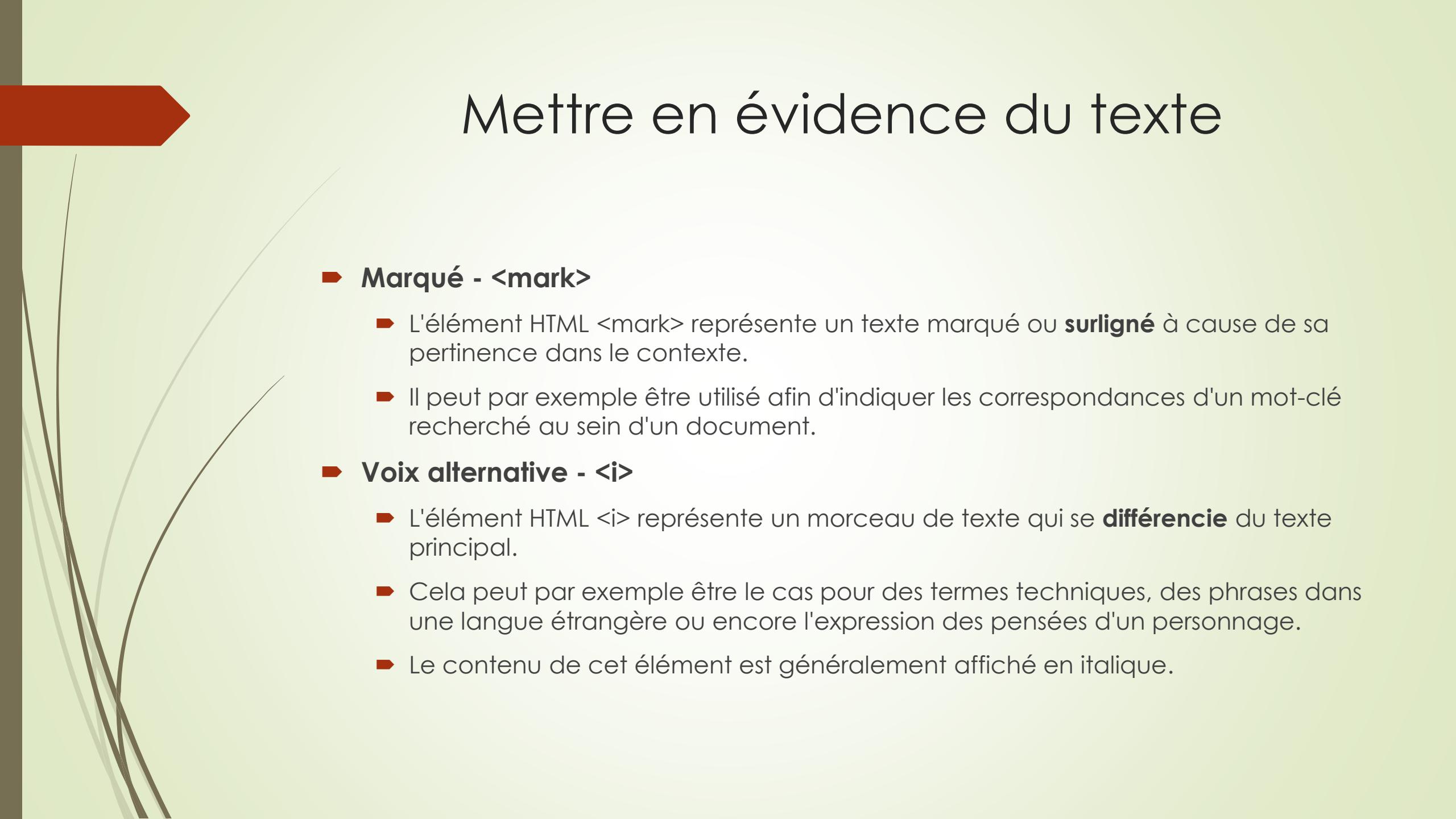
Mettre en évidence du texte

► Importance -

- ▶ L'élément HTML indique que le texte a une **importance** particulière ou un certain sérieux voire un caractère urgent.
- ▶ Cela se traduit généralement par un affichage en gras.

► Emphase -

- ▶ L'élément HTML (pour emphase) est utilisé afin de marquer un texte sur lequel on veut **insister**.
- ▶ Les éléments peuvent être imbriqués, chaque degré d'imbrication indiquant un degré d'insistance plus élevé



Mettre en évidence du texte

► Marqué - <mark>

- ▶ L'élément HTML <mark> représente un texte marqué ou **surligné** à cause de sa pertinence dans le contexte.
- ▶ Il peut par exemple être utilisé afin d'indiquer les correspondances d'un mot-clé recherché au sein d'un document.

► Voix alternative - <i>

- ▶ L'élément HTML <i> représente un morceau de texte qui se **différencie** du texte principal.
- ▶ Cela peut par exemple être le cas pour des termes techniques, des phrases dans une langue étrangère ou encore l'expression des pensées d'un personnage.
- ▶ Le contenu de cet élément est généralement affiché en italique.

Mettre en évidence du texte

► Petit - <small>

- L'élément HTML <small> permet de représenter des commentaires ou des textes à écrire en **petits caractères** (des termes d'un contrat, des mentions relatives au droit d'auteur, etc.) quelle que soit la présentation.

► Barré - <s>

- L'élément HTML <s> permet d'afficher du texte qui est **barré** car il n'est plus pertinent ou car il est obsolète.
- <s> ne doit pas être employé pour indiquer des éditions dans un document (on utilisera alors et <ins>).

►

- L'élément permet d'attirer l'attention du lecteur sur le contenu d'un élément sans que ce contenu revêt une importance particulière

Mettre en évidence du texte

► <u>

- ▶ L'élément HTML <u> permet d'afficher un fragment de texte qui est annoté avec des éléments non textuels.
- ▶ Par défaut, le contenu de l'élément est **souligné**.
- ▶ Cela pourra par exemple être le cas pour marquer un texte comme étant un nom propre chinois, ou pour marquer un texte qui a été mal orthographié.

Exemples

<p>

"Read my lips: <mark>no new taxes</mark>", declared presidential candidate George H. W. Bush in 1988. However, the 1990 budget agreement increased taxes in several areas.
</p>

"Read my lips: **no new taxes**". declared presidential candidate George H. W. Bush in 1988. However, the 1990 budget agreement increased taxes in several areas.

<p>

"Class, pay attention!" *I wonder if they're even listening to me.* "Who's ready for tomorrow's exam?"
</p>

"Class, pay attention!" *I wonder if they're even listening to me.* "Who's ready for tomorrow's exam?"

<p>I am glad you weren't late. </p>

I am *glad* you weren't *late*.

<p>This liquid is highly toxic. </p>

This liquid is **highly toxic**.

Date et heure

- ▶ Balise attribuant une signification standardisée à une date (et heure si nécessaire), compréhensible par un robot ou un programme, via l'attribut **datetime**.
- ▶ La présence de l'attribut **pubdate** signifie que la valeur de temps vaut pour la date de publication du plus proche ancêtre `<article>`.

```
<article>
  <p>Publié le <time datetime="2012-04-01T13:37:00Z">1er avril 2012</time></p>
</article>
```

Abréviations

► <abbr>

- L'élément <abbr> représente une **abréviation** ou un **acronyme** et permet, de façon optionnelle, d'en fournir une description complète.
- S'il est présent, l'attribut title doit contenir cette même description complète et rien d'autre.

```
<p>  
The use of <abbr title="Hypertext Markup Language">HTML</abbr> has  
contributed greatly to the popularity of web-based applications. <br />  
</p>
```

The use of HTML has contributed greatly to the popularity of web-based applications.

Hypertext Markup Language

Subscripts et Superscripts

► <sub>

- L'élément HTML <sub> est utilisé, pour des raisons typographiques, afin d'afficher du texte souscrit (ou en **indice**) (plus bas et généralement plus petit) par rapport au bloc de texte environnant.

► <sup>

- L'élément HTML <sup> est utilisé, pour des raisons typographiques, afin d'afficher du texte en **exposant** (plus haut et généralement plus petit) par rapport au bloc de texte environnant.

```
<p>
H<sub>2</sub>O is the chemical formula for water.<br />
e=mc<sup>2</sup> is the formula for mass-energy equivalence.
</p>
```

H_2O is the chemical formula for water.

$e=mc^2$ is the formula for mass-energy equivalence.

Span et saut de ligne

► Span

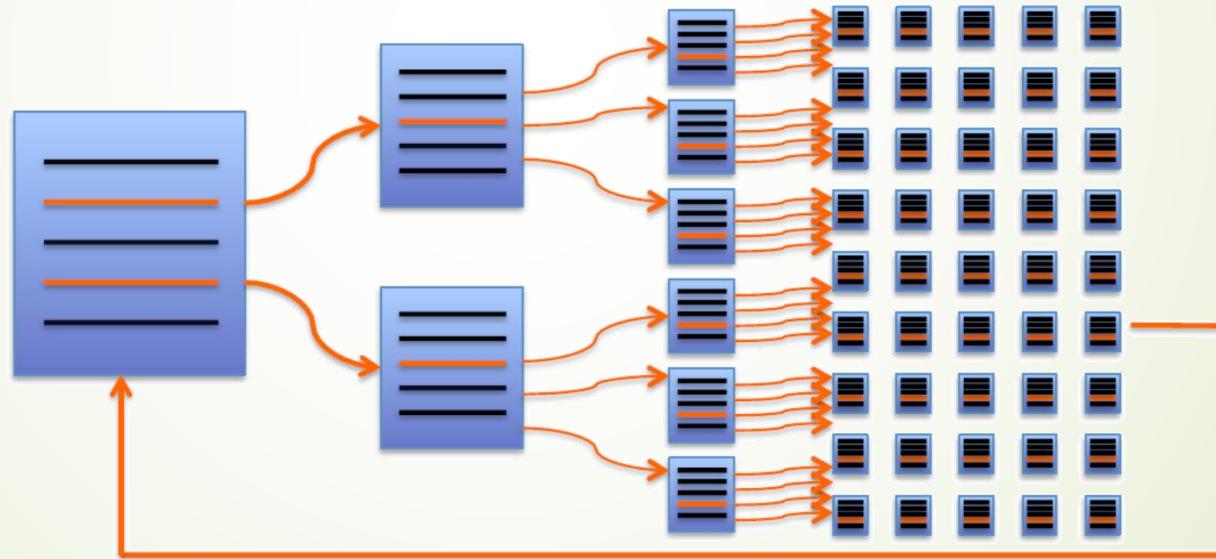
- ▶ L'élément HTML `` est un conteneur générique en ligne (inline) pour les contenus phrasés.
- ▶ Il ne représente rien de particulier.
- ▶ Il peut être utilisé pour **grouper des éléments** afin de les mettre en forme (grâce aux attributs `class` ou `id` et aux règles CSS) ou parce qu'ils partagent certaines valeurs d'attribut comme `lang`.
- ▶ Il doit uniquement être utilisé lorsqu'aucun autre élément sémantique n'est approprié.
- ▶ `` est très proche de l'élément `<div>`, mais l'élément `<div>` est un élément de bloc, alors que `` est un élément en ligne.

► Saut de ligne

- ▶ L'élément HTML `
` crée un saut de ligne (un **retour chariot**) dans le texte.
- ▶ Il s'avère utile lorsque les sauts de ligne ont une importance (par exemple lorsqu'on écrit une adresse ou un poème).

Hyperliens

- ▶ **Hyperliens** sont très importants — ce sont eux qui forment le web (la toile)



Ancre (a)

- ▶ L'élément <a> (pour ancre ou anchor en anglais) définit un **hyperlien vers un autre endroit** de la même page ou vers une autre page sur le Web.
- ▶ Le contenu dans l'élément ancre peut être de type bloc ou inline.
- ▶ Il supporte l'attribut **href**,
 - ▶ qui indique la cible du lien sous la forme d'une URL ou d'un fragment d'URL
 - ▶ Un fragment d'URL est un nom précédé par un dièse (#), qui indique une cible interne au document (un ID)
 - ▶ Les URL ne se limitent pas au documents web sur HTTP.
 - ▶ Les URL peuvent utiliser n'importe quel protocole supporté par le navigateur.
 - ▶ Par exemple, file, ftp et mailto fonctionnent avec la plupart des agents-utilisateur
 - ▶ http: - ressource web
 - ▶ ftp: - transfère de fichier
 - ▶ mailto: - envoie un email
 - ▶ tel: - compose un numéro de téléphone (pour les téléphones mobiles)
 - ▶ file: - ouvre un fichier

Ancre (a)

- ▶ L'attribut **target** indique où afficher la ressource liée.
- ▶ Il s'agit du nom, ou du mot-clé, d'un contexte de navigation (par exemple, un onglet, une fenêtre, ou une <iframe>).
- ▶ Les mots-clés suivants ont un sens particulier :
 - ▶ **_self** :
 - ▶ charge la réponse dans le contexte de navigation courant.
 - ▶ Il s'agit de la valeur par défaut quand l'attribut n'est pas renseigné.
 - ▶ **_blank** :
 - ▶ charge la réponse dans un nouveau contexte de navigation.
 - ▶ Ajouter un lien vers une autre page en utilisant target="_blank" exécutera la page dans le même processus que la page courante.
 - ▶ Si la nouvelle page consomme de nombreuses ressources, les performances de la page courante pourront en pâtir.
 - ▶ Pour éviter ce problème, on pourra utiliser rel="noopener".

Ancre (a)

- ▶ L'attribut **download** indique au navigateur de télécharger l'objet vers lequel pointe l'URL plutôt que d'y diriger l'utilisateur.
- ▶ Cela ouvre une invite pour enregistrer la cible du lien comme un fichier local.
- ▶ Si l'attribut a une valeur, le navigateur doit l'interpréter comme le nom de fichier par défaut pour l'invite qui s'ouvre. Ex: download="MyPicture.png"
- ▶ Les autres attributs fournissent uniquement les informations sémantiques :
 - ▶ **hreflang** - Cet attribut indique la langue de la ressource liée
 - ▶ **rel** - Cet attribut indique la relation entre la cible du lien et l'objet faisant le lien
 - ▶ **type** - Cet attribut indique le type de média de la cible du lien, sous la forme d'un type MIME



Exemples

```
<p>I'm creating a link to  
<a href="https://www.mozilla.org/en-US/"  
    title="The best place to find more information about Mozilla's  
    mission and how to contribute">the Mozilla homepage</a>.  
</p>
```

```
<p>I'm creating a link to  
<a href="https://developer.mozilla.org/en-US/docs/Web#Scripting"  
    title="The best place to learn about scripting"> Scripting</a>.  
</p>
```

Images

- ▶ L'élément HTML permet de représenter une **image** dans un document.
- ▶ C'est un élément vide (ce qui signifie qu'il ne contient ni texte ni balise de fermeture)
- ▶ Exemple :
 - ▶ ``



Images

- ▶ L'attribut **src** est obligatoire et contient le chemin de l'image qu'on souhaite afficher.
- ▶ L'attribut **alt** n'est pas obligatoire mais recommandé et contient une description textuelle de l'image ; il est recommandé pour des raisons d'accessibilité et sera utilisé par les lecteurs d'écran ou sera affiché si l'image ne peut pas être chargée.
 - ▶ Cet attribut définit le texte alternatif utilisé lorsqu'il est impossible d'afficher l'image (par exemple si l'URL est incorrecte ou si l'image n'est pas encore téléchargée)
- ▶ Le standard HTML ne fournit pas de liste exhaustive des formats que doit prendre en charge un agent utilisateur et chaque agent utilisateur couvre différents formats.
- ▶ Un [guide à propos des formats d'image pris en charge par les navigateurs web](#) est disponible
- ▶ JPEG, GIF et PNG sont supportés par tous les navigateurs.



Images

- ▶ L'attribut **height** indique la hauteur intrinsèque de l'image exprimée en pixels
- ▶ L'attribut **width** indique la largeur intrinsèque de l'image, exprimée en pixels
 - ▶ seules les valeurs exprimées en pixels sont acceptées
- ▶ Ces attributs sont utilisés pour s'assurer que l'image occupe un espace stable, y compris lors du chargement

Vidéo

- Avant, une **vidéo** c'était ça :

```
<object width="425" height="344">
  <param name="movie"
        value="http://www.youtube.com/v/9sEI1AUFKw&hl=en_GB&fs=1">
  </param>
  <param name="allowFullScreen" value="true"></param>
  <param name="allowscriptaccess" value="always"></param>

  <embed src="http://www.youtube.com/v/9sEI1AUFKw&hl=en_GB&fs=1"
         type="application/x-shockwave-flash"
         allowscriptaccess="always" allowfullscreen="true"
         width="425" height="344">
  </embed>
</object>
```

Vidéo : principe de base

- ▶

```
<video width="320" height="240" controls>
  <source src="bunny.mp4" type="video/mp4" />
  <source src="bunny.ogg" type="video/ogg" />
  Votre navigateur ne supporte pas le tag video.
</video>
```
- ▶ L'attribut **controls** ajoute les boutons de contrôle (play, stop, etc.)
- ▶ Le premier format supporté sera lu
- ▶ **<track>** via l'attribut **kind** lie un fichier de piste texte : sous-titres, légendes, audiodescription, chapitrage ou métainformations



<https://www.bigbuckbunny.org/>

Vidéo : les formats

- ▶ Il existe une grande bataille d'influence autour de cette balise :
 - ▶ Flash vs HTML5
 - ▶ WebM (Google) vs H.264 (Apple, Microsoft) vs Ogg (Firefox), etc.
 - ▶ Codecs, brevets, contrôles, etc.
- ▶ Une des solutions consiste à transcoder la vidéo dans un des formats supportés par le navigateur client

Navigateur	H.264/MP4	WebM	OGG Theora
Edge	YES	YES	YES
Chrome	YES	YES	YES
Firefox	YES	YES	YES
Safari	YES	YES	NO
Opera	YES	YES	YES

Vidéo : les attributs

- ▶ **src** : sources de la vidéo
- ▶ **width** et **height** : dimension de la vidéo
 - ▶ Si non spécifiés, valent la largeur et la hauteur du fichier vidéo
 - ▶ Si une dimension est spécifiée mais pas l'autre, le navigateur va ajuster la taille de la dimension non spécifiée afin de préserver les proportions de la vidéo (homothétie)
- ▶ **poster** : permet de spécifier une image que le navigateur utilisera alors que la vidéo est en cours de téléchargement, ou jusqu'à ce que l'utilisateur commence la lecture de la vidéo
 - ▶ Si non spécifié, la première image de la vidéo sera utilisée à la place
- ▶ **controls** : si cet attribut booléen est présent, le navigateur affiche ses propres contrôles vidéo pour la lecture et le volume
 - ▶ Si non présent la première image est affichée (ou l'image poster spécifiée)
 - ▶ Lecture via événement JavaScript seulement.

Vidéo : les attributs

- ▶ **autoplay** : spécifie au navigateur de lancer la lecture de la vidéo automatiquement
- ▶ **autobuffer** : indique au navigateur de télécharger la vidéo tout de suite, en anticipant le fait que l'utilisateur lira la vidéo
 - ▶ Cette partie de la spécification est actuellement en pleine mutation et sujette à changement)
- ▶ **loop** : autre attribut booléen indiquant de lire la vidéo en boucle
 - ▶ Tous les attributs ne sont pas encore supportés par tous les navigateurs, mais les plus courants le sont

Audio : principe de base

- ▶ Similaire à la balise <video>
- ▶

```
<audio controls="controls">
  <source src="song.ogg" type="audio/ogg" />
  <source src="song.mp3" type="audio/mpeg" />
  Your browser does not support the audio element.
</audio>
```
- ▶ Comme pour <video> l'attribut **controls** ajoute des boutons de contrôle



Audio : les formats

- ▶ Ici également, il existe une grosse bataille sur les formats car le format **mp3** n'est pas supporté partout
- ▶ Votre serveur doit donc pouvoir fournir plusieurs formats
- ▶ Une des solutions consiste à transcoder l'audio dans un des formats supportés par le navigateur client

Navigateur	MP3	WAV	OGG
Edge/IE	YES	NO	NO
Chrome	YES	YES	YES
Firefox	YES	YES	YES
Safari	YES	YES	NO
Opera	YES	YES	YES

Attributs Universels

- ▶ Les **attributs universels** sont des attributs communs à l'ensemble des éléments HTML.
- ▶ Ces attributs peuvent donc être ajoutés sur tous les éléments (dans certains cas, les attributs n'auront aucun effet).
- ▶ **id**
 - ▶ Cet attribut définit un identifiant, unique au sein de tout le document, pour un élément.
 - ▶ Il doit permettre d'identifier un élément lorsqu'on crée un lien vers lui et/ou lorsque le manipule avec des scripts ou avec CSS.
- ▶ **class**
 - ▶ Une liste de classes, séparées par des espaces, qui permettent de catégoriser l'élément.
 - ▶ Les classes permettent au CSS et à JavaScript de manipuler des éléments spécifiques grâce à des sélecteurs de classe (pour CSS) ou grâce à des fonctions telles que **document.getElementsByClassName()** (pour JavaScript).

Attributs Universels

- ▶ **contenteditable**
 - ▶ Attribut à valeur contrainte qui indique si l'élément peut être édité par l'utilisateur (**true**) ou non (**false**)
- ▶ **data-***
 - ▶ Attributs de données personnalisés permettant d'échanger des informations dans un format propriétaire entre le HTML et le DOM afin de pouvoir les manipuler via des langage de scripts. La propriété **HTMLElement.dataset** permet d'accéder à l'ensemble des attribut définis de cette façon.
- ▶ **lang**
 - ▶ Aide à définir la langue utilisée pour l'élément.
- ▶ **title**
 - ▶ Attribut contenant une représentation textuelle de l'information auquel il est lié. Celle-ci est généralement affichée sous la forme d'une **bulle** d'informations.
- ▶ [Consulter la liste complètes des attributs universels ici](#)

Éléments en ligne vs Éléments de bloc

Éléments en ligne

- ▶ les éléments en ligne (*inline elements* en anglais) sont ceux qui occupent l'espace délimités par leurs balises plutôt que d'interrompre le flux du contenu.
- ▶ Un élément en ligne ne commence pas sur une nouvelle ligne et prend uniquement la largeur qui lui est nécessaire

Éléments de bloc

- ▶ ils sont séparés par un saut de ligne avant et après l'élément
- ▶ Un élément de bloc commence toujours sur une nouvelle ligne et prend toute la largeur disponible (autrement dit, il s'étend le plus possible vers la droite et vers la gauche).

<https://www.w3.org/TR/2011/WD-html5-20110525/content-models.html#kinds-of-content>

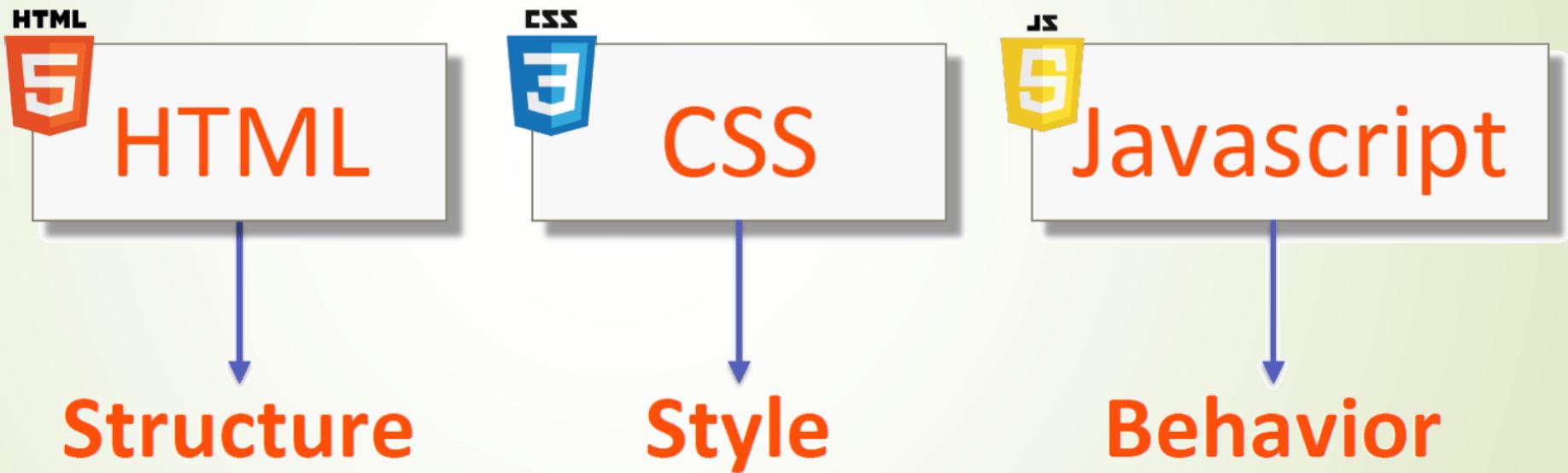
Validation des Documents HTML

- ▶ La page est toujours affichée même s'il y a des erreurs de syntaxe.
- ▶ Chaque Navigateur a des règles spécifiques pour interpréter les balises erronées
- ▶ Mais le résultat est imprévisible
- ▶ C'est la bonne pratique de vous assurer que vos pages sont en HTML légal
- ▶ <https://validator.w3.org/>

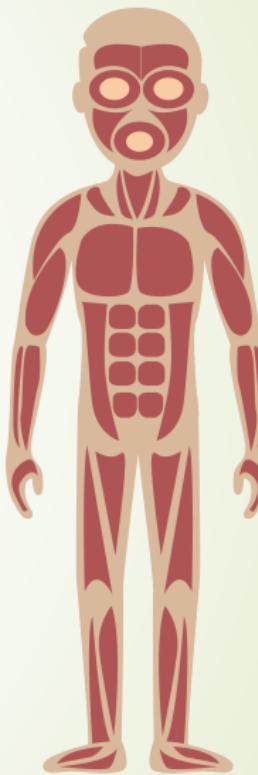
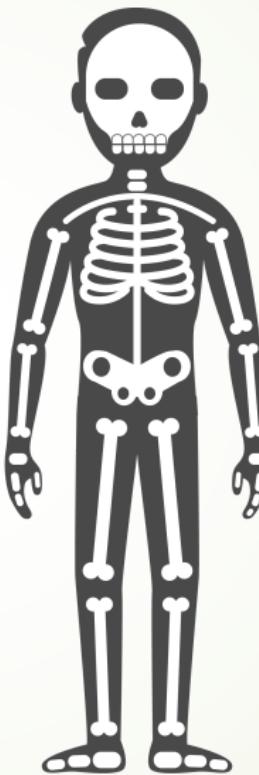
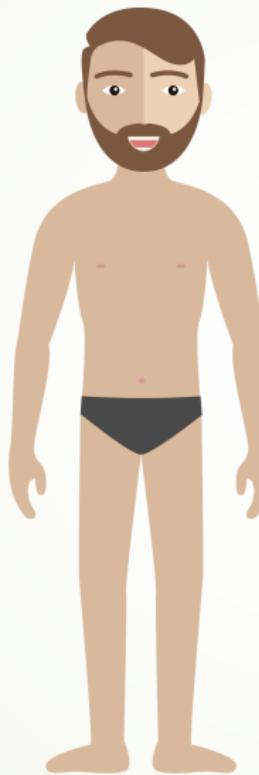


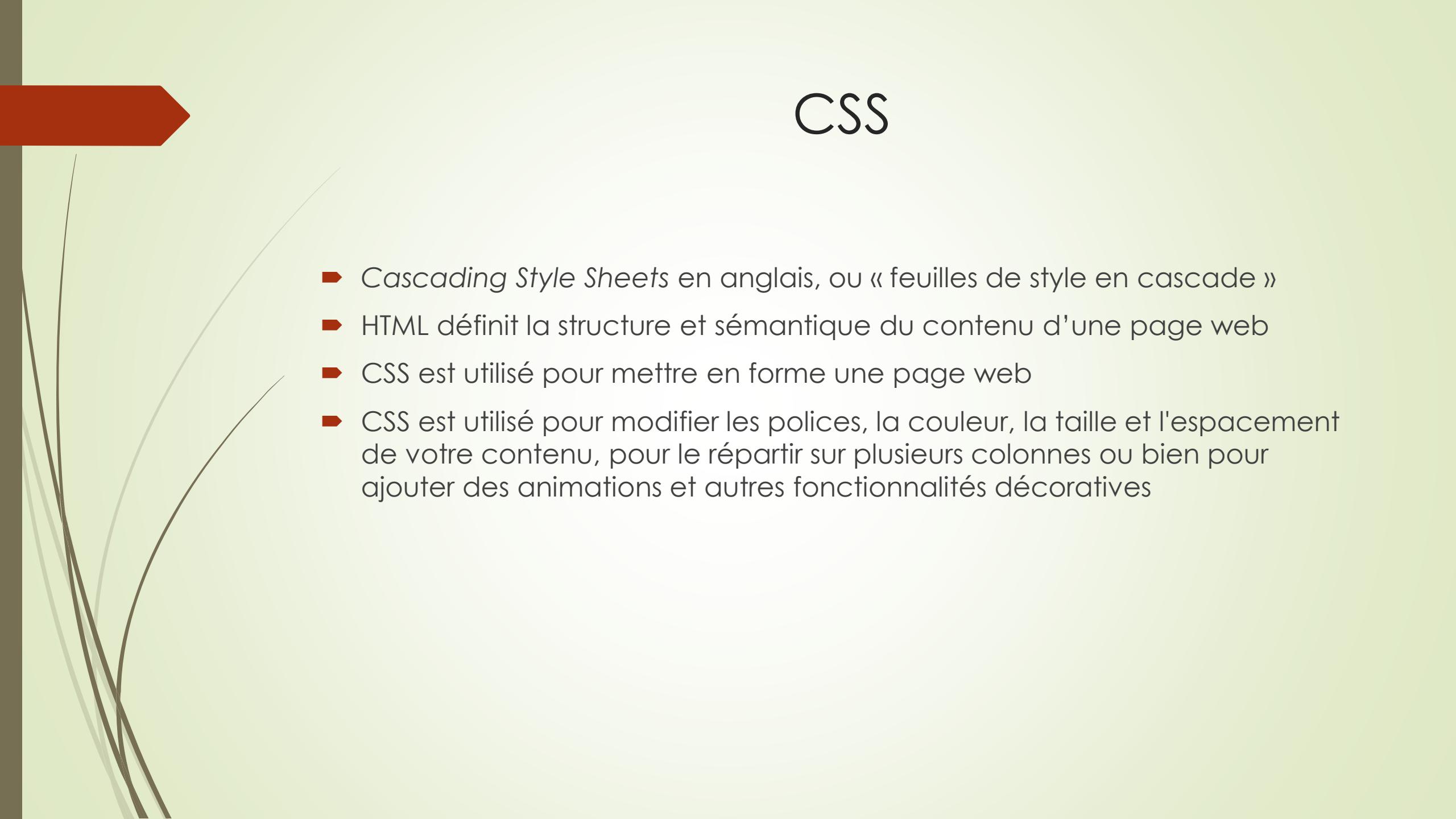
Comprendre la relation entre HTML et CSS

Technologies qui pilotent le web



Technologies qui pilotent le web



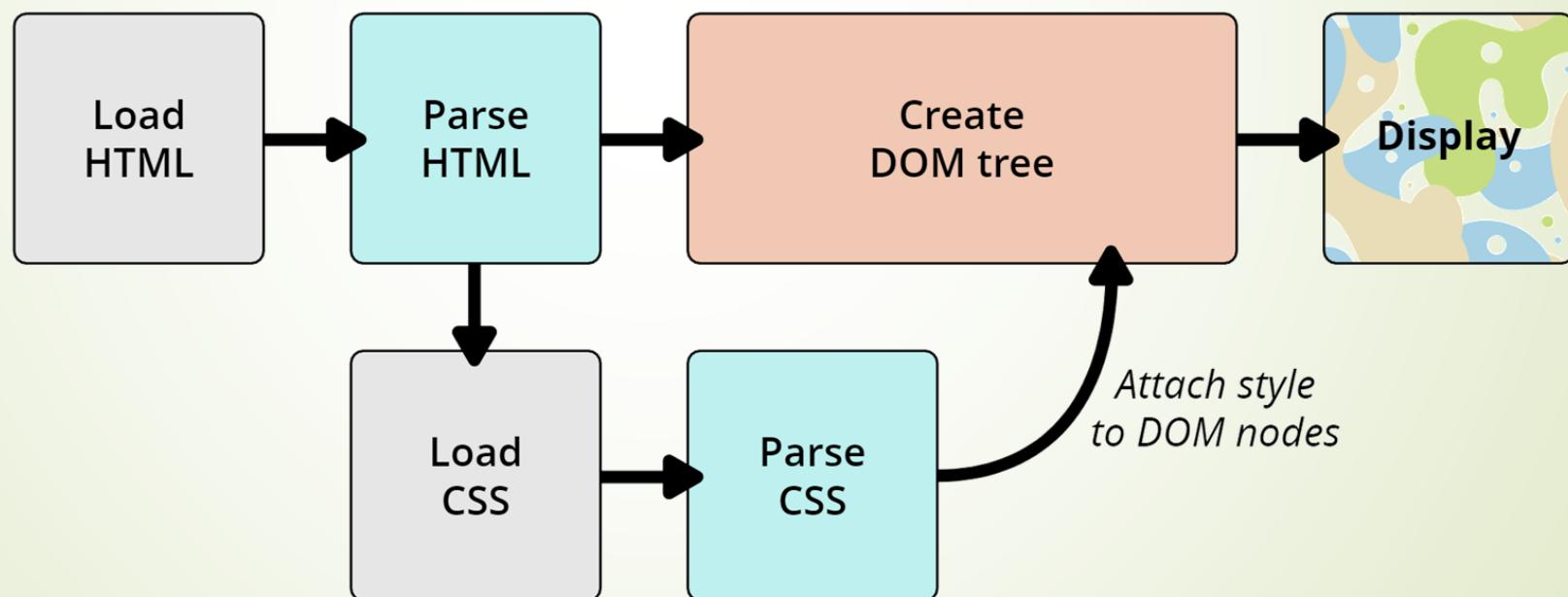


CSS

- ▶ Cascading Style Sheets en anglais, ou « feuilles de style en cascade »
- ▶ HTML définit la structure et sémantique du contenu d'une page web
- ▶ CSS est utilisé pour mettre en forme une page web
- ▶ CSS est utilisé pour modifier les polices, la couleur, la taille et l'espacement de votre contenu, pour le répartir sur plusieurs colonnes ou bien pour ajouter des animations et autres fonctionnalités décoratives

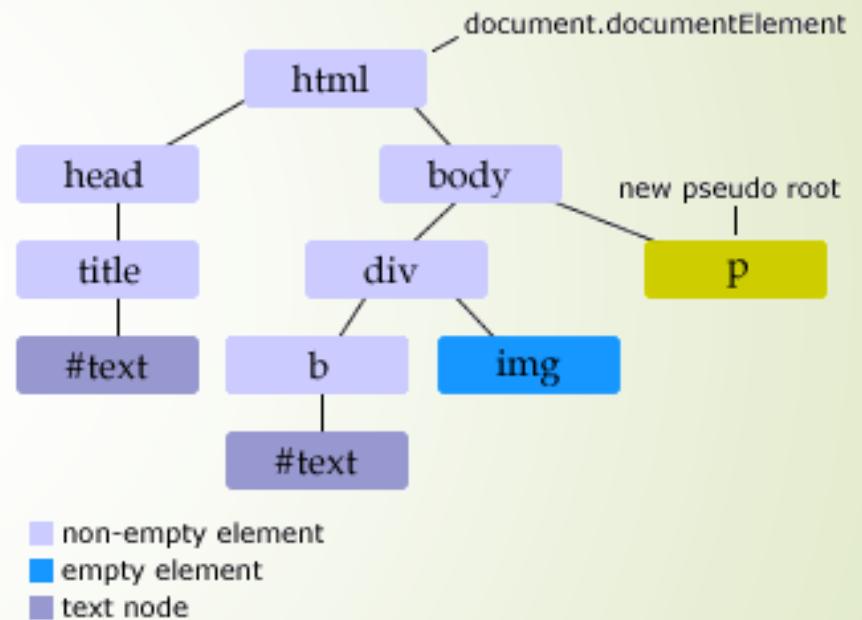
CSS comment ça marche ?

- ▶ Pour afficher un document, un navigateur doit combiner le contenu du document et les informations de mise en forme



Fonctionnement du navigateur

- ▶ Le navigateur lit le document
- ▶ Il en fait l'analyse syntaxique et produit un arbre syntaxique
- ▶ Il lit la feuille de style associée
- ▶ Il interprète les règles de formatage
- ▶ Il parcourt l'arbre et applique à chaque élément sa règle de formatage.
- ▶ En cas de conflit de style, il applique des règles de priorité



Représentation DOM

```
<p>
  Let's use:
  <span>Cascading</span>
  <span>Style</span>
  <span>Sheets</span>
</p>
```

```
P
└ "Let's use:"
  └ SPAN
    └ "Cascading"
  └ SPAN
    └ "Style"
  └ SPAN
    └ "Sheets"
```

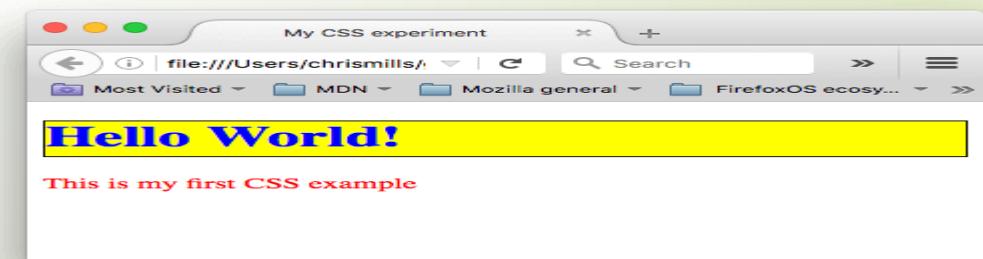
Exemple

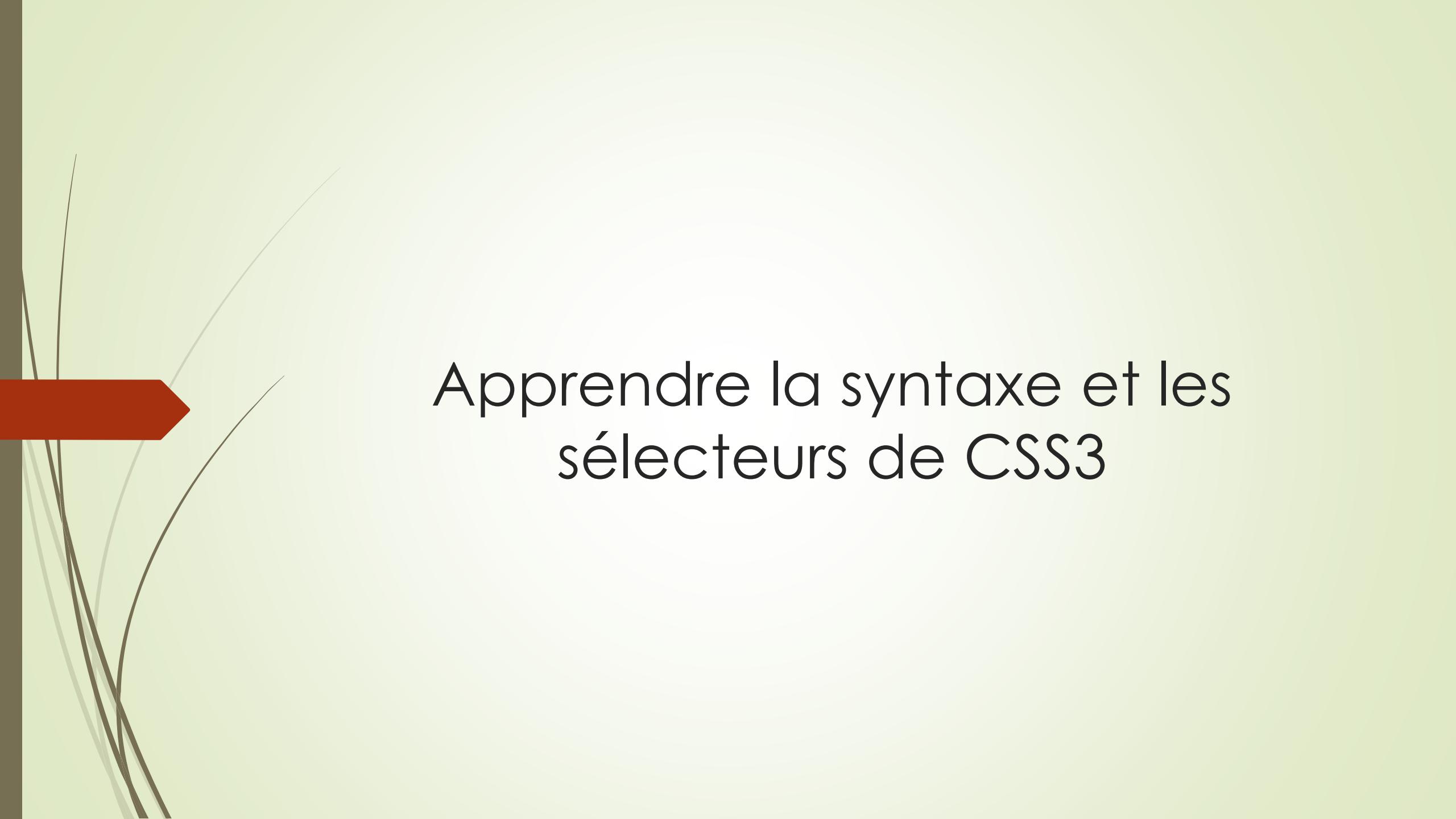
```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>My CSS
experiment</title>
    <link rel="stylesheet"
href="style.css">
  </head>
  <body>
    <h1>Hello World!</h1>
    <p>This is my first CSS
example</p>
  </body>
</html>
```



```
h1 {
  color: blue;
  background-color: yellow;
  border: 1px solid black;
}

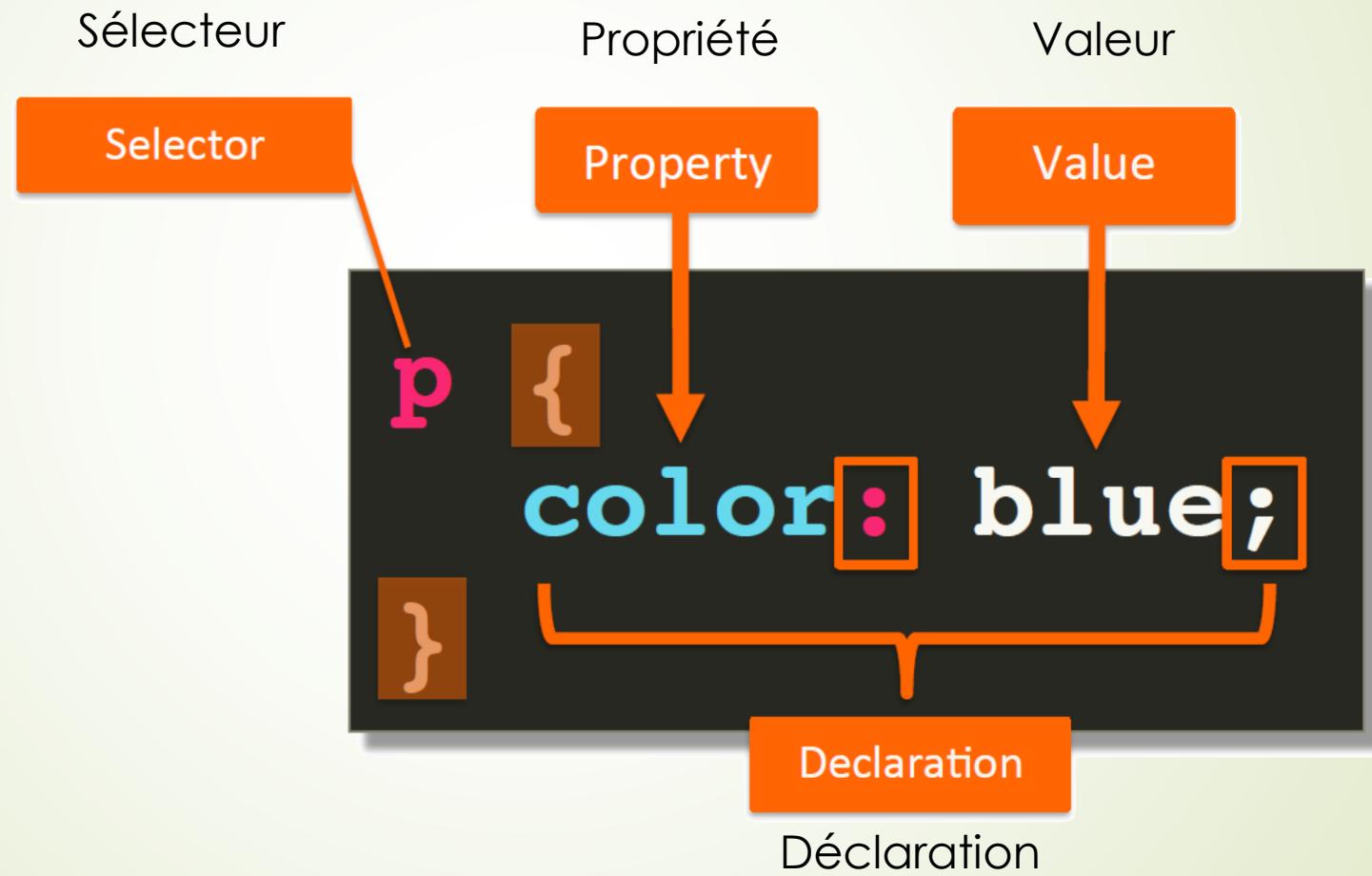
p {
  color: red;
}
```





Apprendre la syntaxe et les
sélecteurs de CSS3

Anatomie d'une règle CSS



Anatomie d'une règle CSS

► Sélecteur

- ▶ C'est le nom de l'élément HTML situé au début de l'ensemble de règles.
- ▶ Il permet de sélectionner les éléments sur lesquels appliquer le style souhaité (en l'occurrence tous les **éléments p**).
- ▶ Pour mettre en forme un élément différent, il suffit de changer le sélecteur.

► Déclaration

- ▶ C'est une règle simple comme **color: blue;** qui détermine les propriétés de l'élément que l'on veut mettre en forme.

► Propriétés

- ▶ Les différentes façons dont on peut mettre en forme un élément HTML (dans ce cas, **color** est une propriété des éléments **p**).
- ▶ En CSS, vous choisissez les différentes propriétés que vous voulez utiliser dans une règle CSS.

Anatomie d'une règle CSS

► Valeur de la propriété

- ▶ À droite de la propriété, après les deux points, on a la valeur de la propriété.
- ▶ Celle-ci permet de choisir une mise en forme parmi d'autres pour une propriété donnée (par exemple, il y a d'autres couleurs que red pour la propriété color).

Zéro ou plusieurs déclarations autorisées

Zero or More Declarations are allowed

```
p {  
    color: blue;  
    font-size: 20px;  
    width: 200px;  
}
```

Anatomie d'une règle CSS

```
p {  
    color: blue;  
    font-size: 20px;  
    width: 200px;  
}  
  
h1 {  
    color: green;  
    font-size: 36px;  
    text-align: center;  
}  
...
```

Feuille de style

Stylesheet

Sélecteur d'élément

Nom d'élément

Element name

```
p {  
  color: blue;  
}
```

Sélecteur d'élément

```
p {  
  color: blue;  
}
```

Texte bleu

Blue text

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

Blue text

Texte bleu

La classe "class" du HTML

- ▶ Class est un attribut "générique" qui s'applique à toutes sortes d'éléments.
- ▶ Une classe permet de définir un sous-ensemble.
- ▶ Elle peut s'appliquer à plusieurs éléments.
- ▶ Elle s'ajoute dans une balise du document html :

```
<h1 class="intro">titre 1</p>  
<p class="intro">texte 2</p>  
<p class="intro">texte 3</p>  
<img class="logo" />
```

Sélecteur de classe

Nom de classe

Class name

```
.blue {  
  color: blue;  
}
```

Sélecteur de classe

```
.blue {  
  color: blue;  
}
```

```
...  
<p class="blue">...</p>  
<p>...</p>  
<div class="blue">...</div>  
...
```

Unaffected

Texte bleu

Blue text

Blue text

Texte bleu

Cet élément n'est pas affecté

Sélecteur de classe

Définit avec .

Defined with .

```
.blue {  
    color: blue;  
}
```

Utilisé sans .

Used without

```
...  
<p class="blue">...</p>  
<p>...</p>  
<div class="blue">...</div>  
...
```

Texte bleu

Blue text

Unaffected

Blue text

Texte bleu

Cet élément n'est pas affecté

L'identifiant id du HTML

- ▶ id est un attribut "générique" qui s'applique à toutes sortes d'éléments.
- ▶ Il sert à identifier une balise précise.
- ▶ Il doit être unique dans un document.
- ▶ Il s'ajoute dans une balise du document html :

```
<p id="intro">texte d'introduction</p>
```

Sélecteur d'id

Valeur d'id

id Value

```
#name {  
    color: blue;  
}
```

Sélecteur d'id

```
#name {  
    color: blue;  
}
```

Cet élément n'est pas affecté

Unaffected

```
...  
<p>...</p>  
<div id="name">...</div>  
...
```

Blue text

Texte bleu

Sélecteur d'id

Défini avec #

Defined with #

```
#name {  
    color: blue;  
}
```

Cet élément n'est pas affecté

Unaffected

```
...  
<p>...</p>  
<div id="name">...</div>  
...
```

Used without

Utilisé sans #

Blue text

Texte bleu

Sélecteur de groupe

Séparez sélecteurs avec les virgules

Separate selectors
with commas

```
div, .blue {  
    color: blue;  
}
```

```
...  
<p class="blue">...</p>  
<p>...</p>  
<div>...</div>  
...
```

Texte bleu

Blue text

Blue text

Texte bleu

Élément avec Sélecteur de classe

Chaque **p** qui a la **class="big"**

Every **p** that has **class="big"**

```
p.big {  
    font-size: 20px;  
}
```

NOTE lack of space between element and class definition

NOTE : Il n'y a pas d'espace entre l'élément et la définition de la classe

Élément avec Sélecteur de classe

```
p.big {  
    font-size: 20px;  
}
```

```
...  
<p class="big">...</p>  
<div class="big">...</div>  
...
```

Taille du texte 20 px

Text size 20px

Unaffected text

Cet élément n'est pas affecté

Sélecteur d'enfant

Chaque **p** qui est un enfant direct d'**article**

Every **p** that is a direct child of **article**

```
article > p {  
    color: blue;  
}
```

Sélecteur d'enfant

```
article > p {  
    color: blue;  
}
```

```
<article>...  
    <p>...</p>  
</article>  
...  
<p>...</p>  
<article>...  
    <div><p>...</p></div>  
</article>
```

Texte bleu

Blue text

Unaffected text

Cet élément
n'est pas
affecté

Unaffected text

Cet élément
n'est pas
affecté

Sélecteur de descendant

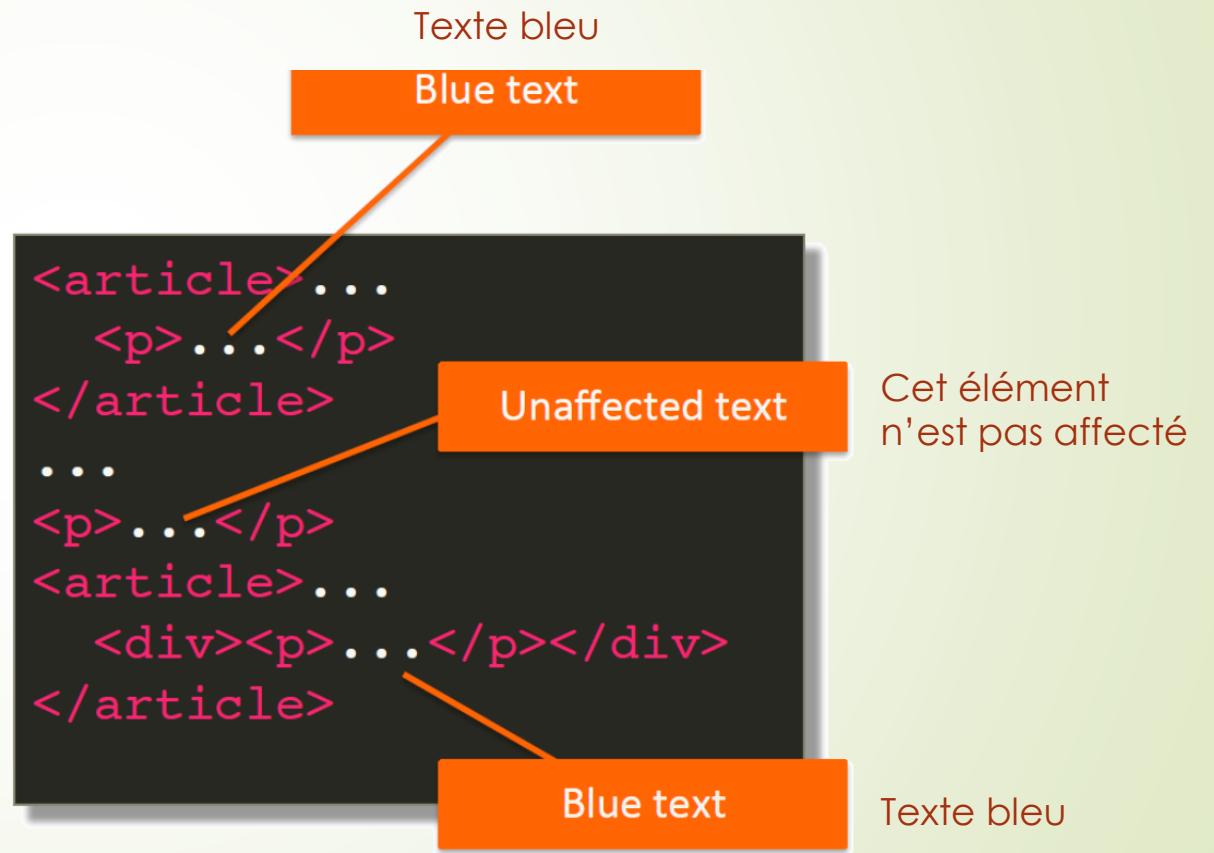
Chaque **p** qui est dans **article** (quelque soit le niveau)

Every **p** that is inside (at any level) of **article**

```
article p {  
    color: blue;  
}
```

Sélecteur de descendant

```
article p {  
    color: blue;  
}
```



Autres combinateurs

- ▶ Les combinateurs sont des sélecteurs qui établissent une relation entre deux sélecteurs ou plus, tel que "A est un enfant de B" ou "A est adjacent à B".
- ▶ **Combinateur de voisin direct A + B**
 - ▶ Indique que les éléments sélectionnés par A et par B ont le même parent et que celui sélectionné par B suit immédiatement celui sélectionné par A.
- ▶ **Combinateur de voisin général A ~ B**
 - ▶ Indique que les éléments sélectionnés par A et par B ont le même parent et que celui sélectionné par B suit celui sélectionné par A, mais pas nécessairement immédiatement après lui.

N'est pas limité au sélecteur d'élément

```
.colored p {  
    color: blue;  
}
```

Every **p** that is inside (at any level) an element with **class="colored"**

Chaque **p** qui est dans l'élément avec **class="colored"** (quelque soit le niveau)

```
article > .colored {  
    color: blue;  
}
```

Every element with **class="colored"** that is a direct child of **article** element

Chaque élément avec **class="colored"** qui est un enfant direct d'**article**



Sélecteur de Pseudo-Classe

```
selector:pseudo-class {  
    ...  
}
```

Sélecteur de Pseudo-Classe

- ▶ Les éléments donnés mais uniquement dans un certain état - par exemple quand on passe la souris dessus :
 - **:link**
 - **:visited**
 - **:hover**
 - **:active**
 - **:nth-child(...)**

Sélecteur de Pseudo-Classe d'ancre

► **:link**

- ▶ permet de sélectionner les liens à l'intérieur d'éléments.
- ▶ Elle sélectionnera tout lien n'ayant pas été visité, même ceux qui seraient déjà mis en forme via des sélecteurs utilisant d'autres pseudo-classes comme :hover, :active ou :visited.

► **:visited**

- ▶ permet de modifier l'aspect d'un lien après que l'utilisateur l'a visité.

► **:hover**

- ▶ permet de spécifier l'apparence d'un élément au moment où l'utilisateur le survole avec le pointeur, sans nécessairement l'activer.

► **:active**

- ▶ permet de cibler un élément lorsque celui-ci est activé par l'utilisateur.

Sélecteur de Pseudo-Classe

- ▶ **:nth-child(an+b)**
 - ▶ Permet de cibler les éléments fils d'un élément dont les positions correspondent au motif $an+b$
- ▶ **:first-child**
 - ▶ permet de cibler un élément qui est le premier élément fils par rapport à son élément parent.
- ▶ **:last-child**
 - ▶ permet de cibler un élément qui est le dernier enfant de son parent.
- ▶ [Consulter la référence des sélecteurs CSS ici](#)

Sélecteur de Pseudo-éléments

- ici appliqués à l'élément p

`p::first-line {font-variant: small-caps;}` → 1re ligne des paragraphes

`p::first-letter {font-size:2em;color:red;}` → 1re lettre des paragraphes

`::before` permet d'insérer un texte avant

`::after` permet d'insérer un texte après

► Exemples

```
td::before { content:"Prix: "; }
```

12,50



```
td::after { content:" euros"; }
```

Prix: 12,50 euros

```
th::before { content:url(carre.gif)" ";
```

Produits vendus

Sélecteur universel *

- ▶ * remplace tout élément du document
- ▶ Se met dans la partie style CSS, par exemple :
`* {margin: 0}`
- ▶ Tous les éléments du document (h1, h2, p, div, ul...) auront une marge de 0.

Sélecteurs d'attribut

Exemples

- ▶ `a[title]`
 - ▶ Les éléments `<a>` avec un attribut `title`
- ▶ `a[href="https://example.org"]`
 - ▶ Les éléments `<a>` avec un `href` qui correspond à <https://example.org>
- ▶ `a[href*="example"]`
 - ▶ Les éléments `<a>` dont `href` contient "example"
- ▶ `a[href$=".org"]`
 - ▶ Les éléments `<a>` dont `href` finit par ".org"

/* Utilisation des commentaires en CSS */

```
/* Auteur : Soupramanien BOUVANESVARY */  
a { color:#333; /* gris sombre */  
}
```

Commentaire conditionnel

- ▶ Si le navigateur est inférieur (**lt** pour less than) à la version d'IE, alors on ajoute un style...

```
<!--[if lt IE 7]>  
  <style type="text/css">  
    div {  
      width:expression(document.body.clientWidth >= 1000? "1000px": "auto" );  
    }  
  </style>  
<![endif]-->
```

Codage des couleurs en CSS

Propriétés CSS pour les couleurs :

- ▶ `background-color` : couleur de l'arrière-plan
- ▶ `border-color` : couleur de la bordure
- ▶ `color` : couleur de l'avant plan (texte, par exemple)

Il existe plusieurs manières de spécifier une couleur en CSS :

- ▶ La plus classique est le code de couleur **Hexadécimal** - `color : #FFB6C1`
- ▶ La moins classique est le code couleur **nommé** (16 sont reconnues par le W3C mais tous les navigateurs actuels reconnaissent les couleurs définies par la norme X11) - `color : CornflowerBlue`
- ▶ Le code couleur **RGB** (Red Green Blue ou Rouge Vert Bleu) - `color : rgb(255,182,193)`
- ▶ Le code couleur **HSL** (Hue Saturation Lightness ou Teinte Saturation Luminosité) - `color : hsl(351,100%,86%)`
- ▶ Il est maintenant aussi possible depuis CSS3 de spécifier une couleur incluant la transparence avec les fonctions **RGBA**, **HSLA** et bientôt `#RRGGBBAA`

Codage des couleurs en CSS

Les notations de type RGB :

- ▶ Pour créer une couleur RGB on doit préciser trois niveaux d'intensité de Rouge, de Vert, et de Bleu qui vont ensuite être mélangés pour créer la couleur finale
- ▶ Chaque niveau d'intensité qu'on va renseigner va être compris entre 0 (intensité nulle ou absence de la couleur en question) et 255 (intensité maximale ou couleur pure).
- ▶ Exemples :
 - ▶ Codage décimal `rgb(x, y, z)` avec nombre de 0 à 255 : `rgb(255, 0, 122)`
 - ▶ Codage décimal `rgb(x%, y%, z%)` en pourcentage : `rgb(100%, 0%, 50%)`
 - ▶ Codage décimal `rgba(x, y, z)` avec nombre de 0 à 255 et transparence : `rgb(255, 0, 122, 0.5)`

Codage des couleurs en CSS

Les valeurs de type hexadécimales : # **XX XX XX**

- ▶ Elles vont reposer sur le même principe que les valeurs de type RGB : préciser trois niveaux d'intensité de rouge, de vert et de bleu pour créer une couleur personnalisée.
- ▶ La seule différence va être la façon dont on va compter. Le système hexadécimal utilise 16 symboles pour compter et représenter les chiffres.

Décimal	0 (ou 00)	1 (ou 01)	2 (ou 02)	9 (ou 09)	10	11	15	16	17	...	254	255
Hexadécimal	0 (ou 00)	1 (ou 01)	2 (ou 02)	9 (ou 09)	A (ou 0A)	B (ou 0B)	F (ou 0F)	10	11	...	FE	FF

- ▶ Exemples :
 - ▶ Codage hexadécimal classique de type #rrggb : #cc0088
 - ▶ Codage hexadécimal abrégé de type #rgb : #c08

#000000

#454545

#999999

#FF0000

#00FF00

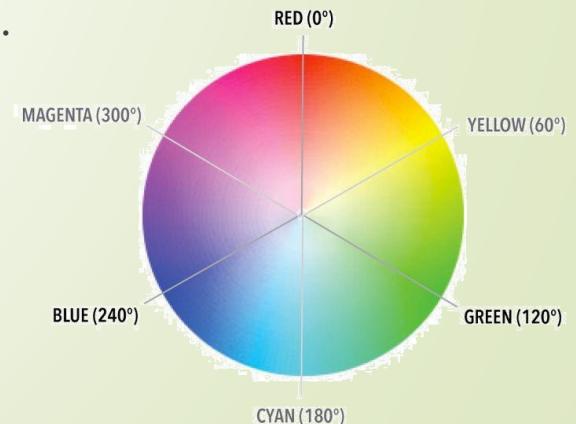
#0000FF

#FFFFFF

Codage des couleurs en CSS

Les notations de type HSL :

- ▶ Pour créer une couleur en utilisant les notations HSL, il faut renseigner trois valeurs :
 - ▶ **La teinte** : nombre entre 0 et 360 qui représente un angle du cercle chromatique. Le rouge correspond à un angle de 0 (degré) ou de 360 (degrés), tandis que le vert va se situer à 120 degrés et le bleu à 240 degrés.
 - ▶ **La saturation** : sous forme de pourcentage. 100% correspond à une saturation maximale tandis que 0% correspond à une saturation minimale.
 - ▶ **La luminosité** : également représentée par un pourcentage. 100% de luminosité correspond à du blanc tandis que 0% correspond à du noir.
- ▶ Exemples :
 - ▶ Teinte = 45° (à mi-chemin entre le orange et le jaune) / Saturation à 100% et luminosité à 60% : `hsl(45, 100%, 60%)`
 - ▶ Teinte à 150° (dans les vert), saturation à 50% et luminosité à 50% : `hsl(150, 50%, 50%)`



Codage des couleurs en CSS

16 couleurs de base

- A l'époque, la plupart des écrans d'ordinateurs avait encore des capacités limitées et ne pouvaient reproduire qu'un jeu réduit de couleurs

#00FFFF	#000000	#0000FF	#FF00FF
Aqua	Black	Blue	Fuchsia
#000080	#808000	#800080	#FF0000
Navy	Olive	Purple	Red
#808080	#008000	#00FF00	#800000
Gray	Green	Lime	Maroon
#COCOCO	#008080	#FFFFFF	#FFFF00
Silver	Teal	White	Yellow

Les couleurs X11

- La norme X11 fut définie pour gérer les terminaux à l'époque des mini-ordinateurs. Elle comporte un nombre relativement étendu de couleurs qui sont reconnues par les principaux navigateurs actuels.

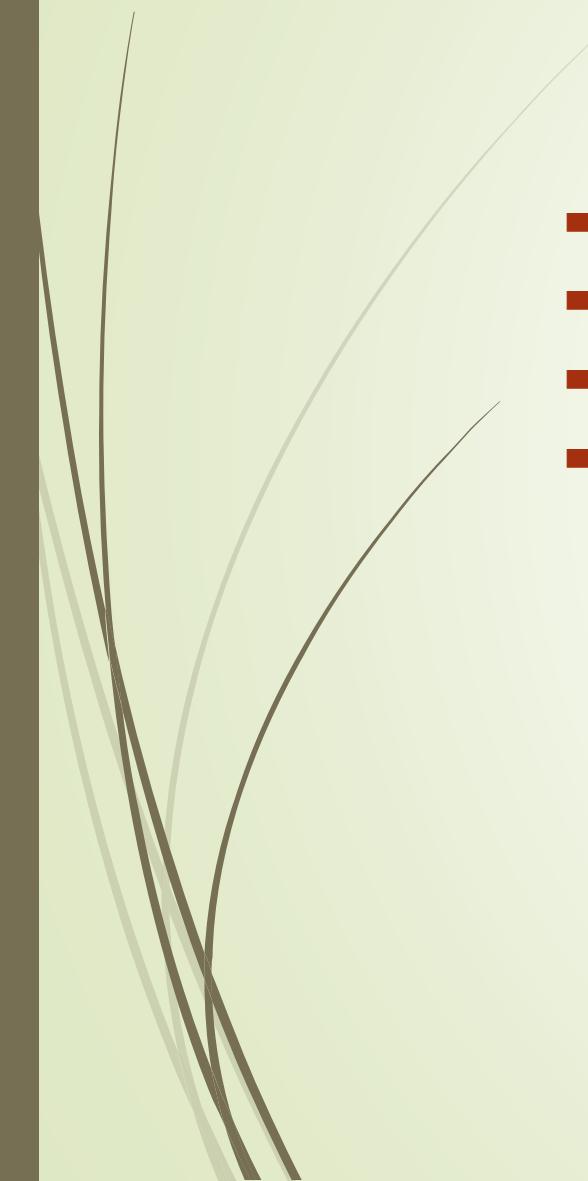
Nom de couleur	Exemple	HTML	Composantes (rouge, vert, bleu)			Décimal
			Hexadécimal			
AliceBlue		#F0F8FF	F0	F8	FF	240 248 255
AntiqueWhite		#FAEBD7	FA	EB	D7	250 235 215
Aqua		#00FFFF	00	FF	FF	0 255 255
Aquamarine		#7FFFDD	7F	FF	D4	127 255 212
Azure		#FOFFFF	F0	FF	FF	240 255 255
Beige		#F5F5DC	F5	F5	DC	245 245 220
Bisque		#FFE4C4	FF	E4	C4	255 228 196

Unités absolues

Unité	Nom	Équivalence
cm	Centimètre	$1\text{cm} = 96\text{px}/2.54$
mm	Millimètre	$1\text{mm} = 1/10\text{e}$ de 1cm
Q	Quart de millimètre	$1\text{Q} = 1/40\text{e}$ de 1cm
in	Pouces (inches)	$1\text{in} = 2.54\text{cm} = 96\text{px}$
pc	Picas	$1\text{pc} = 1/16\text{e}$ de 1in
pt	Points	$1\text{pt} = 1/72\text{e}$ de 1in
px	Pixels	$1\text{px} = 1/96\text{e}$ de 1in

Unités relatives

Unité	Relative à
em	La taille (corps) de police de l'élément
ex	La hauteur d'un x avec la police utilisée par l'élément
rem	La taille (corps) de police de l'élément racine
%	parent
vw	1% de la largeur de la zone d'affichage (viewport)
vh	1% de la hauteur de la zone d'affichage (viewport)
vi	1% de la taille de la zone d'affichage sur l'axe en ligne (inline axis)
vb	1% de la taille de la zone d'affichage sur l'axe de bloc (block axis)
vmin	1% de la zone d'affichage selon sa plus petite dimension
vmax	1% de la zone d'affichage selon sa plus grande dimension



Remarques

- ▶ L'abréviation de l'unité doit être collée à la valeur, sans espace
- ▶ La taille de police est exprimée le plus souvent en pt, en em ou en %
- ▶ Largeur et hauteur d'une image sont exprimées en général en px
- ▶ Les tailles pour l'impression (marges...) sont exprimées en cm ou en in

Propriétés de police de caractères

- ▶ color
- ▶ background-color
- ▶ font-style: normal | italic | oblique
- ▶ font-variant: normal | small-caps
- ▶ font-weight : normal | bold | bolder | lighter | 100 | 700 | 900
- ▶ font-size : n | p% | xx-small | small | smaller | large | larger | xx-large
- ▶ line-height: n p% (interligne entre 2 lignes)
- ▶ font-family: police1, police2, police3
- ▶ **Exemple de propriété raccourcie (ordre important) :**
font: italic small-caps bold 10pt/14pt Times

Propriété font-family

- ▶ **Liste de polices** (pour différents OS + police générique en dernier)
font-family: Verdana, Geneva, Arial, Helvetica, sans-serif;
font-family: Georgia, "Bookman Old Style", serif;

- ▶ **5 familles de police génériques**

- ▶ serif (ex. Times, Times new roman, Palatino, Georgia)
- ▶ sans-serif (ex. Geneva, Arial, Helvetica, Tahoma, Verdana)
- ▶ cursive (ex. Zapf-Chancery, Comic Sans MS)
- ▶ fantasy (ex. Western, Impact)
- ▶ monospace ou à taille fixe (ex. Courier)

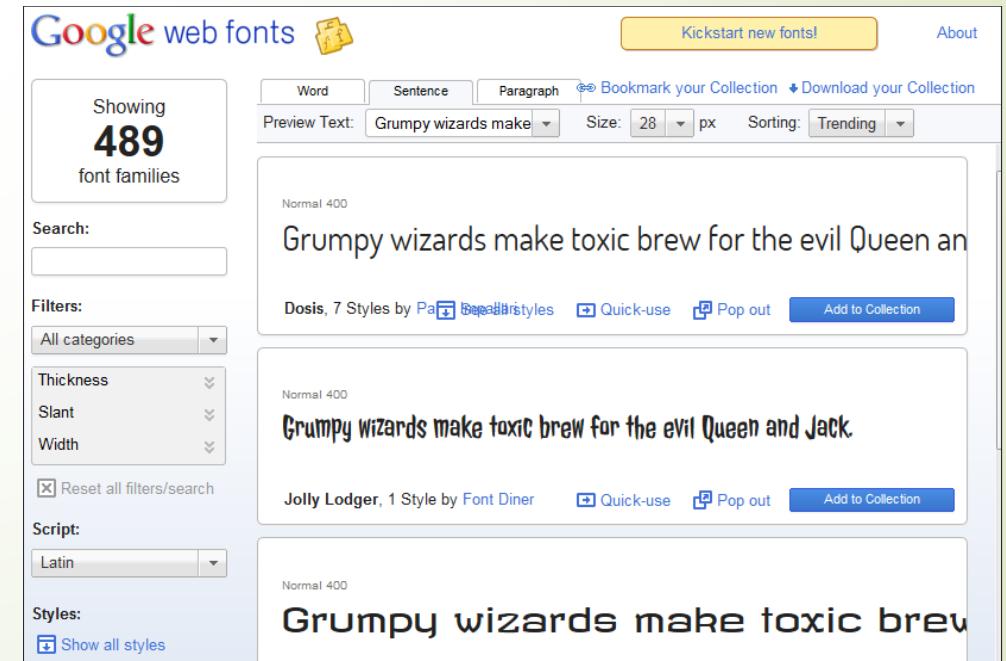
- ▶ **Nom de police avec espace**

font-family: Times, "Times New Roman", serif

font-family: Times, 'Times New Roman', serif

Règle font-face

- ▶ Permet de définir les polices d'écriture à utiliser pour afficher le texte de pages web.
- ▶ Elle peut être chargée depuis un serveur distant **url()** ou depuis l'ordinateur de l'utilisateur **local()**.
- ▶ Plusieurs types de polices :
 - ▶ Embedded OpenType (EOT)
 - ▶ True Type (TTF)
 - ▶ Open Type (OTF)
 - ▶ Web Open Font (WOFF)
 - ▶ SVG sur iPad et iPhone
- ▶ Accès via CDN :
 - ▶ <https://fonts.googleapis.com/>



Propriétés de texte

- ▶ word-spacing: n | -n
- ▶ letter-spacing: n | -n
- ▶ text-decoration: none | underline | overline | blink | line-through
- ▶ text-transform: uppercase | lowercase | capitalize
- ▶ text-align: left | right | center | justify
- ▶ text-indent (retrait de première ligne) : n p%
- ▶ white-space : normal | pre | nowrap

Propriétés de blocs : marges

- ▶ margin-top
- ▶ margin-right
- ▶ margin-bottom
- ▶ margin-left

valeurs possibles :

n p% -n -n% auto

exemples :

5pt -12px 2em 10%

▶ Exemples de propriété raccourcie :

On part d'en haut, on tourne dans le sens des aiguilles d'une montre

margin: 5em 10em 5em 10em

margin: 5em --> 5em pour toutes les marges

margin: 5em 10em --> 5em haut/bas 10em droit/gauche

margin: 5em 3em 2em (=droit) --> 5em haut 3em droit 2em bas 3em gauche



Propriétés de blocs : remplissage

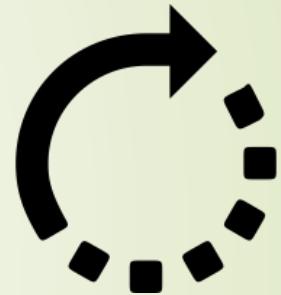
- ▶ padding-top: n p%
- ▶ padding-right
- ▶ padding-bottom
- ▶ padding-left
- ▶ padding

valeurs possibles :

n p% -n -n% auto

exemples :

5pt -12px 2em 10%



▶ Exemples de propriété raccourcie :

on part d'en haut, on tourne dans le sens des aiguilles d'une montre

padding: 5em 10em 5em 10em

padding: 5em --> 5em pour toutes

padding: 5em 10em --> 5em haut/bas 10em droit/gauche

padding: 5em 3em 2em --> 5em haut 3em droit 2em bas 3em gauche

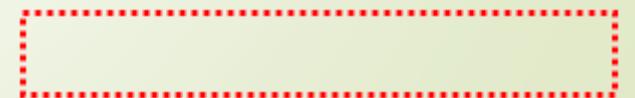
Propriétés de blocs : bordures

- ▶ border-top-width: n | thin | medium | thick (idem avec bottom)
- ▶ border-left-width: n | thin | medium | thick (idem avec right)
- ▶ border-width: 2pt 9pt 9pt 2pt

- ▶ border-color: red blue green black
- ▶ pas de couleur précisée : color de l'élément

- ▶ border-style: none | solid | dotted | dashed | ...
- ▶ border-style: double solid

- ▶ border-top: 2px dashed red
- ▶ border: 2px dotted red --> 4 bordures identiques



Propriétés de blocs : coins arrondis

- ▶ La propriété **border-radius** permet de définir des coins arrondis pour la bordure d'un élément.
- ▶ La courbure de chaque coin est définie avec un ou deux rayons de courbures qui permettent de définir un arc de cercle ou un arc d'ellipse.
- ▶ Exemples :
 - ▶ border-radius: 30px;
 - ▶ **border-radius: 25% 10%;**
 - ▶ border-radius: 10% 30% 50% 70%;
 - ▶ border-radius: 10% / 50%;
 - ▶ border-radius: 10px 100px / 120px;
 - ▶ border-radius: 50% 20% / 10% 40%;

This is a box with rounded corners.

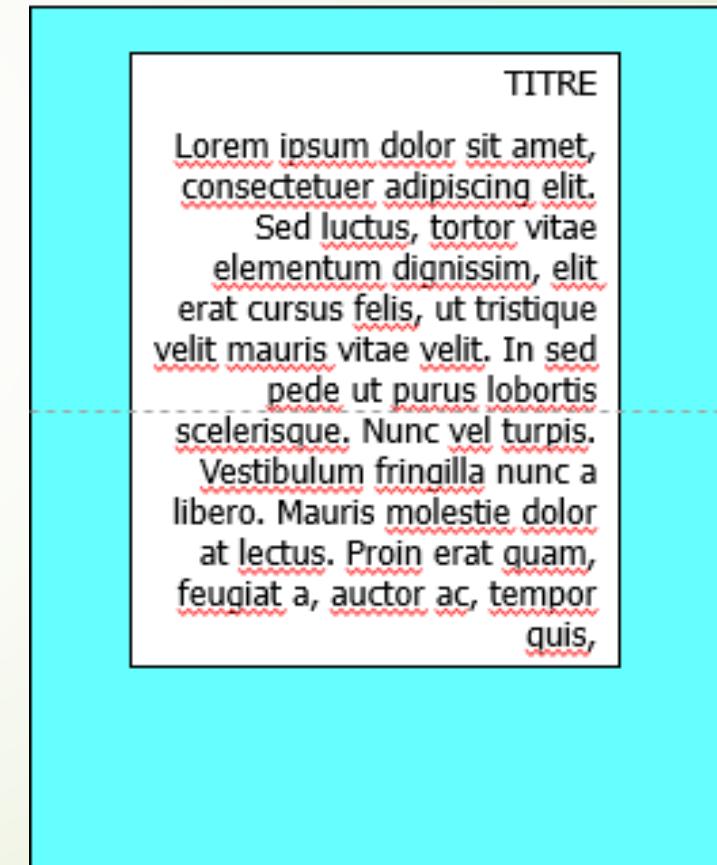
Utilisation de marges automatiques pour centrer un bloc

- ▶ Dans la partie **CSS**

```
#conteneur {  
    width: 600px;  
    background-color:cyan;  
    margin-left: auto;  
    margin-right: auto;  
    padding: 1em;  
    text-align:right;  
}
```

- ▶ en **HTML**

```
<body>  
<div id="conteneur">...  
</div>  
</body>
```



Propriétés d'arrière-plan

- ▶ Background-attachment : scroll | fixed
- ▶ Background-position : 50% 50% (ou 20px 10px)
d'abord verticale puis horizontale
- ▶ Background-color : #ccc
- ▶ Background-image : url(logo.gif) | gradient
- ▶ Background-repeat : repeat | repeat-x | repeat-y | no-repeat

Propriété raccourcie :

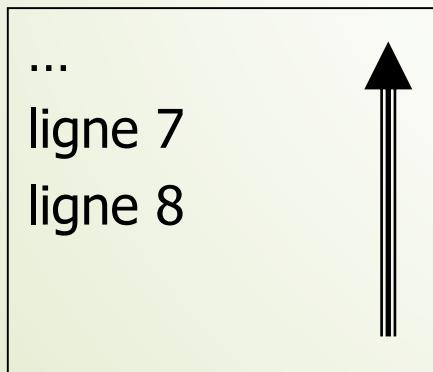
- ▶ `Background: scroll 50% 50% #ccc url(logo.gif) no-repeat`

Background-attachment

haut de l'écran

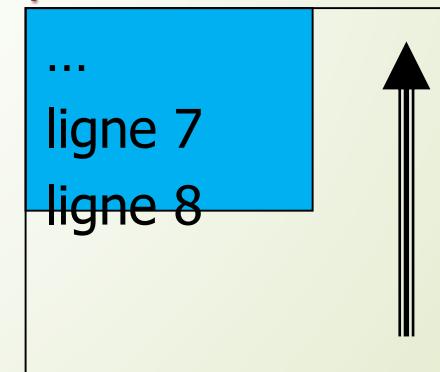


fixed



*bas de l'écran
après défilement*

par défaut : scroll



Background-repeat

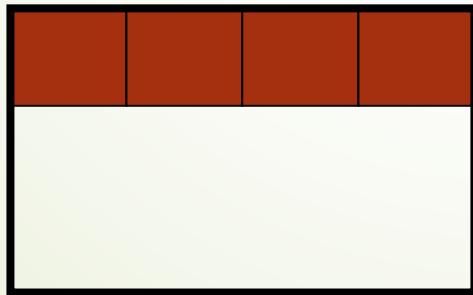
no-repeat



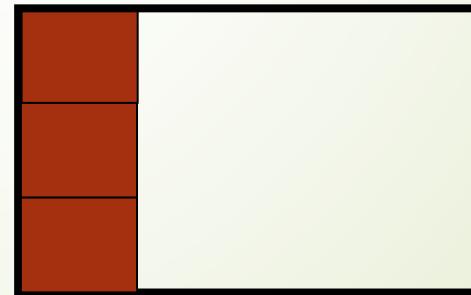
par défaut : repeat



repeat-x



repeat-y



Background-position

background-position: x y (valeur négative possible)

par
défaut

0% 0% top left left top		50% 0% top center center top top		100% 0% top right right top
0% 50% left center center left left		50% 50% center center center		100% 50% right center center right right
0% 100% bottom left left bottom		50% 100% bottom center center bottom bottom		100% 100% bottom right right bottom

Background-image

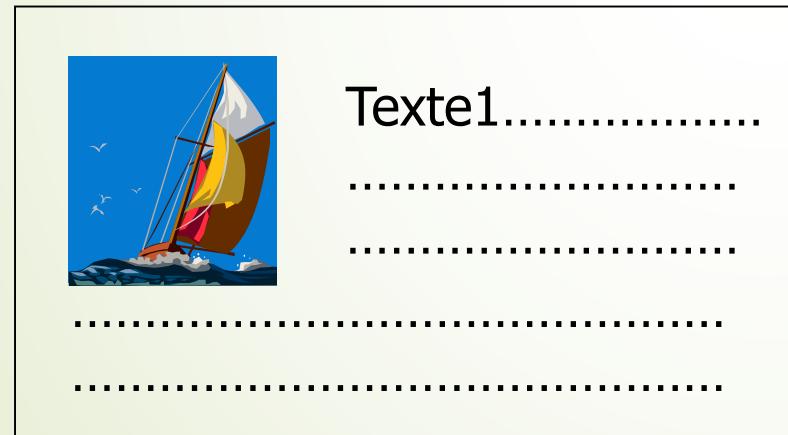
- ▶ Les fonctions **linear-gradient()** et **radial-gradient()** permettent de créer une image représentant un dégradé de couleur linéaire.
- ▶ Exemples :
 - ▶ **linear-gradient(#e66465, #9198e5);**
 - ▶ linear-gradient(red, orange, yellow, green, blue);
 - ▶ linear-gradient(red 0%, orange 25%, yellow 50%, green 75%, blue 100%);
 - ▶ linear-gradient(to left top, blue, red);
 - ▶ **radial-gradient(#e66465, #9198e5);**
 - ▶ radial-gradient(circle at center, red 0, blue, green 100%)



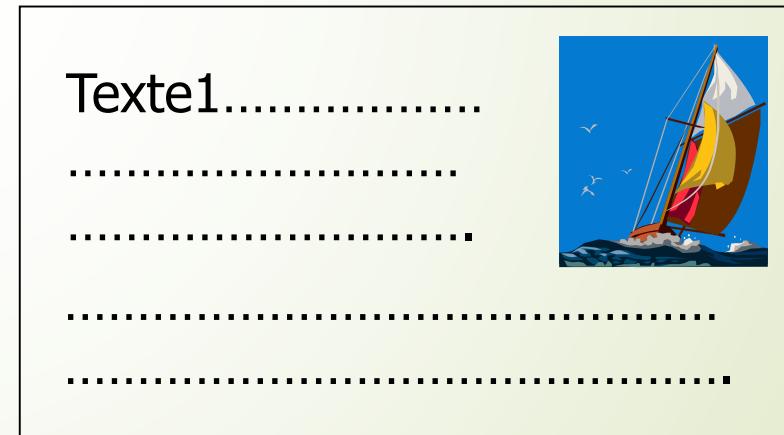
Propriété Float (valeur : right | left)

- float: left | right
- Utilisation : image, menu, bloc de textes, lettrine...
- **Exemple :** <p>Texte1... </p>

► `img {float: left;}`



► `img {float: right;}`



Mise en page d'images et de blocs avec "Float"

► Galerie d'images

```
img {float:left}
```

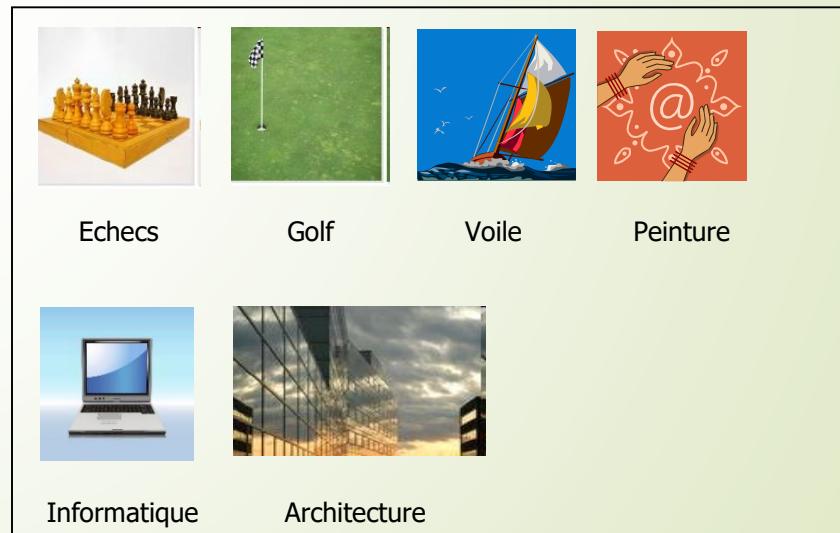
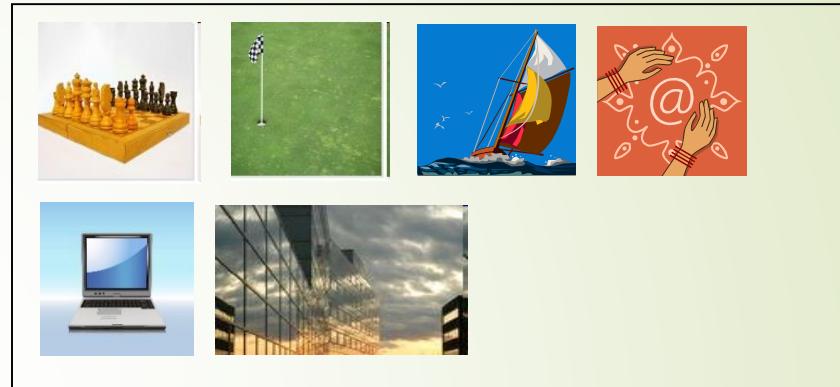
```
  
...
```

► Blocs avec image et légende

```
figure {float:left; width:100px;}
```

```
<figure>  
    
  <figcaption>legende 1</figcaption>  
</figure>  
<figure>  
    
  <figcaption>legende 2</figcaption>  
</figure>...
```

► Plus facile à gérer qu'avec un tableau



Propriété Clear (valeur left | right | both)

- ▶ **code HTML :**

```
<p><img .... /></p>
<p>Texte A...</p>
<p id="textB">Texte B</p>
```

- ▶ **code CSS exemple 1 :**

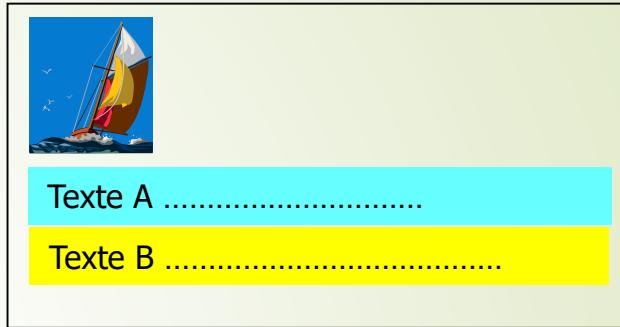
```
img {float : right;}
```

- ▶ **code CSS exemple 2 :**

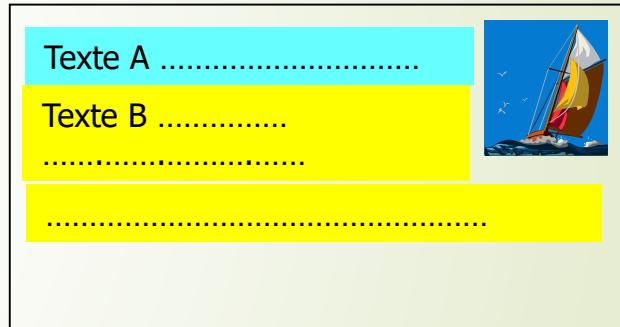
```
img {float : right;}
#textB {clear: right;}
```

- ▶ **Clear** permet qu'un élément se place tel qu'il n'ait rien à sa droite, à sa gauche ou ni l'un, ni l'autre.

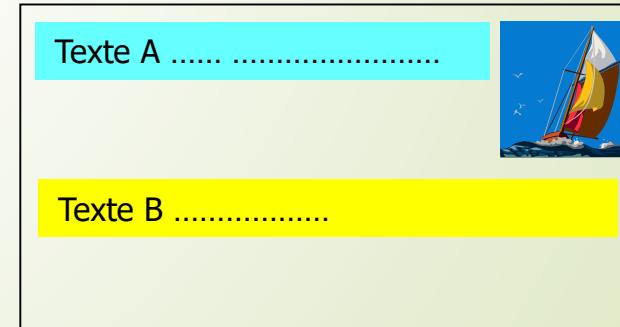
initial



ex 1



ex 2



Propriétés de listes



- ▶ list-style-type : none | disc | circle | square | decimal | lower-roman | upper-roman | lower-alpha | upper-alpha
- ▶ list-style-image : url("chemin/fichier.gif")
- ▶ list-style-position : outside | inside
- ▶ **Propriété raccourcie :**
`list-style : list-style-type list-style-image list-style-position`
- ▶ **Exemples :**
`list-style-image : url(carrebleu1.gif)`
`list-style-type : none`
`list-style : square inside`

• liste avec
outside

• liste avec
inside

Propriété Display

Display : none | inline | block | list-item

- ▶ Exemple 1 :

```
<ul>  
<li>item1</li>  
<li>item2</li>  
</ul>
```

- item1
- item2
- item3

- ▶ en CSS :

```
li {display:inline;}
```

item1 item2

- ▶ Exemple 2 :

```
<a href="#">item1</a>  
<a href="#">item2</a>
```

item1 item2

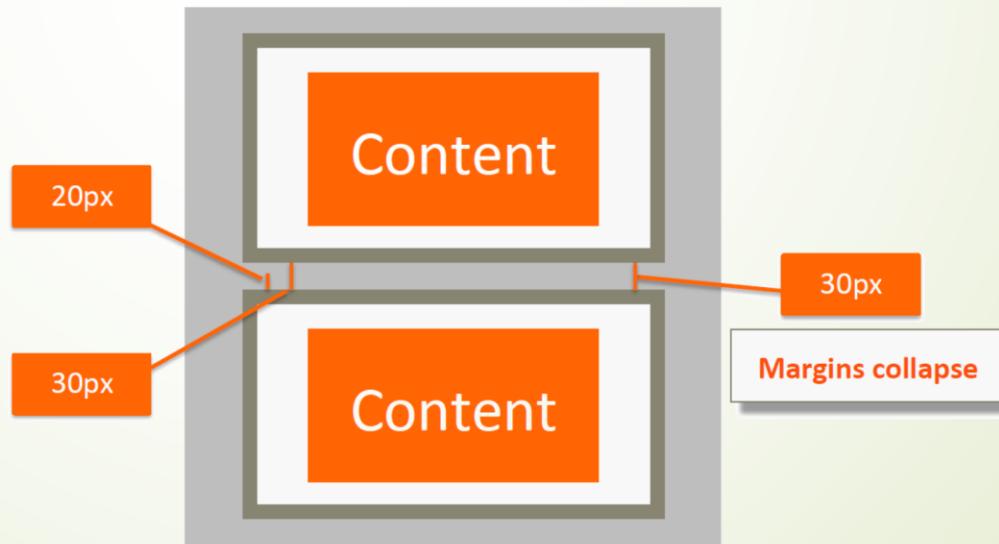
- ▶ en CSS :

```
a {display:block;}
```

item1
item2

Fusion des marges

- ▶ Si deux éléments de votre page ont des margins qui se touchent, alors ces margins **fusionnent pour ne faire plus qu'un seul margin** qui aura pour taille la plus grande des deux tailles des margins initiaux.

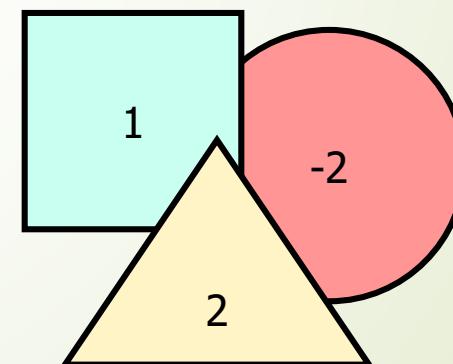


Le positionnement

- ▶ Type de **positionnement** : static | relative | absolute | fixed
 - static (statique) : flux normal du code HTML
 - relative : décalage par rapport à la position statique
 - absolute (absolue) : position par rapport au parent
 - fixed (fixe) : position par rapport à la zone de visualisation
- ▶ **Emplacement**

top : n | p % | -n | -p %
bottom : n | p % | -n | -p %
right : n | p % | -n | -p %
left : n | p % | -n | -p %
- ▶ **Superposition**

z-index : n | -n
plus n est grand, plus l'élément est au-dessus des autres



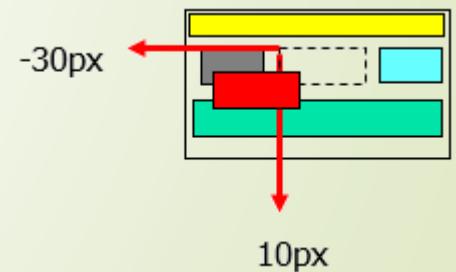
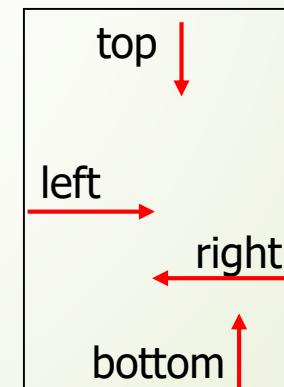
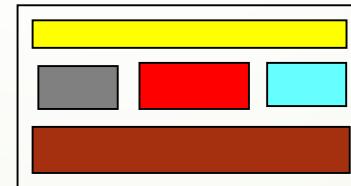
Positionnement static

- ▶ Comportement normal (par défaut).
- ▶ L'élément est alors positionné dans le flux avec sa position.
- ▶ Les propriétés top, right, bottom, left et z-index ne s'appliquent pas.

Positionnement 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.
- ▶ Le décalage n'a pas d'impact sur la position des éléments.
- ▶ Aussi, l'espace fourni à l'élément sur la page est le même que celui fourni avec static.

```
div#bloc1 {  
    position: relative;  
    top: 10px;  
    left: -30px;  
}
```



Positionnement relative

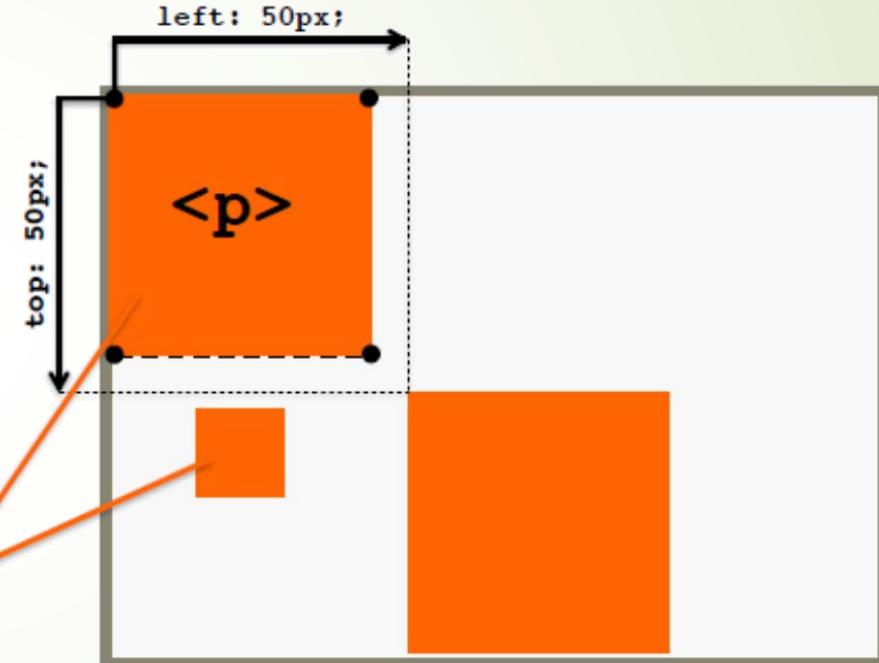
```
p {  
}  
}
```

<p>



Positionnement relative

```
p {  
  position: relative;  
  top: 50px;  
  left: 50px;  
}  
  
from  
from
```



Positionnement absolu

- ▶ L'élément est retiré du flux normal et aucun espace n'est créé pour l'élément sur la page.
- ▶ Il est ensuite positionné par rapport à son ancêtre le plus proche qui est positionné autre que static s'il y en a un ou par rapport au bloc englobant initial sinon.
- ▶ Par défaut, HTML est le seul élément qui a le positionnement non-static.
- ▶ La position finale de l'élément est déterminée par les valeurs de top, right, bottom et left.

Positionnement absolu

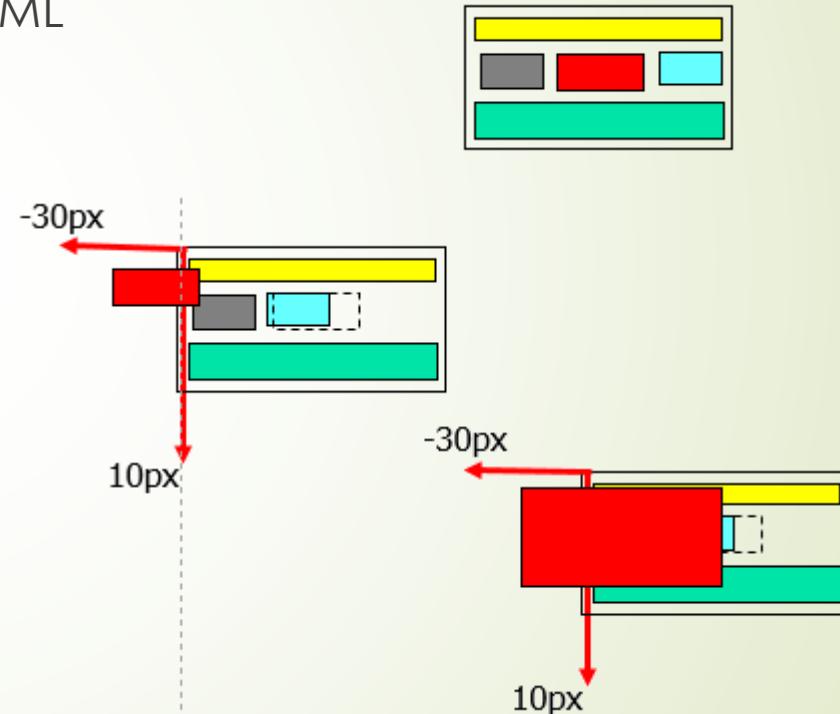
- ▶ **Statique** : flux normal du code HTML

- ▶ **Exemple 1**

```
div#bloc1 {  
    position: absolute;  
    top: 10px;  
    left: -30px; }
```

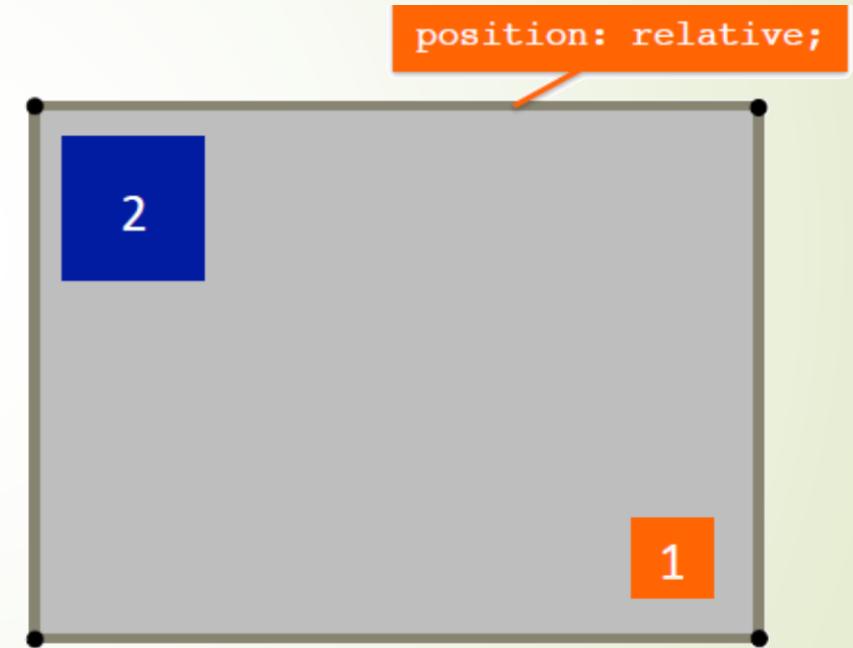
- ▶ **Exemple 2**

```
div#bloc2 {  
    position: absolute;  
    top: 10px;  
    left: -30px;  
    width: 150px;  
    height: 75px; }
```



Positionnement absolu

```
p { /* #1 */  
  position: absolute;  
  bottom: 10px;  
  right: 10px;  
}
```



Le positionnement : superposition d'élément

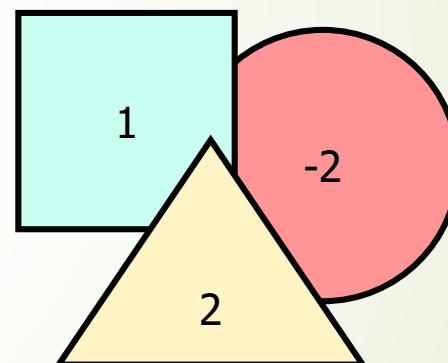
► Superposition

L'élément de **z-index** supérieur est au-dessus des autres.

```
div#cerclle {z-index: -2; ...}
```

```
div#carre{z-index: 1; ...}
```

```
div#triangle {z-index: 2; ...}
```



Propriétés de page pour l'impression

- ▶ Pages imprimées

```
@page {  
    size: landscape;  
    margin: 2cm;  
}
```

- ▶ Pseudo-classes de pages imprimées

`@page:first` → 1ère page du document

`@page:left` → pages de gauche du document

`@page:right` → pages de droite du document

- ▶ propriétés de taille (size) : portrait | landscape (paysage)

Apparition d'une image au survol dans un menu

▶ Partie HTML

```
<ul id="menu">  
    <li><a href="#">menu 1</a></li>  
    <li><a href="#">menu 2</a></li>  
    <li><a href="#">menu 3</a></li>  
</ul>
```

menu 1
menu 2
menu 3

▶ Partie CSS

```
#menu a:hover {  
    background-image: url(images/guillemet.png);  
    background-repeat: no-repeat;  
    background-position: 1% 50%;  
    padding-left: 15px;  
}
```

menu 1
menu 2
» menu 3

Application css : bouton avec relief

► Partie **HTML**

```
<input class="bouton" type="submit" value="Envoyer" />
```

► Partie **CSS**

```
input.bouton {  
    border:2px outset red;  
    font-weight:bold;  
    cursor:pointer;  
}
```

```
input.bouton:hover {  
    border:2px outset white;  
    background-color:white; color:red;  
}
```

```
input.bouton:active {  
    border:2px inset red;  
    background-color:red;  
    color:white;  
}
```



Propriété content

- ▶

```
h1::before {  
    content:"Rubrique : ";  
}
```

-> ajoute l'expression avant l'élément h1
- ▶

```
h2::after {  
    content:url(haut.gif) ;  
}
```

-> ajoute l'image après l'élément
- ▶ Ne fonctionne pas avec IE6

Feuille de style CSS externe liée : link

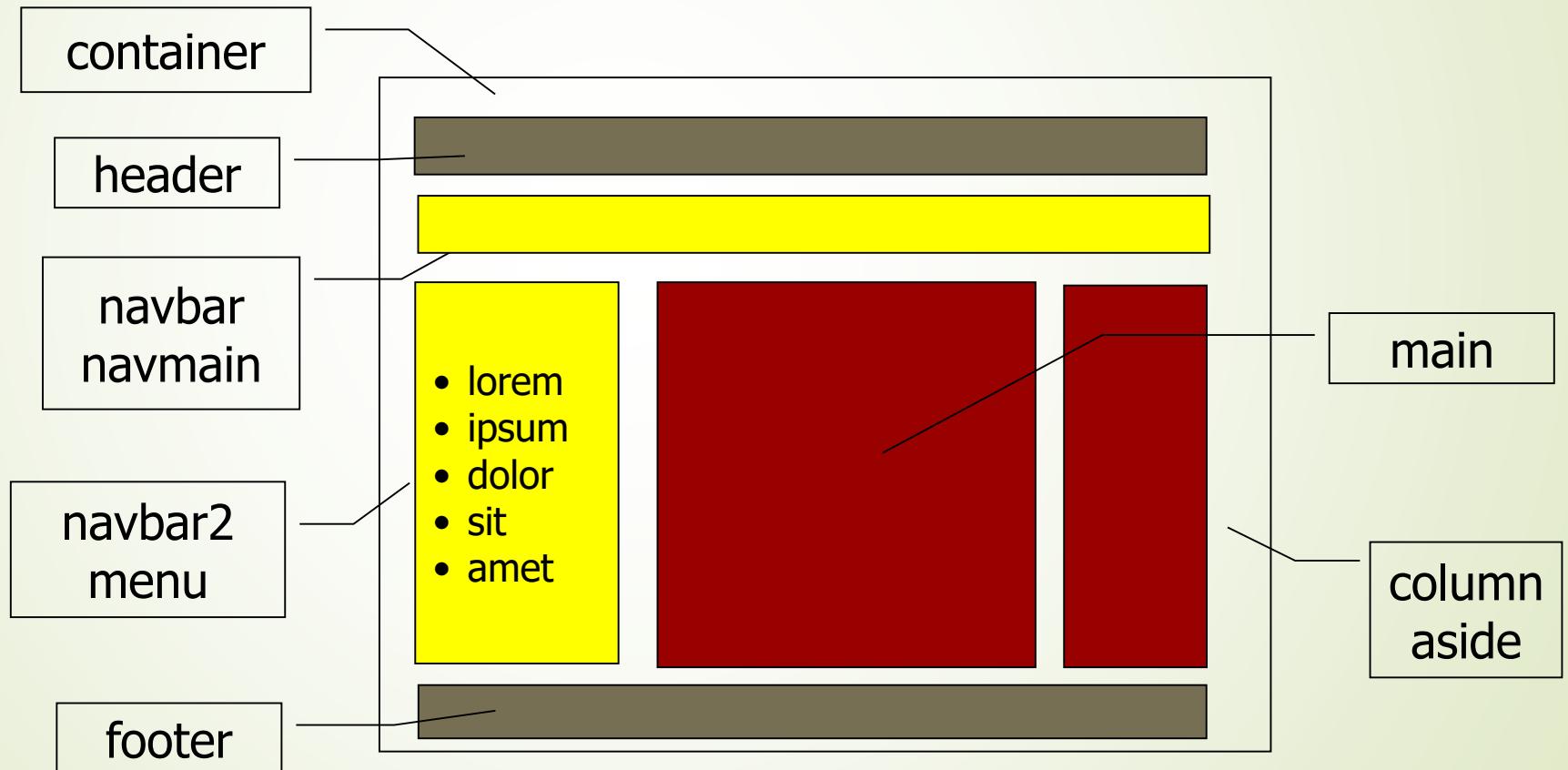
- ▶ Pour **tous périphériques** de sortie

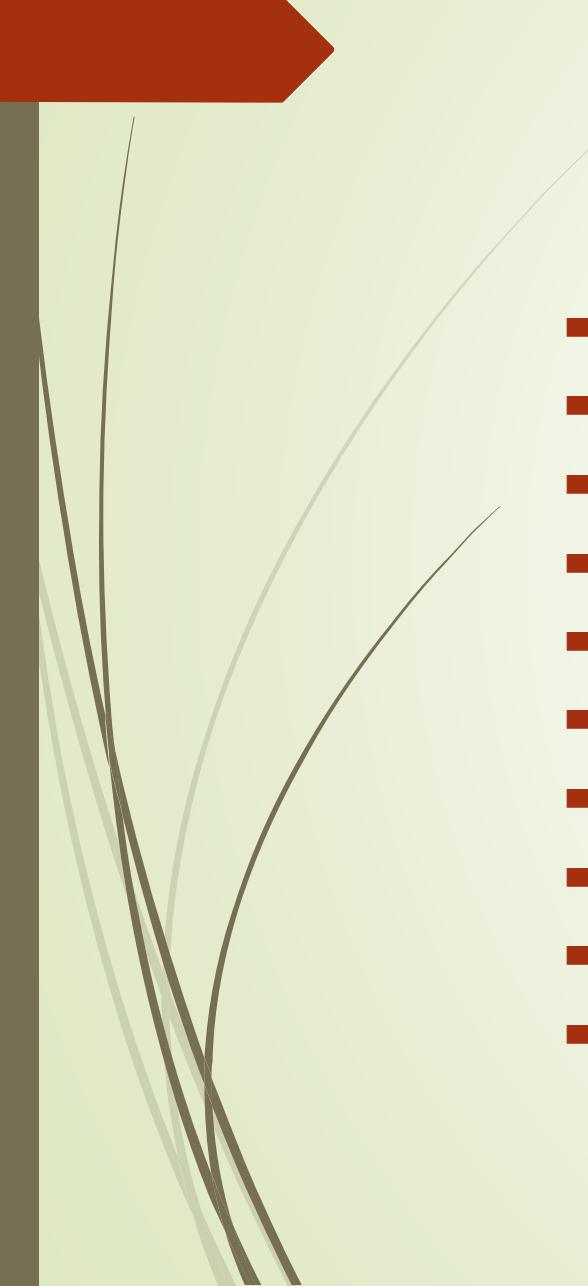
```
<link rel="stylesheet" type="text/css" href="nomdufichier1.css"  
/>>
```

- ▶ Pour **différents périphériques** de sortie

```
<link rel="stylesheet" type="text/css" href="nomdufichier1.css"  
media="screen" />  
<link rel="stylesheet" type="text/css" href="nomdufichier2.css"  
media="print" />  
<link rel="stylesheet" type="text/css" href="nomdufichier2.css"  
media="braille" />  
<link rel="stylesheet" type="text/css" href="nomdufichier2.css"  
media="handheld" />
```

Structure de page : noms usuels des zones



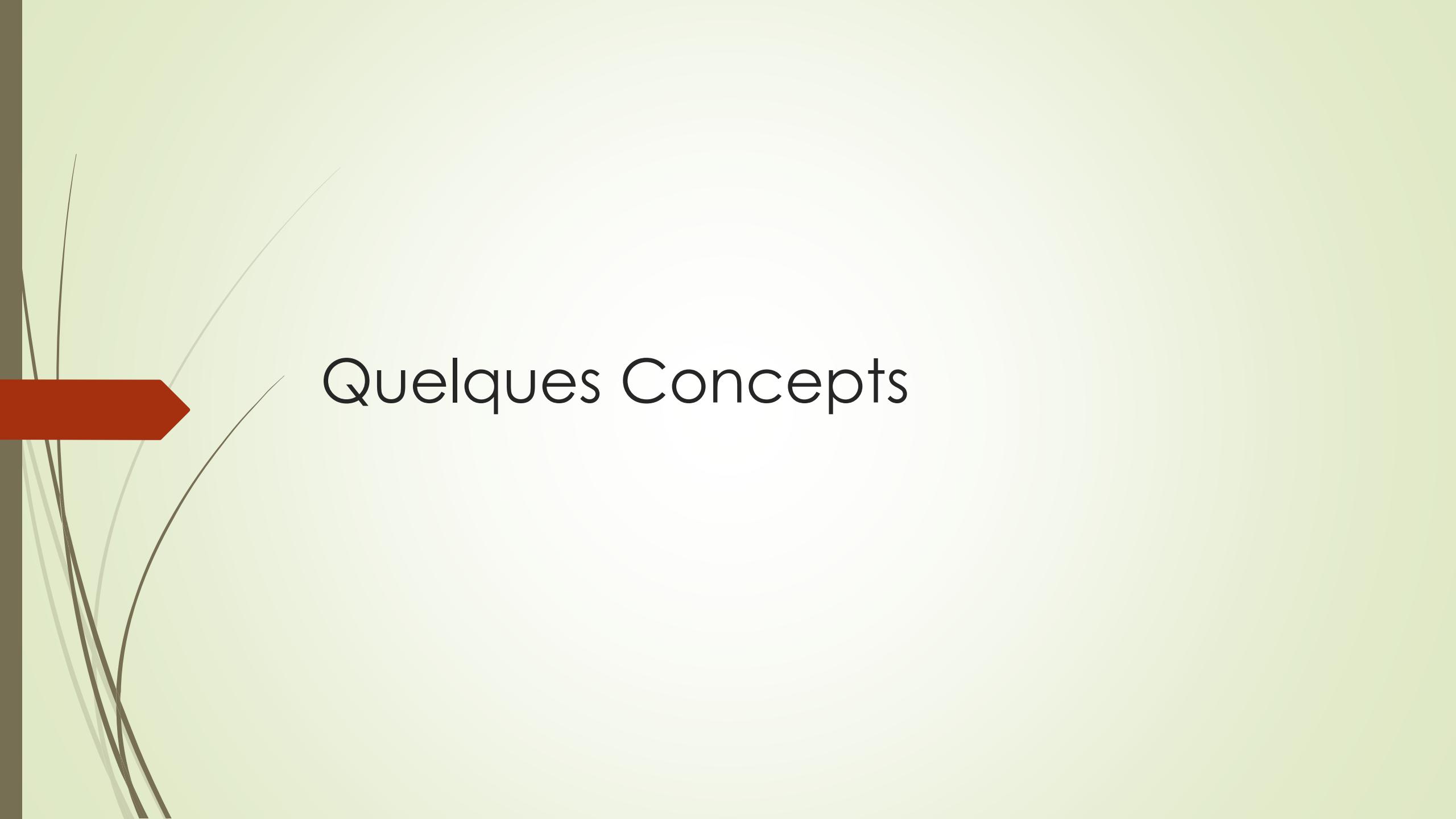


Structuration de la feuille de style : exemple d'ordre

- ▶ Reset global (ex: marge à 0...)
- ▶ Base typographique (h1, h2...)
- ▶ Formulaire
- ▶ Structure de page (container, header, main, footer)
- ▶ Structure des templates (colonne 1, colonne 2...)
- ▶ Éléments de contenu
- ▶ Éléments de navigation
- ▶ Eléments-outils ou toolbox (.clearer, .floatleft)
- ▶ Variantes par gabarit
- ▶ Surcharges pour autres medias (print, handled...)

Structuration en n feuilles de style

- ▶ Disposition de la page
`layout.css`
- ▶ Typographie de la page
`style.css`
- ▶ Typographie de la partie principale de la page
`main.css`



Quelques Concepts

Cascade

- ▶ la cascade des styles signifie que l'ordre d'apparition des règles dans le CSS a une importance ; quand deux règles applicables ont la même spécificité, c'est la dernière déclarée qui sera utilisée pour la mise en forme.
- ▶ Pour rappel, HTML est interprété séquentiellement de haut en bas

```
h1 {  
    color: red;  
}  
h1 {  
    color: blue;  
}
```

```
<h1>This is my heading.</h1>
```

This is my heading.

Feuilles de style en cascades

styl1.css

```
h1.cs {color: blue; text-align:center}
```

styl2.css

```
@import "styl1.css";      → à mettre en premier  
p.cs {color: green}
```

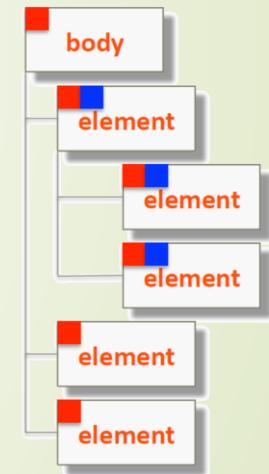
toto.htm

```
<link href="styl2.css" rel="stylesheet">  
<h1 class="cs">texte h1.cs bleu</h1>  
<p class="cs">texte p.cs vert</p>  
<p>texte</>texte p normal</p>
```

Héritage

- ▶ Pour chaque propriété CSS, la spécification indique si, par défaut, cette propriété est héritée ou non.
- ▶ Si on applique une couleur (color) au body, tous les éléments du body héritent de cette couleur.
- ▶ Pas d'héritage de margin et padding.
- ▶ On peut **forcer l'héritage** d'une propriété par défaut non héritable avec la valeur inherit : `{margin: inherit }`

DOM
Tree



Héritage - Exemple



```
body {  
    color: blue;  
}  
span {  
    color: black;  
}
```



```
<p>As the body has been set to have a color of  
blue this is inherited through the  
descendants.</p>  
<p>We can change the color by targetting the  
element with a selector, such as this  
<span>span</span>.</p>
```



As the body has been set to have a color of blue this is inherited through the descendants.

We can change the color by targetting the element with a selector, such as this span.

Spécificité

- ▶ Quand des règles avec des sélecteurs différents s'appliquent sur un même élément, le navigateur choisit la règle qui a **la plus grande spécificité**.
- ▶ La spécificité mesure essentiellement combien la sélection est précise :
 - ▶ Un sélecteur d'élément est peu spécifique — il cible tous les éléments d'un type donné dans la page — son score est donc faible ;
 - ▶ Un sélecteur de classe est plus spécifique — dans la page, il ne cible que les éléments dont l'attribut class a la valeur choisie — son score est plus important.

Spécificité - Exemple



```
.main-heading {  
    color: red;  
}  
h1 {  
    color: blue;  
}
```



```
<h1 class="main-heading">This is my heading.</h1>
```



This is my heading.

Conflits de règles : priorité à la dernière

Règle postérieure > Règle antérieure

Dans un même emplacement (fichier .css, <style>),
la dernière règle annule la règle antérieure.

```
p {font-family: arial}  
p {font-size: 10 pt}  
--> p {font-family:arial ; font-size: 10 pt ;}
```

```
p {font-family: arial ; font-size: 10 pt}  
p {font-size: 12 pt}  
--> p {font-family:arial; font-size: 12 pt ;}
```



Conflits de style : priorité à la forme la plus proche

Style intra-ligne > Style interne > Style externe (lié ou importé)

```
<style>
  p {text-align: center; font-size:10pt;}
</style>
...
<body>
  <p align="right">Texte 1</p>
  <p style="text-align:left">Texte 2</p>
  <p>Texte 3</p>
</body>
```

Conflits de style : sélecteur contextuel

```
h1, p { color: blue }  
em { color: red }  
h1 em { color: red }
```

```
<h1>Chapitre I</h1>  
<p>Texte avec <em>point  
important</em></p>  
<h1>Chapitre II avec <em>appui sur un  
thème</em></h1>
```

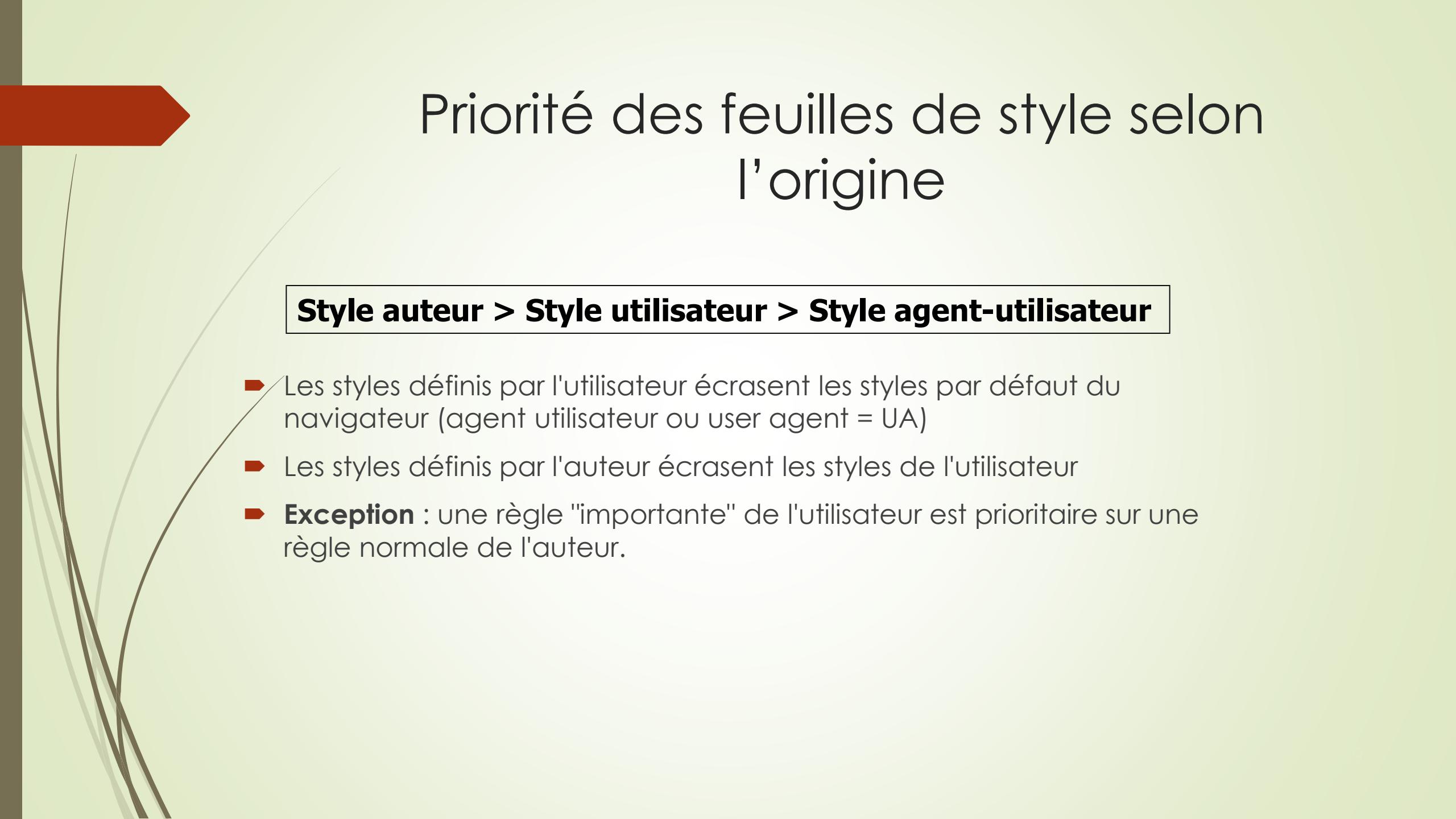
Exemples d'utilisation

```
ul li { list-style-type: disc }  
ul ul li {list-style-type: square ; font-size: 10pt }  
#menu ol a:hover {color: red; }
```

Rendre prioritaire avec !important

- ▶ L'ajout de **!important** rend prioritaire la déclaration quel que soit son emplacement.

```
h2 {color: maroon !important}  
h2 {color: yellow}  
h1 {  
    color: black !important;  
    background: white !important;  
}  
p {  
    font-size: 12pt !important;  
    font-style: italic;  
}
```



Priorité des feuilles de style selon l'origine

Style auteur > Style utilisateur > Style agent-utilisateur

- ▶ Les styles définis par l'utilisateur écrasent les styles par défaut du navigateur (agent utilisateur ou user agent = UA)
- ▶ Les styles définis par l'auteur écrasent les styles de l'utilisateur
- ▶ **Exception** : une règle "importante" de l'utilisateur est prioritaire sur une règle normale de l'auteur.

Priorité en fonction des types de sélecteurs

Sélecteur avec identifiant

> Sélecteur avec classe ou pseudo-classe

> Sélecteur contextuel > Sélecteur simple

sélecteur	déclaration	ID	classe	élément	total
		a	b	c	
li	{...}	a=0	b=0	c=1	= 001
ul li	{...}	a=0	b=0	c=2	= 002
ul ol li	{...}	a=0	b=0	c=3	= 003
.cl	{...}	a=0	b=1	c=0	= 010
li.cl	{...}	a=0	b=1	c=1	= 011
ul ol li.cl	{...}	a=0	b=1	c=3	= 013
#toto	{...}	a=1	b=0	c=0	= 100

Le modèle de boîte (The Box Model)

- ▶ En CSS, tout élément est inclus dans une boîte ("box" en anglais)
- ▶ Il existe deux types de boîtes : les boîtes **en bloc** ("block boxes" en anglais) et les boîtes **en ligne** ("inline boxes").





Boite en bloc

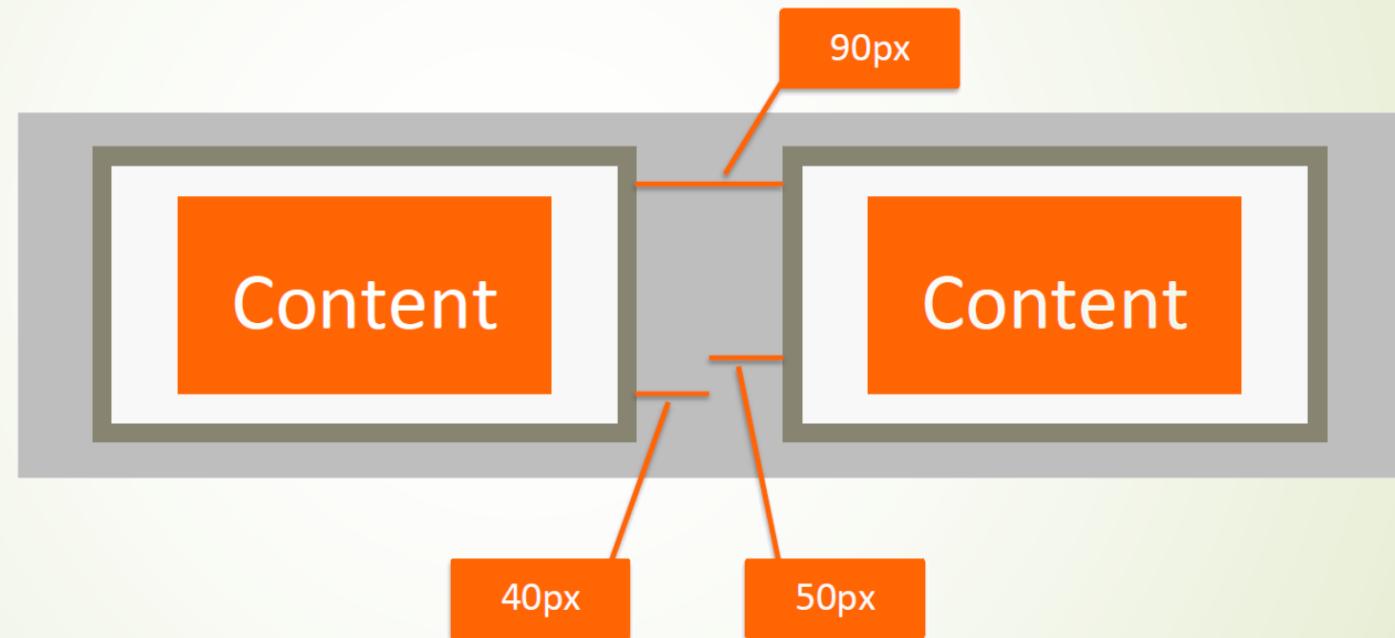
- ▶ La boîte s'étend en largeur pour remplir totalement l'espace offert par son conteneur. Dans la plupart des cas, la boîte devient alors **aussi large que son conteneur**, occupant 100% de l'espace disponible.
- ▶ La boîte occupe sa propre nouvelle ligne et créé un retour à la ligne, faisant ainsi passer les éléments suivants à la ligne d'après.
- ▶ Les propriétés de largeur (width) et de hauteur (height) sont toujours respectées.
- ▶ Les propriétés padding, margin et border — correspondantes respectivement aux écarts de remplissage interne, externe et à la bordure de la boîte — auront pour effet de repousser les autres éléments.
- ▶ Les titres (`<h1>`, `<h2>`, etc.) et les paragraphes (`<p>`) utilisent le mode "bloc" comme propriété de positionnement extérieur par défaut.



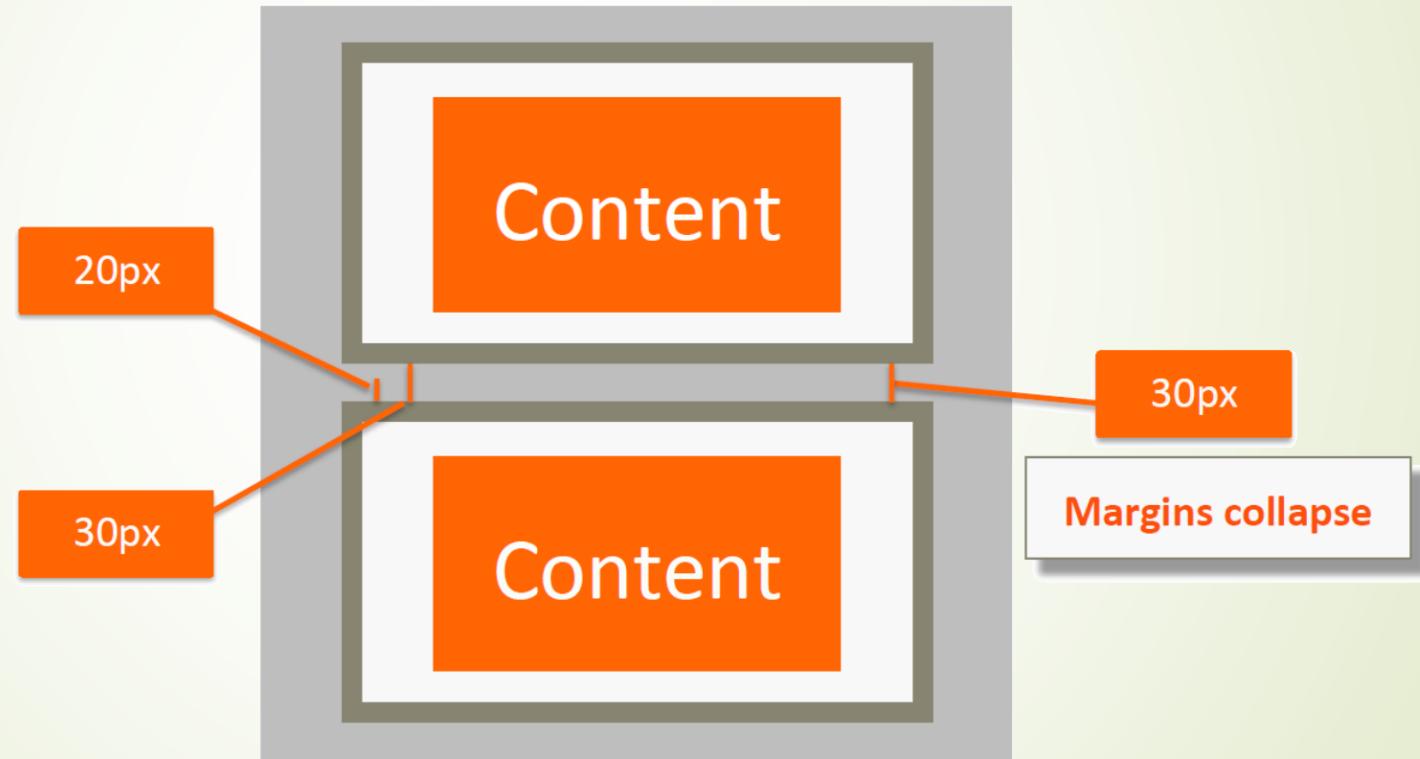
Boite en ligne

- ▶ La boîte **ne crée pas de retour à la ligne**, les autres éléments se suivent donc en ligne.
- ▶ Les propriétés de largeur (width) et de hauteur (height) ne s'appliquent pas.
- ▶ Les propriétés padding, margin et border seront appliquées mais ne provoqueront pas de déplacement des éléments alentours.
- ▶ Les éléments <a>, utilisés pour les liens, ou encore , et sont tous des éléments qui s'affichent "en ligne" par défaut.

Marge et boîte en ligne



Marge et boite en bloc





Méthodes pour appliquer des styles en HTML



Méthodes pour appliquer des styles en HTML

- ▶ **Feuille de style interne au document**

- ▶ liste des règles CSS dans balise `<style...></style>`
 - ▶ placée dans la partie `<head>`

- ▶ **Feuille de style externe liée**

- ▶ liste des règles CSS dans un fichier externe `.css`
 - ▶ appel à ce fichier externe avec une balise `<link ... />`

- ▶ **Feuille de style externe importée**

- ▶ liste des règles CSS dans un fichier externe `.css`
 - ▶ appel à un fichier externe avec `@import`

- ▶ **Style en ligne (intra-ligne ou incorporé)**

- ▶ attribut `style="..."` dans la balise d'un élément

Feuille de style interne au document

- ▶ Balise `<style type="text/css"></style>` placée dans l'en-tête du document
- ▶ Liste de règles CSS placée dans cette balise ouvrante et fermante
- ▶ Règle appliquée à tous les sélecteurs correspondant du document
- ▶ Utilisation pour un document isolé, éventuellement pour une page particulière d'un site
- ▶ Utilisation ne convenant pas pour mettre en forme un site entier

Exemple

```
<head>
  <style type="text/css">
    body {background-color: silver; color: maroon;}
    h1 {text-align: center; font-size: 1.5em; color: black;}
  </style>
</head>
<body>
  <h1>Titre A</h1>
  <p>Texte 1</p>
  <p>Texte 2</p>
</body>
```

Feuille de style CSS externe liée

- ▶ Liste de règles CSS placée dans un fichier CSS externe (extension css)
- ▶ Balise **link** à ajouter dans l'en-tête du document html permettant de faire un "lien" vers ce fichier css
- ▶ Utilisation bien adaptée à un site web dont toutes les pages HTML comporteront la balise <link>
- ▶ Règle appliquée à tous les sélecteurs correspondant dans les documents liés

Exemple

1. Code CSS : contenu d'un fichier CSS nommé monstyle.css

```
body { background-color: white ;  
      color: blue;  
}  
h1 { text-align: center;  
     font-size: 1.5em ;  
     color: black;  
}
```

2. Code HTML (extrait du fichier)

```
...  
<head><title>...</title>  
<link rel="stylesheet" type="text/css" href="chemin/monstyle.css">  
</head>  
...
```

Feuille de style CSS importée

- ▶ Liste de règles CSS dans un fichier CSS externe
- ▶ Règle **@import** à ajouter dans la feuille de style (interne ou externe)
- ▶ Se place avant tout autre déclaration de style
- ▶ Règle appliquée à tous les sélecteurs correspondant dans les documents liés
- ▶ Utilisation adaptée à un site web, notamment pour "cascader les styles"
- ▶ Syntaxe :
 - ▶ `@import "fichier.css"`
 - ▶ `@import url("fichier.css")`

Exemple

```
<style type="text/css">
    @import url("fichier.css")
    p {
        color:red;
    }
</style>
```

Style en ligne, dans une balise HTML

- ▶ Bloc de déclarations CSS
 - ▶ incorporé dans la balise
 - ▶ introduit par l'attribut **style**
- ▶ Exemple :
`<p style="text-align: center; background-color:#ccc;">Texte</p>`
- ▶ S'applique à une balise précise du document
- ▶ Intéressant pour un essai lors du développement
- ▶ Déconseillé actuellement : à remplacer plutôt par l'utilisation d'un identifiant

HTML : `<element id="truc">`

CSS : `#truc {text-align: center; background-color:#ccc; }`

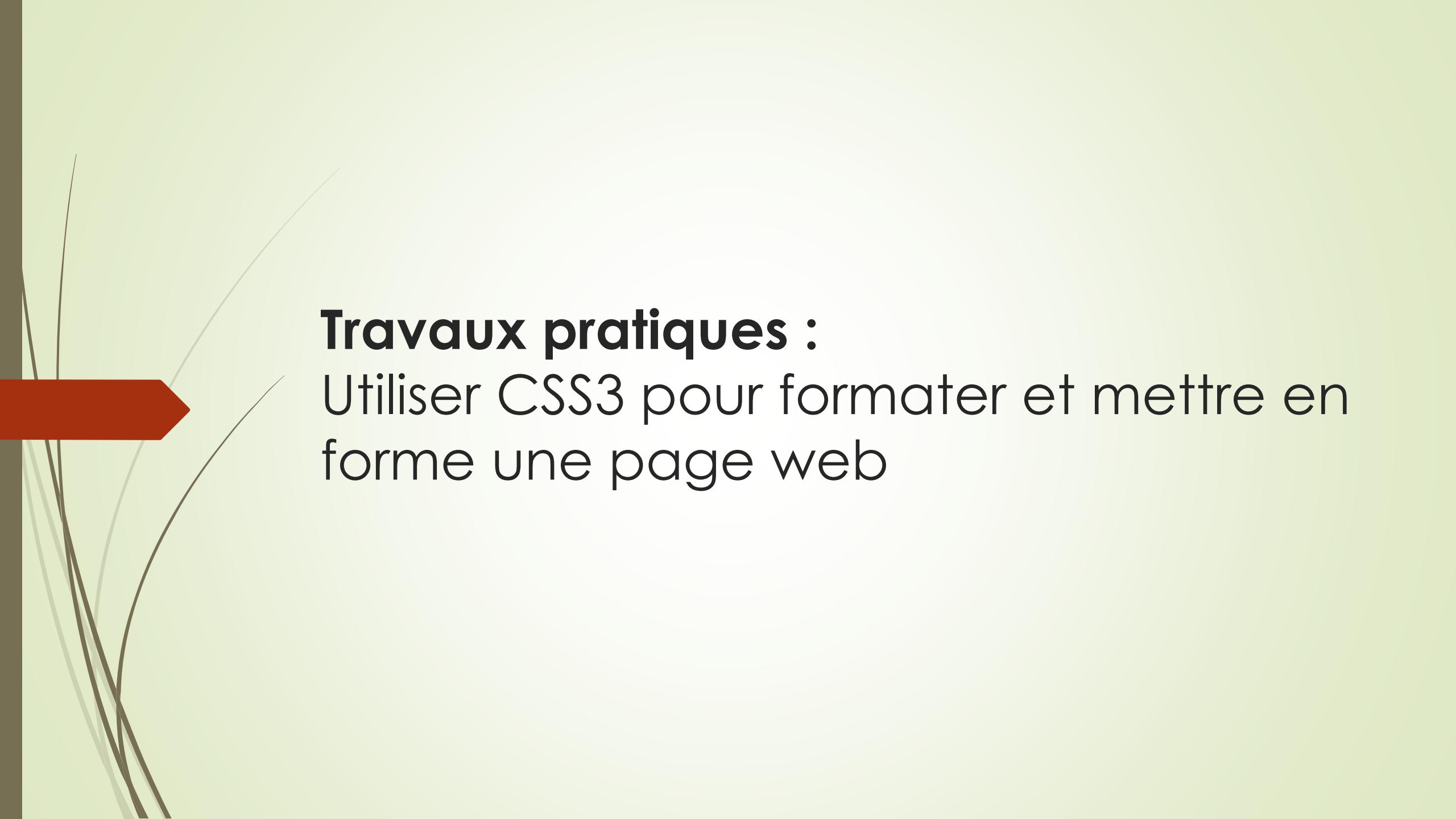
4 manières d'appliquer un style : exemple

```
<html>
<head>
<title>...</title>
<link rel="stylesheet" type="text/css" href="fichier1.css" />
<style type="text/css">
    @import url("fichier2.css");
    h1 { color: blue }
</style>
</head>
<body>
    <h1>Ce texte est en bleu à cause du style interne</h1>
    <p style="color: green" >Ce paragraphe est en vert.</p>
</body>
</html>
```

Pour déboguer une page HTML/CSS

- ▶ Passez le code HTML au validateur HTML : <https://validator.w3.org/>
- ▶ Passez le code CSS au validateur CSS : <http://www.css-validator.org/>
- ▶ Ajoutez au début du CSS la règle :

```
* {  
    margin : 0;  
    padding : 0;  
}
```
- ▶ Appliquez une couleur de fond aux éléments posant problème.
- ▶ Placez en commentaires /* ... */ les propriétés susceptibles de poser problème.
- ▶ Essayez dans différents navigateurs et différentes versions.
- ▶ Recherchez sur le web ou dans des ouvrages si le problème est connu.



Travaux pratiques :
Utiliser CSS3 pour formater et mettre en forme une page web

Exemple

The screenshot shows a blog interface. At the top, there's a dark header with the word "BLOG" on the left and a search bar on the right containing the placeholder "Search here...". Below the header, a navigation menu includes links for "HOME", "ARCHIVE", "BOOKS", "ABOUT", and "CONTACT". The main content area features a large, bold title: "Useful Talks & Videos for Mastering CSS". Below the title, a smaller text line reads "By John Doe on January, 10 2014". At the bottom of the content area is a horizontal image of a sunset or sunrise through bare tree branches.

BLOG

Search here...

HOME ARCHIVE BOOKS ABOUT CONTACT

Useful Talks & Videos for
Mastering CSS

By John Doe on January, 10 2014



Responsive Web Design

Généralités



Variabilité des résolutions d'écran

- ▶ Une variabilité des résolutions d'écran
 - ▶ Plus de 2000 types de résolution différentes !
 - ▶ De moins en moins de standard
 - ▶ Des proportions variables (16/9, 4/3, etc...)
 - ▶ Donc impossible de choisir une résolution spécifique



Common Screen Resolutions 2019

source: w3counter.com

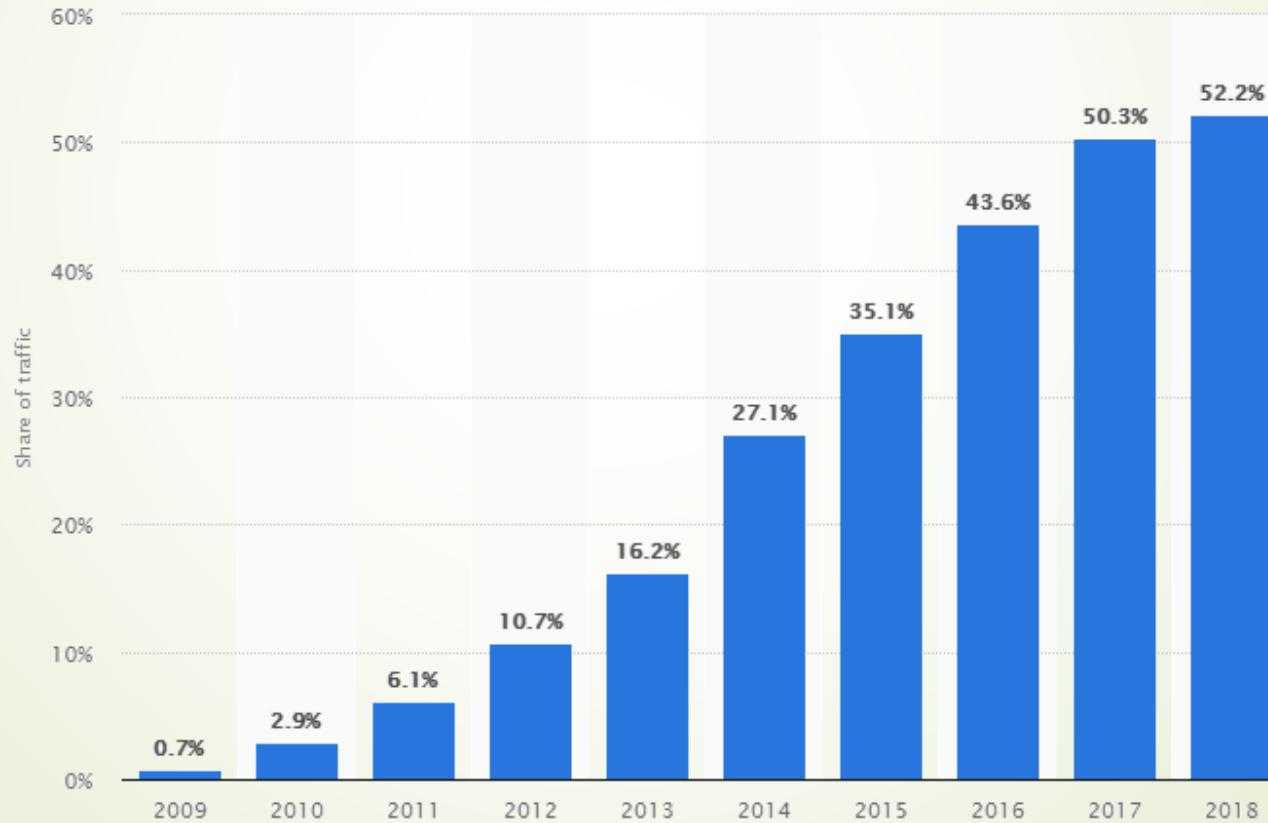
- 640x360
- 1366x768
- 1920x1080
- 1024x768
- 667x375
- 720x360
- 736x414
- 1440x900
- 1280x800
- 1600x900



Progression des supports mobiles

- ▶ Les supports mobiles (tablettes et smartphones) en forte progression

Percentage of all global web pages served to mobile |





Quelles solutions pour les développeurs Web ?

► La mauvaise solution

- ▶ Créer plusieurs sites Web spécifiques à chaque support
- ▶ Coût important de développement et de maintenance

► La bonne solution

- ▶ Penser « Mobile First »
- ▶ 1 seule URL, 1 contenu identique, 1 code unique (HTML5), des règles CSS3
- ▶ C'est le « responsive design »

Site Web responsive

- ▶ Définition : site Web responsive est un site Web dont la conception vise, grâce à différents principes et techniques, à offrir une consultation confortable sur des écrans de tailles très différentes.

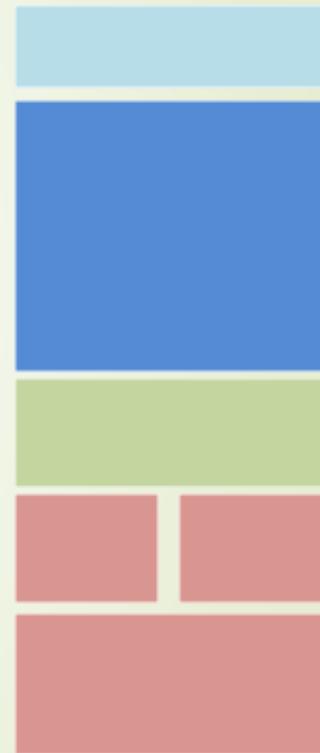
-Wikipédia
- ▶ L'utilisateur peut ainsi consulter le même site web à travers une large gamme d'appareils (moniteurs d'ordinateur, smartphone, tablettes, télévision...) avec le même confort visuel et sans avoir recours au défilement horizontal ou au zoom avant/arrière sur les appareils tactiles notamment, manipulations qui peuvent parfois dégrader l'expérience utilisateur, tant en lecture qu'en navigation.
- ▶ Les meilleurs exemples de web design : <http://mediaqueri.es/>

Les points clés du responsive design

- ▶ Une grille flexible
- ▶ Un contenu flexible
- ▶ Un contenu adaptable à la taille de l'écran



Largeur classique (« desktop »)

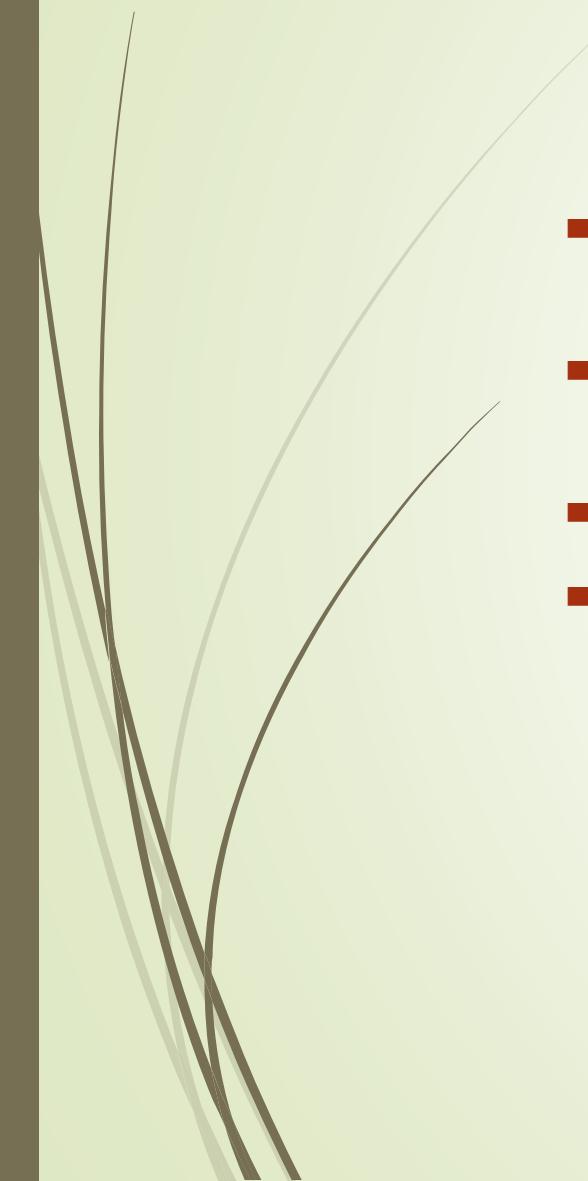


Largeur réduite (« mobile »)



Les avantages du responsive design

- ▶ Gain de temps de développement
- ▶ Economies
- ▶ Meilleures performances
- ▶ Supporte tous les navigateurs modernes
- ▶ Améliore le référencement



Balise meta viewport

- ▶ Le viewport représente de manière schématique la partie visible d'une page web par un utilisateur ou la fenêtre active
- ▶ La taille de cette fenêtre va bien évidemment varier en fonction de la taille de l'écran de l'utilisateur et de l'appareil utilisé
- ▶ **<meta name="viewport" content="width=device-width, initial-scale=1">**
- ▶ Ici, nous définirons une largeur de viewport égale à la largeur de l'appareil dans le viewport ainsi qu'un niveau de zoom initial égal à 1

Quelques exemples

- ▶ **<meta name="viewport" content="width=device-width, maximum-scale=3, minimum-scale=0.5" />**
 - ▶ Contrôle le niveau de zoom : cet exemple autorise les utilisateurs à zoomer jusqu'à 3 fois (zoom) la largeur d'écran et 0,5 fois (dézoom) largeur d'écran.
- ▶ **<meta name="viewport" content="initial-scale=1.0, user-scalable=no" />**
 - ▶ Nous interdirons les utilisateurs de zoomer ou de dézoomer.
- ▶ **<meta name="viewport" content="width=device-width, initial-scale=1.0"/>**
 - ▶ C'est utilisé dans la plupart de cas

Taille d'écran & viewport

- ▶ Taille d'écran est la taille de son écran réel
- ▶ Le viewport représente de manière schématique la partie visible d'une page web par un utilisateur ou la fenêtre active



Appliquer les media queries

- ▶ 3 solutions pour la mise en place

1. Plusieurs fichiers CSS spécifiques

```
<link rel="stylesheet" href="style.css" />
<!-- Pour tout le monde --&gt;
&lt;link rel="stylesheet" media="screen and (max-width: 1280px)"
href="petite_resolution.css" /&gt;
<!-- Pour ceux qui ont une résolution inférieure à 1280px --&gt;</pre>
```

2. Des règles dans un fichier CSS

```
@media screen and (max-width: 1280px)
{ /* Rédigez vos propriétés CSS ici */ }
```

3. dans @import

```
@import url(smallstyle.css) screen and (min-width:600px)
```

Mise en place des media queries

- De nombreuses règles dont voici les principales :

Nom	Résumé
width	La largeur de la zone d'affichage (viewport)
height	La hauteur de la zone d'affichage
aspect-ratio	Le rapport largeur/hauteur de la zone d'affichage
orientation	L'orientation la zone d'affichage
resolution	La densité de pixel pour l'appareil d'affichage
color	Le nombre de bits pour chaque composante de couleur pour l'appareil d'affichage (ou 0 si l'appareil ne gère pas les couleurs)
media	Type de média : all, screen, print, speech

Mise en place des media queries

- ▶ **Types 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. Pour plus d'informations, voir l'article sur les média paginés.
- ▶ **screen**
 - ▶ Correspond aux appareils dotés d'un écran.
- ▶ **speech**
 - ▶ Correspond aux outils de synthèse vocale.

Combinaison des règles

► A l'aide des mots clés suivants :

- only : « uniquement »
- and : « et »
- not : « non »

Combinaison des règles

► Exemples :

```
/* Sur les écrans, quand la largeur de la fenêtre fait au maximum  
1280px */
```

```
@media screen and (max-width: 1280px)
```

```
/* Sur tous types d'écran, quand la largeur de la fenêtre est  
comprise entre 1024px et 1280px */
```

```
@media all and (min-width: 1024px) and (max-width: 1280px)
```

```
/* Sur tous types d'écrans orientés verticalement */
```

```
@media all and (orientation: portrait)
```

Exemple

```
/* Paragraphes en bleu par défaut */  
p { color: blue; }
```

```
/* Nouvelles règles si la fenêtre  
fait au plus 1024px de large */  
@media screen and (max-width: 1024px)  
{ p  
    { color: red;  
        background-color: black;  
        font-size: 1.2em;  
    }  
}
```





Responsive Web Design

Techniques



Responsive image/video sans media queries

- ▶ Création d'une image/video
- ▶ Exemple : [image resizing](#)
- ▶ Exemple : [video resizing](#)



Responsive image/video avec media queries

- Exemple : [Resizing using media queries](#)



Responsive menu avec media queries

- ▶ Exemple : [navigation](#)

Exemples

- ▶ [Layout Responsive avec min et max width](#)
- ▶ [Controler layout avec relative padding](#)
- ▶ [Ajouter un média query dans css](#)
- ▶ [Création d'un layout avec media queries](#)
- ▶ [Modifier la taille d'image](#)
- ▶ [Cacher un élément](#)
- ▶ [Exemple complet](#)

En savoir plus sur les media queries

- ▶ Documentation officielle
 - <http://www.w3.org/TR/css3-mediaqueries/>
- ▶ Exemples
 - <http://mediaqueri.es/>



Bootstrap

Bibliothèque CSS

Bootstrap

- ▶ Développé au sein de l'équipe de Twitter
- ▶ Sous licence Apache depuis aout 2011
- ▶ Un kit facile d'utilisation qui propose un bon nombre de fonctionnalités
- ▶ Une bibliothèque totalement open source
- ▶ Une mise en page basée sur une grille de 12 colonnes
- ▶ Du code responsive design fondé sur HTML5 et CSS3
- ▶ Des plugins jQuery de qualité
- ▶ Une garantie de compatibilité maximale sur tous les navigateurs récents
- ▶ Une bonne documentation

Une grille de 12 colonnes

- ▶ Une classe CSS row pour définir des lignes
- ▶ Des classes CSS span1, span2,..., span12 pour définir des colonnes.
- ▶ La version 3 de Bootstrap propose un système de 4 grilles :
 - ▶ xs : Extra small devices ~ Phones (< 768px)
 - ▶ sm : Small devices ~ Tablets (>= 768px)
 - ▶ md : Medium devices ~ Desktops (>= 992px)
 - ▶ lg : Large devices ~ Desktops (>= 1200px)

La grille adaptative

- ▶ Concept issu de l'imprimerie
- ▶ Produit une mise en page bien structurée et harmonieuse
- ▶ Sert d'armature pour organiser la page et son contenu
- ▶ Favorise l'intégration CSS
- ▶ Très pratique en production.



Une grille de 12 colonnes

```
<div class="container">  
  <div class="row">  
    <div class="col-xs-4 col-1d-6">Colonne 1</div>  
    <div class="col-xs-8 col-1d-6">Colonne 2</div>  
  </div>  
</div>
```





Eléments de base de Bootstrap

- ▶ De nombreux éléments de style pour
 - ▶ Les listes à puces
 - ▶ Les tableaux
 - ▶ Les boutons
 - ▶ Les formulaires
 - ▶ Les images

Des composants intégrés

- ▶ Sans javascript
 - ▶ Barre de navigation
 - ▶ Effets typographiques
 - ▶ Onglets (ou Thumbnails)
- ▶ Avec Javascript
 - ▶ Alertes
 - ▶ Barre de progression
- ▶ Consulter le site de Bootstrap pour tous les exemples et la documentation détaillée : <http://getbootstrap.com>
- ▶ Exemple : [bootstrap examples](#)



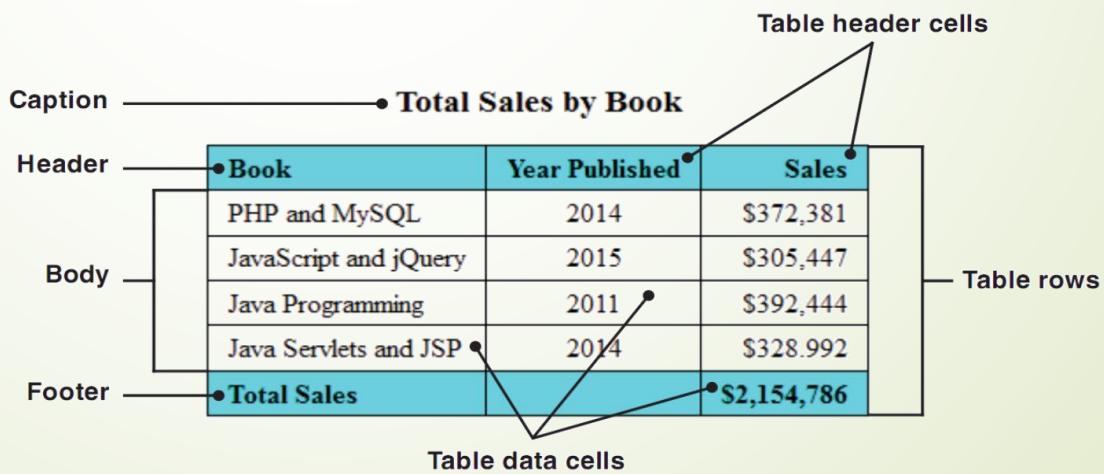
Tableau HTML

Création et application du style CSS

Tableau (Table)

► L'élément HTML <table>

- ▶ permet de représenter un tableau de données, c'est-à-dire des informations exprimées sur un tableau en deux dimensions.
- ▶ est un ensemble structuré de données (**table de données**) présentées en lignes et colonnes.
- ▶ **ligne** est une ligne d'information horizontale
- ▶ **colonne** est une ligne d'information verticale
- ▶ **cellule** est l'intersection d'une ligne et une colonne et contient une donnée



Éléments de tableau

Élément	Indique le début et la fin de	Contient
<table> ... </table>	un tableau de données dans une page web	Tous les éléments liés à un tableau
<tr> ... </tr>	une ligne de cellules dans un tableau	Cellules de données et d'en-tête
<th> ... </th>	une cellule d'un tableau comme une cellule d'en-tête	Contenu d'en-tête
<td> ... </td>	une cellule d'un tableau qui contient des données	Contenu de cellule de donnée

Exemple

Éléments de tableau

Élément	Indique le début et la fin de	Contient
<caption> ... </caption>	la légende (ou le titre) d'un tableau	la légende (ou le titre) d'un tableau
<thead> ... </thead >	L'en-tête d'un tableau	un ou plusieurs éléments d'en-tête
<tbody> ... </tbody >	le corps d'un tableau	un ou plusieurs éléments de corps
<tfoot> ... </tfoot >	le pied de page d'un tableau	un ou plusieurs éléments de pied de page

Exemple

Propriétés CSS

- ▶ border-collapse : collapse | separate
 - ▶ détermine si les bordures d'un tableau sont séparées ou fusionnées.
 - ▶ Quand elles sont séparées, chaque cellule du tableau a ses propres bordures, distinctes.
 - ▶ Quand elles sont fusionnées, les bordures des cellules sont partagées.
- ▶ Border-spacing : n
 - ▶ définit la distance qu'il y a entre les bordures de cellules adjacentes d'un tableau (uniquement lorsque border-collapse vaut separate).
- ▶ Padding : n
 - ▶ permet de définir les différents écarts de remplissage sur les quatre côtés d'un élément

Propriétés CSS

- ▶ **Text-align : left | right | center | justify**
 - ▶ définit l'alignement horizontal d'un élément de bloc ou de la boîte d'une cellule de tableau
- ▶ **Vertical-align : baseline | sub | super | text-top | text-bottom | middle | top | bottom**
 - ▶ définit l'alignement vertical d'une boîte en ligne (inline) ou d'une cellule de tableau

Exemple

Pseudo-Classes CSS3

► Sélecteurs

syntaxe	description
:nth-child(n)	Nième enfant de l 'élément parent
:nth-last-child(n)	Nième enfant de l 'élément parent (le comptage s'effectue depuis la fin)
:nth-of-type(n)	Nième enfant de l 'élément parent d'un type donné
:nth-last-of-type(n)	Nième enfant de l 'élément parent d'un type donné (le comptage s'effectue depuis la fin)

► Valeur de n

valeur	description
odd	Tous les enfants impair
even	Tous les enfants pair
n	Nème enfant
2n	Similaire à even
3n	Tous les 3 ^{ème} enfants (3,6,9,...)
2n+1	Similaire à odd
3n+1	Tous les 3 ^{ème} enfants commençant à 1 (1,4,7, ...)

[Exemple](#)

Fusion des cellules

- ▶ Attributs de <th>, <td>
- ▶ Colspan
 - ▶ Cet attribut contient un entier positif indiquant le nombre de colonnes sur lesquelles s'étend la cellule
 - ▶ La valeur par défaut est 1
- ▶ Rowspan
 - ▶ Cet attribut contient un entier positif indiquant sur combien de lignes s'étend la cellule.
 - ▶ La valeur par défaut est 1

Exemple

Accessibilité

- ▶ Tableaux sont difficiles à déchiffrer pour les personnes malvoyantes
- ▶ Caption
 - ▶ représente la légende (ou le titre) d'un tableau.
 - ▶ Il doit être le première élément parmi les descendants de l'élément `<table>`
 - ▶ une description claire et concise permet aux utilisateurs de décider s'ils doivent lire le contenu du tableau ou le passer
- ▶ Attributs pour améliorer l'accessibilité
 - ▶ Headers : list<String>
 - ▶ Cet attribut est une liste de chaînes de caractères séparées par des espace.
 - ▶ Chacune correspond à l'attribut id de l'élément `<th>` qui s'applique à cet élément
 - ▶ Scope : row | col | rowgroup | colgroup
 - ▶ Cet attribut référence les cellules auxquelles l'élément `<th>` est lié

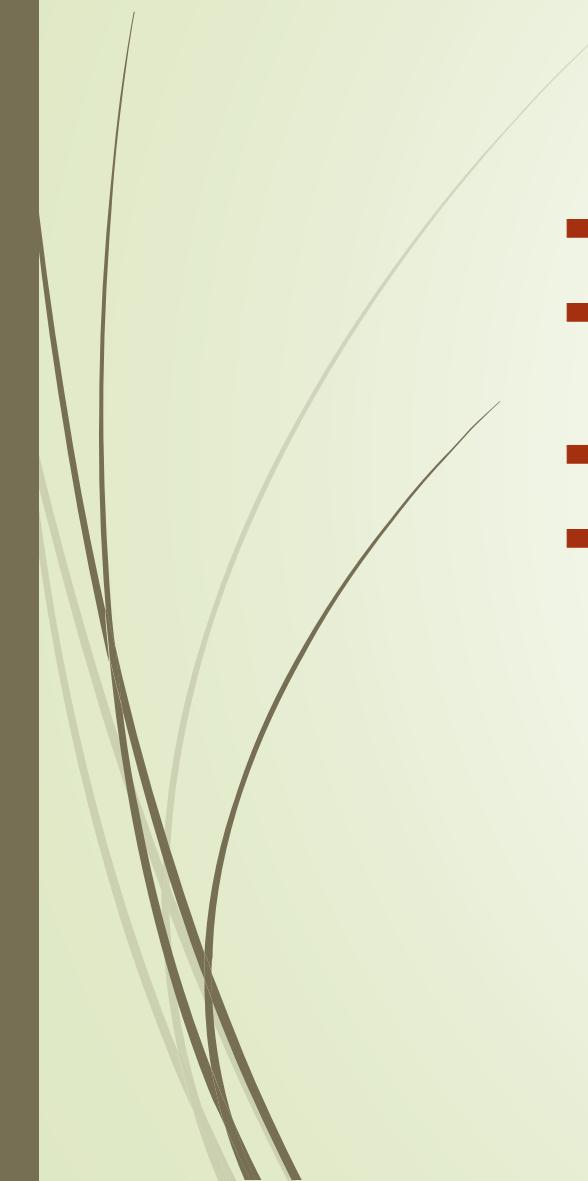
[Exemple](#)

Tableaux imbriqués

- ▶ Il est possible d'inclure un tableau dans un autre, à condition d'en spécifier la structure complète, y compris l'élément <table>
- ▶ [Exemple](#)

Contrôler le passage à la ligne

- ▶ Par défaut, le contenu d'une cellule passe à la ligne automatiquement. Les passages à la ligne sont faits naturellement pour remplir les boîtes.
- ▶ La propriété qui stoppe le passage à la ligne est la suivante:
 - ▶ Table { white-space: nowrap }
- ▶ [Exemple](#)



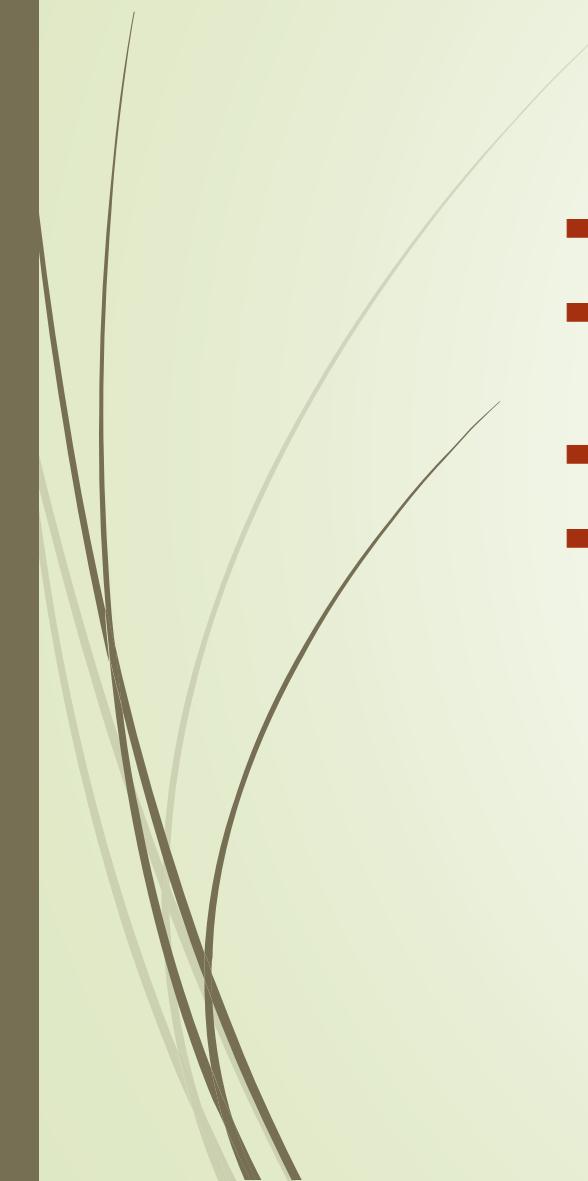
Utilisation des tableaux

- ▶ Les tableaux HTML ne doivent être utilisés que pour des données tabulaires
- ▶ Utiliser les tableaux pour la mise en page au lieu des techniques des CSS est une mauvaise idée
- ▶ L'avantage du tableau tient dans sa rigueur
- ▶ L'information est facilement interprétée par des associations visuelles entre les en-têtes de lignes et colonnes



Formulaire HTML

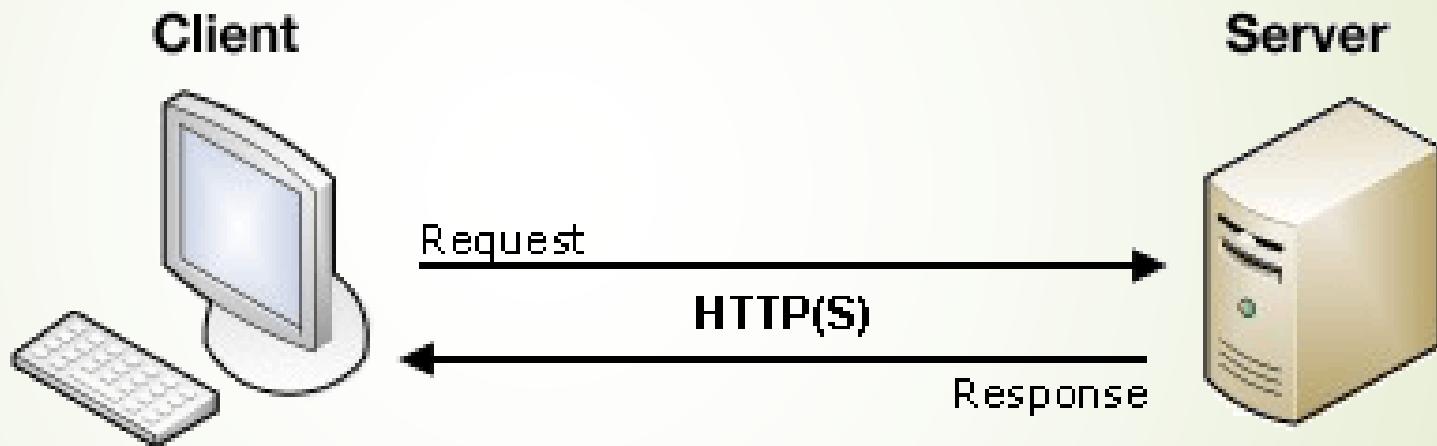
Création et application du style CSS



Formulaire HTML

- ▶ On utilise les formulaires HTML pour créer des pages Web dynamiques
- ▶ Ils sont un des principaux vecteurs d'interaction entre un utilisateur et un site Web ou une application
- ▶ Ils permettent à l'utilisateur d'envoyer des données au site Web
- ▶ La plupart du temps, ces données sont envoyées à des serveurs Web mais la page peut aussi les intercepter et les utiliser elle-même

Architecture client/server



Création d'un formulaire

- ▶ Tous les formulaires HTML débutent par un élément **<form>**

Attributs d'un formulaire :

- ▶ Action : L'attribut **action** définit l'emplacement (une URL) où doivent être envoyées les données collectées par le formulaire.
- ▶ Method : L'attribut **method** définit la méthode HTTP utilisée pour envoyer les données (cela peut être « get » ou « post »).
- ▶ Name : L'attribut **name** définit le nom du formulaire.
- ▶ Enctype : Lorsque la valeur de l'attribut method est post, cet attribut définit le type MIME qui sera utilisé pour encoder les données envoyées au serveur. C'est un attribut énuméré qui peut prendre les valeurs suivantes :
 - ▶ application/x-www-form-urlencoded : la valeur par défaut si l'attribut n'est pas défini
 - ▶ multipart/form-data : la valeur utilisée par un élément **<input>** dont l'attribut type vaut "file"

Élément <Input>

- ▶ L'élément HTML <input> est utilisé pour créer **un contrôle interactif dans un formulaire** Web qui permet à l'utilisateur de saisir des données.
- ▶ Les saisies possibles et le comportement de l'élément <input> dépend fortement de la valeur indiquée dans son attribut type.

Les différents types de champs <input>

Les types de champs disponibles sont :

- ▶ **button** : un bouton sans comportement défini.
- ▶ **checkbox** : une case à cocher qui permet de sélectionner/désélectionner une valeur
- ▶ **color** : un contrôle HTML5 qui permet de choisir une couleur.
- ▶ **date** : un contrôle HTML5 qui permet de saisir une date (composé d'un jour, d'un mois et d'une année).
- ▶ **datetime-local** : un contrôle HTML5 qui permet de saisir une date et une heure (sans fuseau horaire).
- ▶ **email** : un champ HTML5 qui permet de saisir une adresse électronique.
- ▶ **file** : un contrôle qui permet de sélectionner un fichier. L'attribut accept définit les types de fichiers qui peuvent être sélectionnés.

Les différents types de champs <input>

Les types de champs disponibles sont :

- ▶ **hidden** : un contrôle qui n'est pas affiché mais dont la valeur est envoyée au serveur.
- ▶ **image** : un bouton graphique d'envoi du formulaire. L'attribut src doit être défini avec la source de l'image et l'attribut alt doit être défini avec le texte alternatif. Les attributs height et width permettent de définir la taille de l'image en pixels.
- ▶ **month** : un contrôle HTML5 qui permet de saisir un mois et une année (sans fuseau horaire).
- ▶ **number** : un contrôle HTML5 qui permet de saisir un nombre.
- ▶ **password** : un champ texte sur une seule ligne dont la valeur est masquée. Les attributs maxlength et minlength définissent la taille maximale et minimale de la valeur à saisir dans le champ.

Les différents types de champs <input>

Les types de champs disponibles sont :

- ▶ **radio** : un bouton radio qui permet de sélectionner une seule valeur parmi un groupe de différentes valeurs.
- ▶ **range** : un contrôle HTML5 qui permet de saisir un nombre dont la valeur exacte n'est pas importante.
- ▶ **reset** : un bouton qui réinitialise le contenu du formulaire avec les valeurs par défaut.
- ▶ **search** : un champ texte HTML5 sur une ligne pour des termes de recherche. Les sauts de ligne sont automatiquement retirés.
- ▶ **submit** : un bouton qui envoie le formulaire.
- ▶ **tel** : un contrôle HTML5 pour saisir un numéro de téléphone.

Les différents types de champs <input>

Les types de champs disponibles sont :

- ▶ **text** : un champ texte sur une seule ligne. Les sauts de ligne sont automatiquement retirés. **C'est la valeur par défaut !**
- ▶ **time** : un contrôle HTML5 pour saisir l'heure sans la zone UTC
- ▶ **url** : un champ HTML5 permettant de saisir une URL.
- ▶ **week** : un contrôle HTML5 permettant de saisir une date représentée par un numéro de semaine et une année (sans indication de fuseau horaire).

Attributs communs à l'ensemble des types Input

Attribut	Description
autocomplete	Une chaîne de caractères qui indique le type d'autocomplétion à utiliser.
autofocus	Un attribut booléen qui passe le focus sur le champ lorsque le formulaire est affiché.
disabled	Un attribut booléen qui indique si le champ doit être désactivé.
form	L'identifiant du formulaire (la valeur de l'attribut id de l'élément <form>) auquel le champ est rattaché. Si cet attribut est absent, le champ sera rattaché au formulaire le plus proche qui le contient s'il existe.
list	L'identifiant (valeur de l'attribut id) d'un élément <datalist> qui fournit une liste de suggestions.
name	Le nom du champ qui sera rattaché à la donnée envoyée via le formulaire.

► Example : [01_form.html](#)

Attributs communs à l'ensemble des types Input

Attribut	Description
readonly	Un attribut booléen qui indique si le champ peut être édité ou non.
required	Un attribut booléen qui indique que le champ doit être renseigné avant de pouvoir envoyer le formulaire.
tabindex	Une valeur numérique qui indique à l'agent utilisateur l'ordre selon lequel naviguer au clavier grâce à la touche Tab.
type	Une chaîne de caractère qui indique le type de champ représenté par l'élément <input>.
value	La valeur du champ.

► Exemple : [exemples\form\01_form.html](#)

Boutons dans le formulaire

► Attributs des éléments boutons

Attribut	Description
alt	Pour les boutons image, le texte alternatif si l'image n'est pas chargée
height	Pour les boutons image, la longueur de l'image en pixels
src	Pour les boutons image, l'URL de l'image
width	Pour les boutons image, la largeur de l'image
formaction	L'URL à laquelle envoyer les données du formulaire. Cette valeur prend le pas sur l'attribut action du formulaire s'il est défini.
formenctype	Une chaîne de caractères qui indique le type d'encodage à utiliser pour les données du formulaire.
formmethod	La méthode HTTP à utiliser pour envoyer le formulaire.
formnovalidate	Un booléen qui, lorsqu'il est présent, indique que les champs du formulaire ne sont pas soumis aux contraintes de validation avant l'envoi des données au serveur.

► Exemple : [exemples\form\02_buttons.html](#)

Champ de saisie avec du texte

- ▶ Les éléments <input> dont l'attribut type vaut "text" permettent de créer des champs de saisie avec du texte sur une seule ligne.

Attribut	Description
maxlength	Le nombre de caractères maximal qui peut être écrit dans ce champ.
minlength	Le nombre de caractères minimal qui peut être écrit dans ce champ pour qu'il soit considéré comme valide.
pattern	Une expression rationnelle à laquelle doit correspondre le texte saisi pour être valide.
placeholder	Une valeur d'exemple qui sera affichée lorsqu'aucune valeur n'est saisie.
readonly	Un attribut booléen qui indique si le contenu du champ est en lecture seule.
size	Un nombre qui indique le nombre de caractères affichés par le champ.
spellcheck	Cet attribut contrôle l'activation de la vérification orthographique sur le champ ou si la vérification orthographique par défaut doit être appliquée.

- ▶ Exemple : [exemples\form\03_textfields.html](#)

Boutons radio et Boîtes à cocher

► Boutons radio

- Les éléments `<input>` dont l'attribut `type` vaut **radio** sont généralement utilisés pour construire des groupes d'options parmi lesquelles on ne peut choisir qu'une valeur.
- Les « boutons radio » sont représentés par des cercles remplis lorsqu'ils sont sélectionnés.

► Boîtes à cocher

- Les éléments `<input>` de type **checkbox** sont affichés sous la forme de boîtes à cocher qui sont cochées lorsqu'elles sont activées.
- Elles permettent de sélectionner une ou plusieurs valeurs dans un formulaire.

Attribut	Description
<code>checked</code>	Un attribut booléen. Si celui-ci est présent, la case à cocher sera cochée.

- Exemple : [exemples\form\04_radio_and_checkbox.html](#)

Liste déroulante

- ▶ L'élément HTML <**select**>
 - ▶ représente un contrôle qui fournit une liste d'options parmi lesquelles l'utilisateur pourra choisir
 - ▶ Attribut **multiple**
 - ▶ permet de choisir plusieurs options simultanément
 - ▶ Attribut **size**
 - ▶ indique le nombre d'options affichées en même temps

Liste déroulante

- ▶ L'élément HTML **<option>**
 - ▶ Chaque option est définie grâce à un élément **<option>** qui se situe à l'intérieur de l'élément **<select>**
 - ▶ Chaque élément **<option>** doit avoir un attribut **value** qui contient la valeur qui sera envoyée au serveur lorsque l'option est sélectionnée.
 - ▶ Si aucune valeur n'est fournie, la valeur par défaut sera le texte contenu dans l'élément.
 - ▶ Il est possible d'inclure un attribut **selected** sur un élément **<option>** afin que cette option soit sélectionnée par défaut au chargement de la page.
- ▶ L'élément HTML **<optgroup>**
 - ▶ Il est possible de regrouper plusieurs éléments **<option>** à l'intérieur d'éléments **<optgroup>** afin de créer des groupes d'options distincts.
- ▶ Exemples : [exemples\form\05_lists.html](#), [exemples\form\06_list_box.html](#)

Textarea

- ▶ L'élément HTML **<textarea>** représente un contrôle qui permet d'éditer du texte sur plusieurs lignes.

Attributs :

▶ **rows** et **cols**

- ▶ permettent de définir la taille exacte qui doit être occupée par l'élément **<textarea>**.
- ▶ Les navigateurs pouvant être différents, c'est une bonne idée que d'utiliser ces attributs pour garantir une certaine homogénéité.

▶ **wrap**

- ▶ indique la gestion des retours à la ligne et la façon d'afficher le texte saisi lorsque celui-ci atteint le bord de la zone du **<textarea>**

- ▶ Exemple : [exemples\form\07_textarea.html](#)

Étiquettes

- ▶ L'élément HTML **<label>** représente une légende pour un objet d'une interface utilisateur.
- ▶ Il peut être associé à un contrôle en utilisant l'attribut **for** ou en plaçant l'élément du contrôle à l'intérieur de l'élément **<label>**.
- ▶ Attribut **for**
 - ▶ L'identifiant (la valeur de l'attribut id) de l'élément de formulaire associé, appartenant au même document que l'élément label.
 - ▶ Le premier élément du document dont l'identifiant correspond est alors le contrôle étiqueté par l'élément.
- ▶ Exemple : [exemples\form\08_labels.html](#)

Groupe de contrôles et Légende

- ▶ L'élément HTML **<fieldset>** est utilisé afin de regrouper plusieurs contrôles interactifs ainsi que des étiquettes (**<label>**) dans un formulaire web.
- ▶ L'élément HTML **<legend>** représente une légende pour le contenu de son élément parent **<fieldset>**.
- ▶ Exemple : [exemples\form\09_fieldset.html](#)

Téléversement de fichier

- ▶ Les éléments <input> dont l'attribut type vaut "**file**" permettent à un utilisateur de sélectionner un ou plusieurs fichiers depuis leur appareil et de les téléverser (ou uploader) vers un serveur via un formulaire

Attribut	Description
accept	Un ou plusieurs identifiants de type de fichier décrivant les types de fichier autorisés (ex : .pdf, image/*, audio/*, video/*, ...).
multiple	Un attribut booléen qui, lorsqu'il est présent, indique que plusieurs fichiers peuvent être sélectionnés.

- ▶ Exemple : [exemples\form\10_fileupload.html](#)

Aligner des contrôles

- ▶ On utilise CSS pour aligner correctement les contrôles.
- ▶ Exemple : [exemples\form\11_align_controls.html](#)

Formater les contrôles

- ▶ On utilise CSS pour formater les contrôles
- ▶ Exemple : [exemples\form\12_format_controls.html](#)

Navigation au clavier et Raccourci clavier

► **tabindex**

- Une valeur numérique qui définit si le champ peut recevoir le focus via la navigation au clavier (en navigant avec la touche Tab) et la façon dont l'élément s'inscrit dans l'ordre de la navigation au clavier.

► **accesskey**

- Cet attribut fournit une indication permettant de générer un raccourci clavier pour l'élément courant. Cet attribut se compose d'une liste de caractères séparés par des espaces. Le navigateur doit utiliser le premier caractère qui existe selon la disposition du clavier utilisée.
- Exemple : [exemples\form\13_access_keys.html](#)

Validation de données

- ▶ Une des caractéristiques de HTML5 est la possibilité de valider la plupart des données utilisateur sans avoir recours à des scripts.
- ▶ Ceci se fait en utilisant des attributs de validation sur les éléments de formulaire
- ▶ Ils permettent de spécifier des règles pour une entrée de formulaire comme :
 - ▶ une valeur doit-elle être remplie ?
 - ▶ y a-t-il une longueur minimale et/ou maximale des données ?
 - ▶ doit-elle être un nombre, une adresse e-mail ou autre chose ?
 - ▶ doit-elle correspondre à un modèle ?
- ▶ Si les données saisies suivent toutes ces règles, elles sont considérées comme valides ;
- ▶ si ce n'est pas le cas, elles sont considérées comme non valides.

Validation de données

- ▶ L'attributs de validation
- ▶ **required**
 - ▶ La fonctionnalité de validation HTML5 la plus simple à utiliser est l'attribut required
 - ▶ si vous voulez rendre une entrée obligatoire, vous pouvez marquer l'élément en utilisant cet attribut.
 - ▶ Lorsque cet attribut est mis, le formulaire ne sera pas soumis (et affichera un message d'erreur) si l'entrée est vide (l'entrée sera également considérée comme invalide).
- ▶ **novalidate**
 - ▶ Cet attribut booléen indique si le formulaire doit être validé au moment de sa soumission.
 - ▶ S'il n'est pas défini, le formulaire sera validé lors de sa soumission.
 - ▶ Il peut être surchargé par l'attribut formnovalidate des éléments <button> ou <input> appartenant au formulaire.
- ▶ Exemple : [exemples\form\14_validation.html](#)

Expressions régulières (regex) pour la validation de données

- ▶ L'attribut **pattern** permet d'indiquer une expression rationnelle que devra respecter la valeur saisie afin d'être valide
- ▶ L'attribut **title** pourra être utilisé afin d'afficher une bulle d'informations qui explique les conditions à respecter.
- ▶ Quelques motifs:

Used for	Pattern
Password (6+ alphanumeric)	[a-zA-Z0-9]{6,}
Zip code (99999 or 99999-9999)	\d{5}([-\d{4})?
Phone number (999-999-9999)	\d{3}[-]\d{3}[-]\d{4}
Date (MM/DD/YYYY)	[01]?[0-3]\d/[0-3]\d/\d{4}
URL (starting with http:// or https://)	https?:\/\/.+
Credit card (9999-9999-9999-9999)	^\d{4}-\d{4}-\d{4}-\d{4}\$

Expressions régulières (regex)

► Quelques exemples de base :

- ▶ a — correspond à un caractère qui doit être un a (ni b, ni aa, etc.)
- ▶ abc — correspond à a, suivi de b, suivi de c.
- ▶ a^* — correspond au caractère a, absent ou présent plusieurs fois (+ correspond à un caractère une ou plusieurs fois).
- ▶ $[^a]$ — correspond à un caractère qui n'est pas un a.
- ▶ $a | b$ — correspond à un caractère qui est a ou b.
- ▶ $[abc]$ — correspond à un caractère qui est a, b ou c.
- ▶ $[^abc]$ — correspond à un caractère qui n'est pas a, b ou c.
- ▶ $[a-z]$ — correspond à tout caractère de la plage a-z, en minuscules seulement (utilisez $[A-Za-z]$ pour minuscules et majuscules et $[A-Z]$ pour les majuscules uniquement).

Expressions régulières (regex)

- ▶ **Quelques exemples de base :**
 - ▶ a.c — correspond à a, suivi par n'importe quel caractère, suivi par c.
 - ▶ a{5} — correspond à a, 5 fois.
 - ▶ a{5,7} — correspond à a, 5 à 7 fois, mais ni plus, ni moins.
- ▶ Vous pouvez utiliser des nombres ou d'autres caractères dans ces expressions, comme :
 - ▶ [-] — correspond à une espace ou un tiret.
 - ▶ [0-9] — correspond à tout nombre compris entre 0 et 9.
- ▶ Exemple : [exemples\form\15_regex.html](#)

Contenus auto-complétés (Datalist)

- ▶ L'élément HTML <**datalist**> contient un ensemble d'éléments <option> qui représentent les valeurs possibles pour d'autres contrôles.
- ▶ Une fois la liste de données affiliée au widget de formulaire, ses options s'utilisent comme complémentation du texte saisi par l'utilisateur
- ▶ Exemple : [exemples\form\16_datalist.html](#)

Email, URL et Tel

► Email

- ▶ Les éléments `<input>` dont l'attribut type vaut "**email**" permettent à un utilisateur de saisir et d'éditer une adresse mail ou,
- ▶ si l'attribut multiple est indiqué, une liste d'adresses mail.
- ▶ La valeur saisie est automatiquement validée afin de vérifier que le champ est vide ou que l'adresse (ou la liste d'adresses) est correcte.

► URL

- ▶ Les éléments `<input>` dont l'attribut type vaut "**url**" sont employées afin de permettre à un utilisateur de saisir ou d'éditer une URL.

► Tel

- ▶ Les éléments `<input>` dont l'attribut type vaut "**tel**" permettent à un utilisateur de saisir un numéro de téléphone.
- ▶ Contrairement aux contrôles utilisés pour `<input type="email">` et `<input type="url">`, la valeur saisie n'est pas automatiquement validée selon un format donné car les formats des numéros de téléphone varient à travers le monde.

▶ Exemple : [exemples\form\17_email_url_tel_controls.html](#)

Number et Range

▶ Number

- ▶ Les éléments <input> dont l'attribut type vaut **number** permettent à un utilisateur de saisir des nombres dans un formulaires.
- ▶ De tels contrôles incluent des mécanismes de validation natifs afin de rejeter les valeurs non-numériques.
- ▶ Le navigateur peut agrémenter le contrôle avec des flèches afin d'incrémenter/décrémenter la valeur grâce à la souris ou avec le doigt.

▶ Range

- ▶ Les éléments <input> dont l'attribut type vaut **range** permettent à l'utilisateur d'indiquer une valeur numérique comprise entre deux bornes.

Attribut	Description
max	La valeur maximale autorisée.
min	La valeur minimale autorisée.
step	Le pas utilisé pour incrémenter la valeur du champ. Cette valeur est utilisée pour l'interface utilisateur du contrôle et pour la validation de la valeur.

▶ Exemple : [exemples\form\18_number_range_controls.html](#)

Date et Time

► Date

- ▶ Les éléments `<input>` dont l'attribut type vaut **date** permettent de créer des champs permettant de saisir des dates (composées d'une année, d'un mois et d'un jour)

► Month

- ▶ Les éléments `<input>` dont l'attribut type vaut "**month**" permettent de créer des contrôles où l'utilisateur peut saisir un mois et année.
- ▶ La valeur associée à un tel élément suit le format "YYYY-MM", où YYYY représente l'année sur quatre chiffre et MM le mois sur deux chiffres.

► Week

- ▶ Les éléments `<input>` dont l'attribut type vaut **week** permettent de créer des champs de saisie où l'on peut saisir une année et le numéro de la semaine pendant cette année (allant de 1 à 52 ou 53, suivant la norme ISO 8601).

Date et Time

► Time

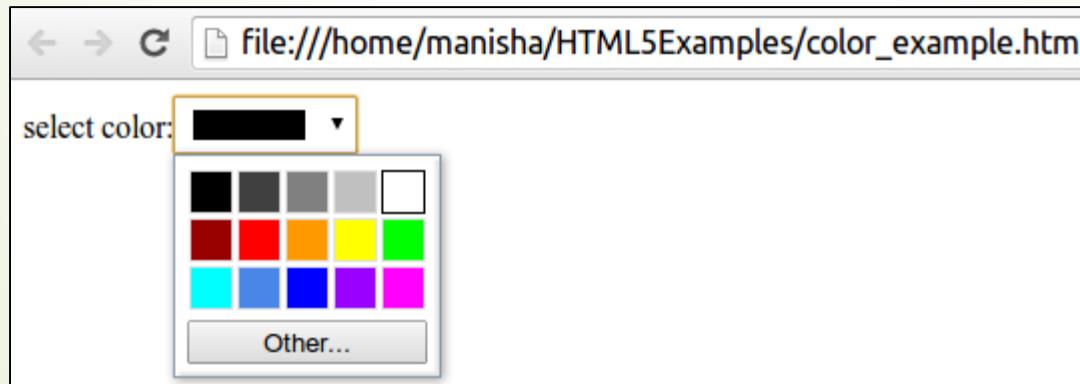
- Les éléments `<input>` dont l'attribut type vaut **time** permettent de créer des contrôles où l'utilisateur peut saisir une heure (avec des minutes et éventuellement des secondes).
- Exemple : [exemples\form\19_dates.html](#)

Champ de recherche

- ▶ Les éléments <**input**> dont l'attribut type vaut **search** permettent à un utilisateur de saisir des termes de recherche.
- ▶ Example : [exemples\form\20_search.html](#)

Couleur

- ▶ Les éléments <**input**> de type "color" permettent de sélectionner une couleur via
 - ▶ une interface (un sélecteur de couleur) ou
 - ▶ en saisissant le code hexadécimal de la couleur au format "#rrggbb".



- ▶ Example : [exemples\form\21_color.html](#)

Exemple complet

► Exemple : [tickets.html](#)



Animations CSS3

Transitions

Transitions CSS3

- ▶ Transitions sont des moyens simples pour créer des effets visuels entre deux états avec CSS
 - ▶ Ex: un élément qui passe un état à l'autre quand la souris est pointé sur cet élément.
- ▶ La propriété **transition** permet de définir la transition entre deux états d'un élément.
- ▶ Les différents états peuvent être définis à l'aide de pseudo-classes telles que :hover ou :active ou être définis dynamiquement avec JavaScript.
- ▶ Voici la liste des propriétés CSS (transition):
 - ▶ <http://www.w3.org/TR/css3-transitions/>
 - ▶ https://www.w3schools.com/cssref/css_animatable.asp

Les propriétés de transition

► **transition-property**

- désigne les noms des propriétés CSS sur lesquelles un effet de transition devrait être appliqué.
- Les valeurs possibles sont : all, none, background-color, text-shadow,...

► **transition-duration**

- définit le nombre de secondes ou de millisecondes que doit durer une animation.
- La valeur par défaut, 0s, indique qu'il n'y aura aucune animation.

► **transition-timing-function**

- décrit la façon dont les valeurs intermédiaires des propriétés CSS affectées par un effet de transition sont calculées.
- Ceci permet donc de définir une courbe d'accélération, de manière à ce que la vitesse de la transition varie au cours de sa durée.

► **transition-delay**

- indique la durée à attendre avant de débuter la transition qui s'applique pour un changement de propriété.

Syntaxe

```
.style {  
    transition-property: all;  
    transition-duration: 1s;  
    transition-timing-function: ease;  
    transition-delay: 0s;  
  
    /*syntax raccourcie*/  
    transition: all 1s ease 0s;  
}
```

Définir plusieurs durées

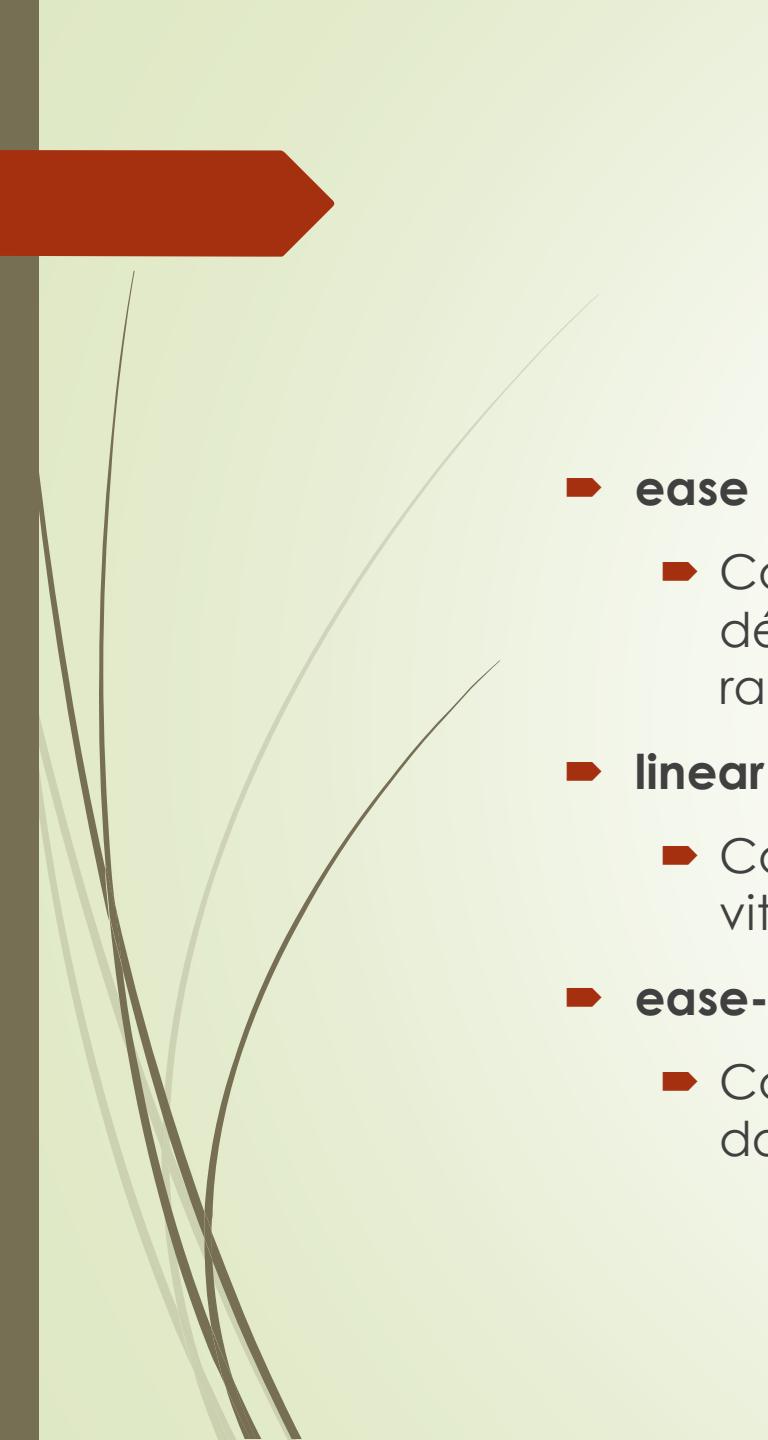
- ▶ Il est possible de définir plusieurs durées, chacune sera appliquée à la propriété correspondante listée par transition-property (qui agit comme un index des propriétés impactées par les animations).
- ▶ S'il y a moins de durées que d'éléments dans cette liste, l'agent utilisateur dupliquera les durées.
- ▶ S'il y a trop de durées, la liste sera tronquée. Dans les deux cas, la déclaration CSS sera considérée comme valide.

```
.style {  
    transition-property: border, color, text-shadow;  
    transition-duration: 2s, 3s, 8s;  
}
```



Fonctions temporelles (Timing functions)

- ▶ Une fonction temporelle définit une courbe d'accélération, de manière à ce que la vitesse de la transition varie au cours de sa durée
- ▶ Les valeurs avec des mots-clés (ease, linear, ease-in-out, etc.) correspondent à une **courbe de Bézier cubique** fixe avec quatre valeurs prédéfinies;
- ▶ La fonction **cubic-bezier()** permet de paramétriser une courbe spécifique.
- ▶ Les fonctions **en escalier** permettent de diviser la transition en intervalles de même durée.
- ▶ **La fonction par défaut : ease**
- ▶ Pour comprendre et visualiser le fonctionnement des fonctions temporelles
 - ▶ Visitez : <https://easings.net/> et <http://cubic-bezier.com/>



Fonctions temporelles (Timing functions)

- ▶ **ease**

- ▶ Correspond à cubic-bezier(0.25, 0.1, 0.25, 1.0) : c'est la valeur par défaut, la vitesse de la transition augmente au milieu de celle-ci puis ralentit à la fin.

- ▶ **linear**

- ▶ Correspond à cubic-bezier(0.0, 0.0, 1.0, 1.0) : la transition s'effectue à vitesse constante.

- ▶ **ease-in**

- ▶ Correspond à cubic-bezier(0.42, 0, 1.0, 1.0) : la transition commence doucement puis la vitesse augmente jusqu'à ce qu'elle soit terminée.



Fonctions temporelles (Timing functions)

- ▶ **ease-out**
 - ▶ Correspond à cubic-bezier(0, 0, 0.58, 1.0) : la transition commence rapidement puis ralentit jusqu'à la fin.
- ▶ **ease-in-out**
 - ▶ Correspond à cubic-bezier(0.42, 0, 0.58, 1.0) : la transition commence lentement, accélère puis ralentit à nouveau avant la fin.
- ▶ **cubic-bezier(p1, p2, p3, p4)**
 - ▶ Une courbe de Bézier paramétrable à l'aide de quatre coefficients compris entre 0 et 1.

Fonctions temporelles (Timing functions)

► **steps(n, <jumpterm>)**

- La transition s'effectue selon n étapes de durées égales. Ainsi, si n vaut 5, la transition se composera de cinq paliers.
- Selon la valeur du paramètre jumpterm, ces paliers se trouveront entre 0%, 20%, 40%, 60% et 80%, ou entre 20%, 40%, 60%, 80% et 100%, ou incluront également 0% et 100% (soit 0%, 25%, 50%, 75% et 100%) :
- jump-start | start
 - La fonction est continue à gauche et le premier saut se produit en début de transition.
- jump-end | end
 - La fonction est continue à droite et le dernier saut se produit à la fin de la transition.
- jump-none
 - Il n'y a aucune rupture au début ou à la fin. Il y a un palier constant après 0% et un palier constant avant 100% (chacun durant 1/n).
- jump-both
 - Une pause est présente aux niveaux 0% et 100%, ce qui ajoute un niveau pendant la transition.



Fonctions temporelles (Timing functions)

- ▶ **step-start**
 - ▶ Synonyme de steps(1, jump-start)
- ▶ **step-end**
 - ▶ Synonyme de steps(1, jump-end)
- ▶ Exemple : [exemple01](#)



Animations CSS3

Transformations 2D et 3D

Transformations CSS3

- ▶ En modifiant l'espace des coordonnées, les transformations CSS permettent de changer la position d'un contenu affecté sans perturber le flux normal.
- ▶ Elles sont implémentées en utilisant un ensemble de propriétés CSS qui vous permettent d'appliquer des transformations affines aux éléments HTML.
- ▶ Ces transformations incluent la rotation, l'inclinaison et la translation à la fois dans le plan (2D) ou dans un espace (3D).

Transformations 2D

► **scale**

- ▶ modifie la taille d'un élément.
- ▶ Elle permet d'appliquer une homothétie définie par un vecteur.
- ▶ Ce vecteur décrit l'amplification ou la réduction de l'élément selon chacun des deux axes plans.
- ▶ Si les deux coordonnées sont égales, l'homothétie sera uniforme et la forme de l'élément sera conservée.

► **translate**

- ▶ permet de déplacer un élément sur le plan formé par le document.
- ▶ Cette transformation est caractérisée par un vecteur contenant deux coordonnées dont chacune définit la quantité de déplacement horizontale et verticale.

Transformations 2D

► **rotate**

- ▶ définit une transformation qui déplace l'élément autour d'un point fixe (défini par la propriété transform-origin) sans le déformer.
- ▶ La quantité de mouvement est définie par l'angle passé en argument.
- ▶ Si la valeur fournie est positive, le mouvement se fait dans le sens des aiguilles d'une montre (sens horaire) et sinon dans le sens inverse (sens anti-horaire).

► **skew**

- ▶ permet de décrire une distorsion de l'élément où chaque point est tourné d'un angle dans chaque direction.
- ▶ Plus le point est éloigné de l'origine, plus grande sera la transformation appliquée.

► **matrix**

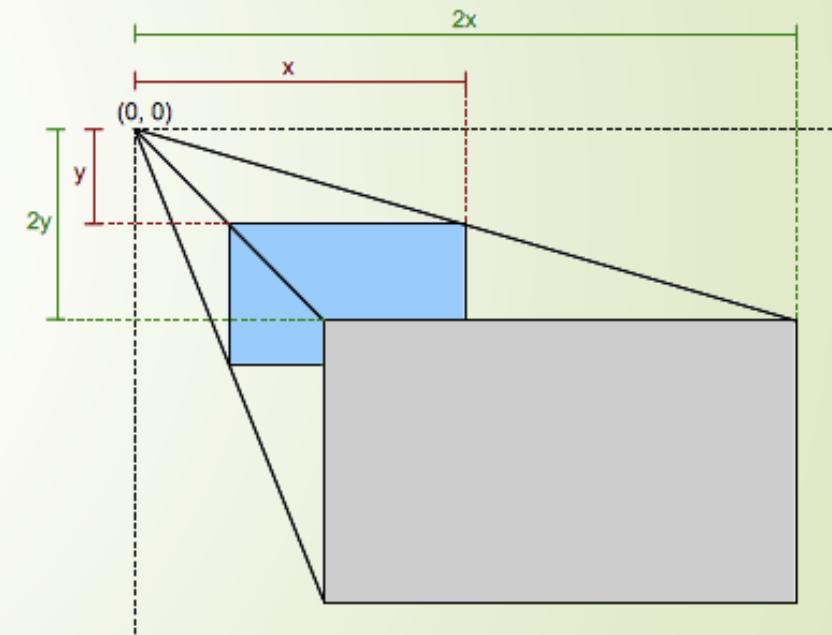
- ▶ permet de définir une matrice de transformation homogène en deux dimensions.
- ▶ Cette matrice contient six coefficients.
- ▶ Les valeurs constantes de ces matrices sont implicites et ne font donc pas partie des paramètres.
- ▶ Les autres paramètres sont décrits dans l'ordre des colonnes.

Scale

- ▶ Modifie la taille d'un élément
- ▶ Syntaxe

```
.scale:hover {  
    /* agrandit l'élément à 2 fois*/  
    /* identique à scaleX(2) scaleY(2) */  
    transform: scale(2);  
  
    /* diminue l'élément à moitié  
    transform: scale(0.5); */  
}
```

Exemple : [exemple02](#)

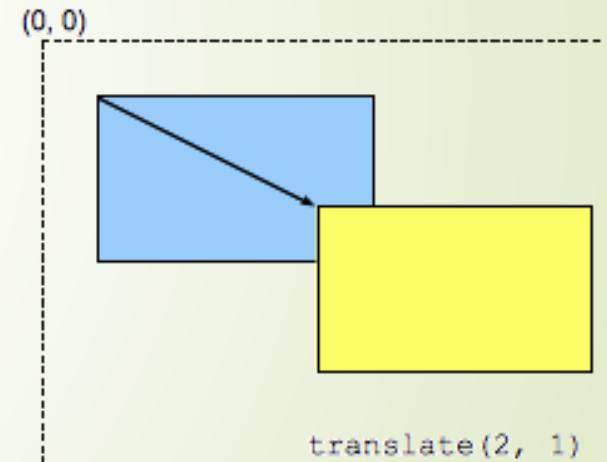


`scale(2, 2)`

Translate

- ▶ Déplace un élément sur le plan représenté par le document
- ▶ Syntaxe

```
.translate:hover {  
    transform: translate(-20px, -20px);  
    /*  
        transform: translateX(-20px);  
        transform: translateY(-20px);  
    */  
}
```



Exemple : [exemple02](#)

Centrer un élément avec position absolue : [exemple03](#)

Rotate

- ▶ Définit une transformation qui déplace un élément autour d'un point fixe
- ▶ Syntaxe

```
.rotate:hover {  
    transform: rotate(30deg);  
}
```

Exemple : [exemple02](#)

Skew

- ▶ opérer une distorsion en étirant chaque point de l'élément d'un certain angle dans une direction du plan
- ▶ Syntaxe

```
.skew:hover {  
    transform: skew(40deg, 12deg);  
}
```

Example : [exemple02](#)

Matrix

- ▶ définit une matrice homogène de transformation, définie par 6 coefficients
- ▶ Syntaxe

```
.matrix:hover {  
    transform: matrix(1.678, -0.256, 1.522, 2.333, -51.533, -1.989);  
}
```

Pour mieux comprendre la syntaxe, visiter ce site web : [matrix](#)

Pour non-mathématicien, visiter ce site web : <http://www.useragentman.com/matrix/>

Exemple : [exemple02](#)

Propriété transform-origin

- ▶ Spécifie la position de l'origine.
- ▶ Par défaut, celle-ci est au centre de l'élément et peut être déplacée.
- ▶ Elle est utilisée par de nombreuses transformations, comme les rotations, les homothéties ou l'inclinaison, qui nécessitent un point spécifique pour paramètre.
- ▶ Syntaxe

```
.matrix:hover {  
    transform: matrix(1.678, -0.256, 1.522, 2.333, -51.533, -1.989);  
    transform-origin: 70% 10%;  
}
```

Exemple : [exemple02](#)

Transformations 3D

- ▶ CSS permet d'appliquer des transformations affines aux éléments HTML dans un espace 3D
- ▶ Réaliser des transformations CSS dans l'espace est un petit peu plus complexe.
- ▶ Il faut d'abord définir l'espace 3D en lui donnant une perspective
- ▶ Puis il faut configurer le comportement des éléments 2D dans cet espace.

Définir une perspective

- ▶ Le premier élément à définir est la perspective.
- ▶ La perspective est ce qui donne l'impression de troisième dimension.
- ▶ Plus les éléments sont loin de l'observateur, plus ils seront petits.
- ▶ Exemple : [exemple04](#)



Animations CSS3

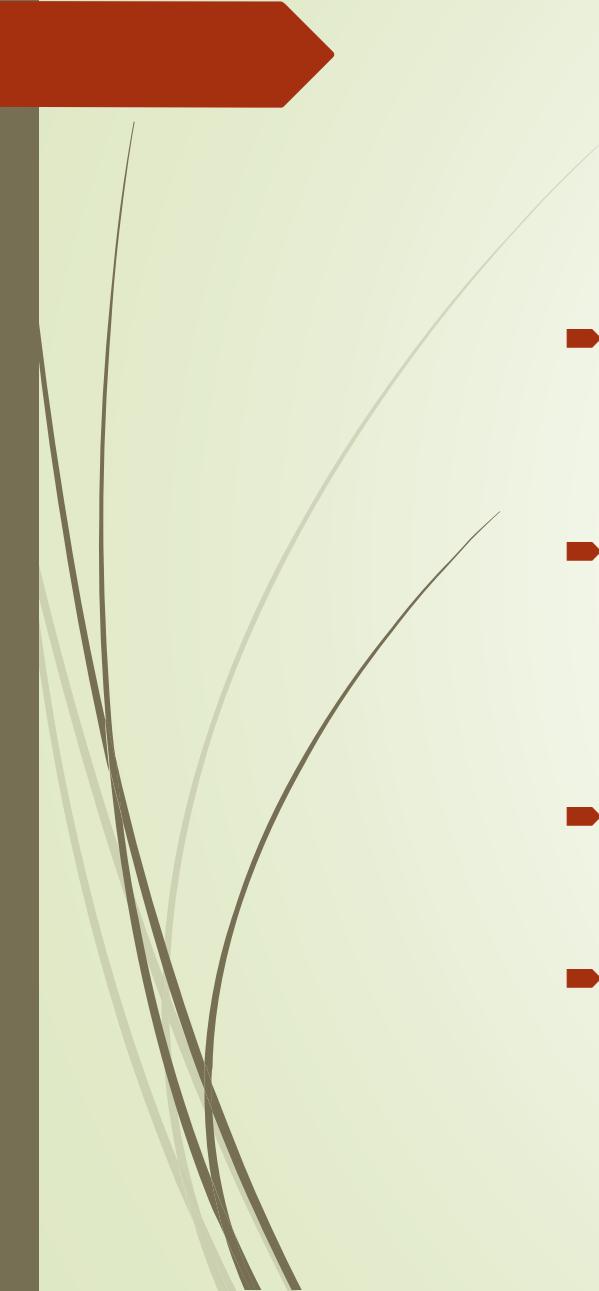
Animation avec keyframes

Animation CSS3

- ▶ Les animations CSS permettent de créer des transitions entre deux états de mise en forme.
- ▶ Une animation est décrite par deux choses
 - ▶ des propriétés propres à l'animation d'une part et
 - ▶ un ensemble d'étapes (keyframes) qui indiquent l'état initial, final et éventuellement des états intermédiaires d'autre part.
- ▶ Supporté par la majorité des navigateurs
- ▶ Exemple : [exemple05](#)

Propriété animation

- ▶ Pour créer une animation CSS, il faut utiliser la propriété raccourcie **animation** ou les propriétés détaillées correspondantes sur un ou plusieurs éléments.
- ▶ Cette propriété permet de configurer
 - ▶ la durée,
 - ▶ le minutage et
 - ▶ d'autres détails à propos de l'animation.
 - ▶ Attention, cela ne détermine pas l'apparence visuelle de l'animation.



Les propriétés détaillées rattachées à la propriété raccourcie animation

- ▶ **animation-delay**
 - ▶ Cette propriété définit le délai entre le moment où l'élément est chargé et le moment où l'animation commence.
- ▶ **animation-direction**
 - ▶ Cette propriété indique si l'animation doit alterner entre deux directions de progressions (faire des allers-retours) ou recommencer au début à chaque cycle de répétition.
- ▶ **animation-duration**
 - ▶ Cette propriété définit la durée d'un cycle de l'animation.
- ▶ **animation-fill-mode**
 - ▶ Cette propriété indique les valeurs qui doivent être appliquées aux propriétés avant et après l'exécution de l'animation.



Les propriétés détaillées rattachées à la propriété raccourcie animation

► **animation-iteration-count**

- Cette propriété détermine le nombre de fois que l'animation est répétée. On peut utiliser le mot-clé infinite afin de répéter une animation infiniment.

► **animation-name**

- Cette propriété permet de déclarer un nom qui pourra être utilisé comme référence à l'animation pour la règle @keyframes.

► **animation-play-state**

- Cette propriété permet d'interrompre (« pause ») ou de reprendre l'exécution d'une animation.

► **animation-timing-function**

- Cette propriété configure la fonction de minutage d'une animation, autrement dit comment celle-ci accélère entre l'état initial et l'état final notamment grâce à des fonctions décrivant des courbes d'accélération.



Animation Properties



```
.animation-properties {  
    animation-name: warning;  
    animation-duration: 1.5s;  
    animation-timing-function: ease-in-out;  
    animation-iteration-count: infinite;  
    animation-play-state: running | paused;  
    animation-delay: 0s;  
    animation-fill-mode: none | forwards | backwards | both;  
    animation-direction: normal | reverse | alternate | alternate-reverse;  
}
```

@keyframes

- ▶ Une fois qu'on a définit les propriétés propres à l'animation, on doit définir la mise en forme qui évolue lors de cette animation.
 - ▶ Pour cela on définit deux étapes ou plus grâce à la règle @keyframes.
 - ▶ Chaque étape décrit la façon dont l'élément animé doit être affiché à un instant donné lors de l'animation.
- ▶ La durée de l'animation est définie avant et la règle @keyframes utilise donc des valeurs exprimées en pourcentages (type CSS percentage) pour indiquer l'instant correspondant à cet état.
 - ▶ **0%** indique l'état initial de l'animation et **100%** indique l'état final.
 - ▶ Ces deux états étant très important, il existe 2 alias pour les décrire : **from** et **to**.
 - ▶ Ces états sont optionnels et si from/0% ou to/100% ne sont pas définis, le navigateur utilisera les valeurs calculées des différentes propriétés.
 - ▶ Il est également possible d'ajouter des étapes intermédiaires, entre l'état initial et l'état final de l'animation.
- ▶ Exemple : [exemple06](#)
- ▶ Veille sur l'animations CSS3 : <http://dev.w3.org/csswg/css3-animations/>



Boîtes flexibles CSS

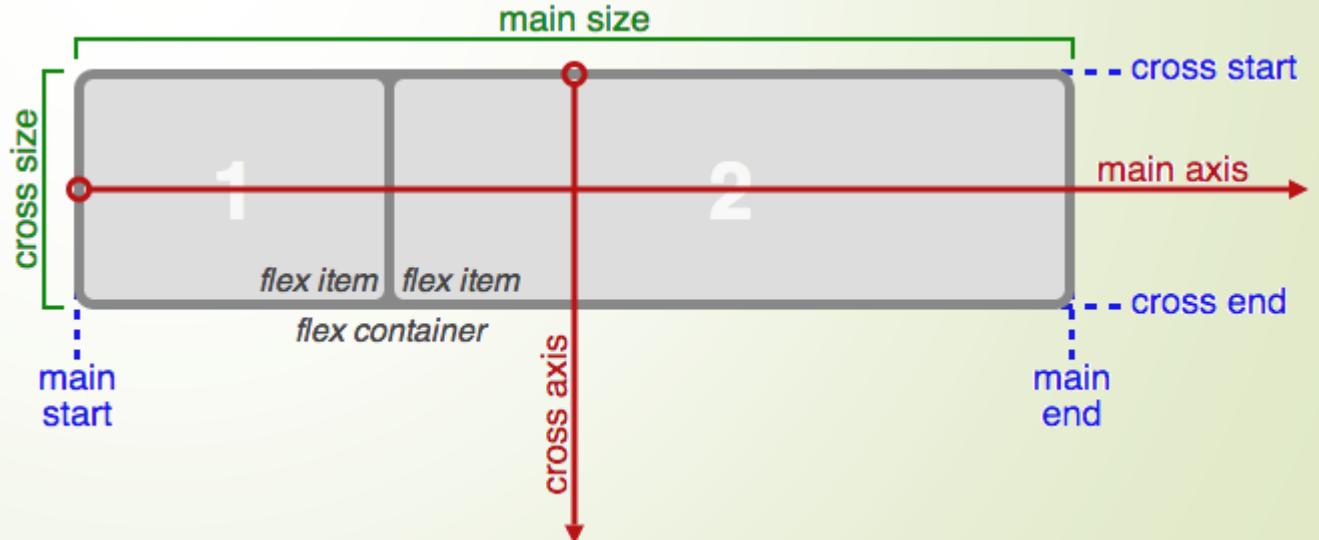
CSS Flexible Box Layout

Boîtes flexibles CSS

- ▶ Le module de disposition des boîtes flexibles CSS (CSS Flexible Box Layout)
 - ▶ Est un module de CSS qui définit un modèle de boîtes optimisé pour la conception des interfaces utilisateurs.
 - ▶ Groupes « flexibles »
 - ▶ Dans tous les navigateurs récents : <http://caniuse.com>
 - ▶ Permet de faire facilement, par exemple, des éléments de même hauteur
 - ▶ Doc et démos sympathiques : css-tricks.com

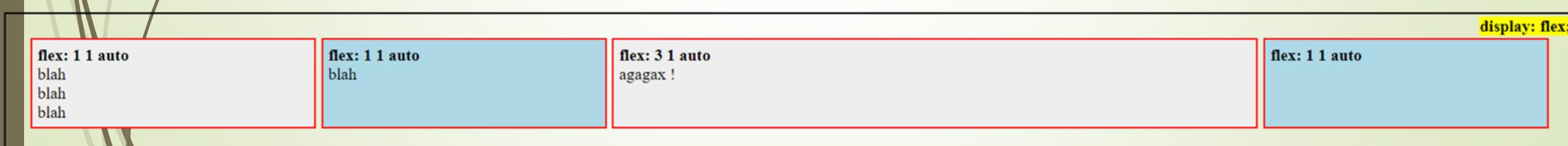
Principe

- ▶ On marque la boîte parente en **display: flex**
- ▶ Les boîtes filles sont organisée le long d'un axe principal (horizontal ou vertical)
- ▶ Dans l'autre dimension, elles auront par défaut la même taille : pour un axe horizontal, on aura des boîtes disposées côte à côte et de même hauteur
- ▶ on peut demander à répartir sur plusieurs lignes (ou colonnes)
- ▶ nombreux réglages possibles
(par exemple : alignement)



Exemple

- ▶ Cases de même hauteur sur une seule ligne
- ▶ L'élément parent a la propriété **display: flex**
- ▶ par défaut, il est organisé en lignes
- ▶ il fait par défaut une seule ligne
- ▶ Ses enfants se partageront l'espace selon la valeur de leur propriété **flex**
- ▶ dans l'autre direction (ici la verticale), ils auront par défaut la même hauteur



Autre exemple

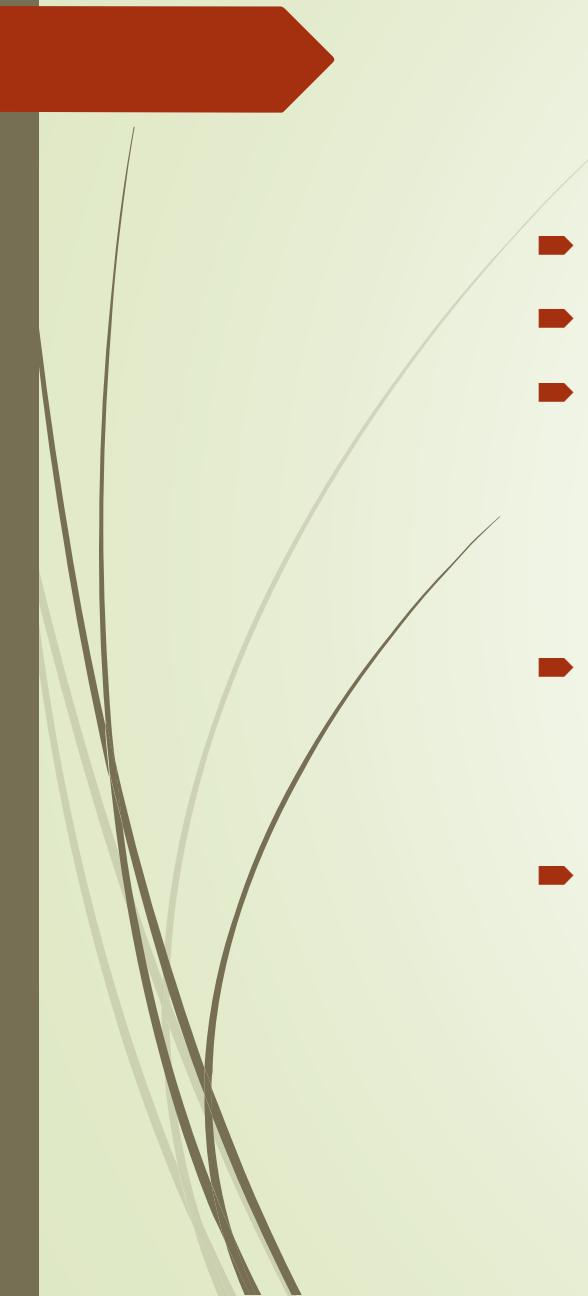
- ▶ Ici, la valeur de **flex** n'est pas précisée pour les **enfants** ;
 - ▶ elle équivaut à **flex: 0 1 auto;**
 - ▶ On voit ici que les éléments se sont placés les uns à côté des autres, en prenant chacun la place dont il a besoin.
 - ▶ Ils ont tous la même hauteur.

Premier élément, Premier élément.
Second paragraphe

Second élément

Troisième élément

display: flex;



La propriété flex

- ▶ Elle est définie sur l'élément enfant
- ▶ 3 paramètres : flex-grow, flex-shrink et flex-basis
- ▶ **grow**
 - ▶ capacité de l'élément à utiliser de l'espace libre.
 - ▶ 0: l'élément ne grossit pas.
 - ▶ Un élément avec grow=2 grossit deux fois plus qu'un élément à 1.
- ▶ **shrink**
 - ▶ capacité de l'élément à rétrécir
 - ▶ fonctionne comme grow.
- ▶ **basis**
 - ▶ fixe la taille de départ (à partir de laquelle on essaie de grossir ou de rétrécir).
 - ▶ Peut être une dimension (comme 3cm ou 2em),
 - ▶ content, qui utilise le contenu de l'élément lui-même pour calculer sa taille, ou
 - ▶ auto, qui réparti l'espace total disponible dans cette alignment (c'est à dire celui laissé par tous les éléments).

Exemple 1

- ▶ Ici, on a trois éléments en **flex : 1 1 auto**
- ▶ On remarque que:
 - ▶ à cause de **auto**, les éléments ont une taille de base qui dépend de la largeur « naturelle » de leur contenu

flex: 1 1 auto;

Premier élément, Premier élément, Premier élément.

flex: 1 1 auto;

Second élément

display: flex;

flex: 1 1 auto;

Troisième élément

Exemple 2

flex: 1 1 auto;

Premier élément. Premier élément. Premier élément. Premier élément. Premier élément. Premier élément.

aaaaaaa a a a a a

flex: 1 1 auto;

flex: 1 1 auto;

bla bla bla

flex: 1 1 auto;

aaaaaaa

```
flex: 1 1  
auto;  
du texte
```

```
flex: 1 1  
auto;  
bla bla bla
```

flex: 1 1 auto;

a a a a a
a a a a a

```
flex: 1 ;  
auto;  
du texte
```

```
flex: 1 1  
auto;  
bla bla  
bla
```

flex: 1 1 auto;

a a
a a
a a
a a

flex
1 1
auto
du
texte

```
flex:  
1 1  
auto;  
bla  
bla  
bla
```



Auto et élément large

- ▶ Ici, nous sommes toujours sur des **flex à 1 1 auto**, mais le premier élément est trop long pour que tout tienne sur une ligne.
- ▶ Flex doit réduire les éléments (en utilisant la propriété shrink qui est à 1). L'espace assigné par défaut dépend de la largeur « naturelle » de l'élément, ainsi que de sa capacité à être réduit.
- ▶ Observez le premier et le second élément en particulier.

Basis à auto

- ▶ avec une **basis** à **auto**, les éléments se partagent l'espace restant disponible : on retire la taille naturelle des éléments de l'espace disponible, et on partage équitablement (en fonction de **grow**) l'espace qui reste.

display: flex;

flex: 1 1 auto; Premier élément. Premier.

flex: 1 1 auto; 2

flex: 1 1 auto; 3

Basis à 0em

- ▶ avec une **basis** de **0em**, les éléments se partagent équitablement l'espace **total** (les colonnes ont alors toutes la même largeur)

flex: 1 1 0em; Premier élément, Premier élément, Premier élément,
Premier élément

flex: 1 1 0em; Second élément

flex: 1 1 0em; Troisième élément

display: flex;

- ▶ Comparez avec :

flex: 1 1 auto; Premier élément, Premier élément, Premier élément, Premier élément

flex: 1 1 auto; Second élément

flex: 1 1 auto; Troisième élément

display: flex;

Basis fixée à un pourcentage

- ▶ On peut aussi décrire basis avec un pourcentage
- ▶ Ici, le troisième élément a une basis de 80%
- ▶ Du coup, il ne reste plus beaucoup d'espace pour les autres

`flex: 1 1
0em;Premier
élément,
Premier
élément,
Premier
élément,
Premier
élément`

`flex: 1 1
0em;Second
élément`

`flex: 1 1 80%;Troisième élément`

`display: flex;`

Basis fixée à un pourcentage

- ▶ En ajoutant la possibilité de placer les éléments en plusieurs couches (avec wrap), on pourra utiliser une largeur de 100% pour occuper tout une ligne :

`flex: 1 1 0em;Premier élément, Premier élément, Premier élément, Premier élément`

`flex: 1 1 0em;Second élément`

`display: flex; flex-flow: row wrap;`

`flex: 1 1 100%;Troisième élément`

...
...
...
...
...
...
...

Grow

- ▶ **grow** permet de fixer comment l'espace disponible sera réparti
- ▶ Un élément avec un grow de 4 récupérera 4 fois plus d'espace disponible qu'un élément avec un grow de 1
- ▶ Un élément avec un grow de 0 ne sera pas étendu
- ▶ Ici, avec une **basis** de 0em, si l'élément central a un coefficient grow de 4, il prend 4 fois plus de place qu'un élément avec un grow à 1

display: flex;

flex: 1 1 0em; Premier élément,
Premier élément, Premier élément,
Premier élément

flex: 4 1 0em; Second élément

flex: 1 1 0em; Troisième élément

Grow

- ▶ Même chose, mais avec **basis=auto**
 - ▶ s'il n'y a pas d'espace à répartir, c'est l'élément avec le plus de contenu qui sera le plus large.
 - ▶ S'il y a de l'espace, il sera partagé selon la valeur de grow

`flex: 1 1 auto; Premier élément, Premier élément, Premier élément, Premier élément`

`flex: 4 1 auto; Second élément`

`flex: 1 1 auto; Troisième élément`

`display: flex;`

Shrink à 0

- ▶ Si on place le **shrink** du premier élément à 0, il n'est plus capable de diminuer, et essaie d'être sur une seule ligne

display: flex;

display: flex;

`flex: 1 1 auto;` Second element

`flex: 1 1 auto;` Troisième élément

Utilisation de flex-flow

- ▶ Cette propriété se place sur le parent et permet :
 - ▶ des mises en page en ligne ou en colonne (row ou column)
 - ▶ d'activer ou de désactiver la possibilité de mettre en page sur plusieurs couches (wrap ou nowrap)
 - ▶ **par défaut : row nowrap**
 - ▶ est un raccourci pour les deux propriétés flex-direction et flex-wrap.

Utilisation de flex-flow - Exemple

- ici, les éléments sont tous en **flex 1 1 auto**, et **flex-flow** est **row wrap**
 - Le premier élément prend sa taille naturelle, qui est de la largeur de la page, et repousse les deux autres sur la ligne suivante
 - Dans le cas présent, les éléments ont les mêmes caractéristiques : c'est juste la taille du contenu de la première case qui déclenche la mise en page sur plusieurs rangées

display: flex; flex-flow: row wrap;

flex: 1 1 auto; Second élément

flex: 1 1 auto; Troisième élément

Utilisation de flex-flow - Exemple

- ▶ En enlevant du contenu au premier élément, on obtient :

flex: 1 1 auto;Premier élément, Premier élément

flex: 1 1 auto;Second élément

flex: 1 1 auto;Troisième élément

display: flex; flex-flow: row wrap;

Quelques essais en jouant sur les paramètres

- En fixant une basis différente d'auto ou zéro, on se donne une taille de départ en dessous de laquelle on aura recours à un saut de ligne

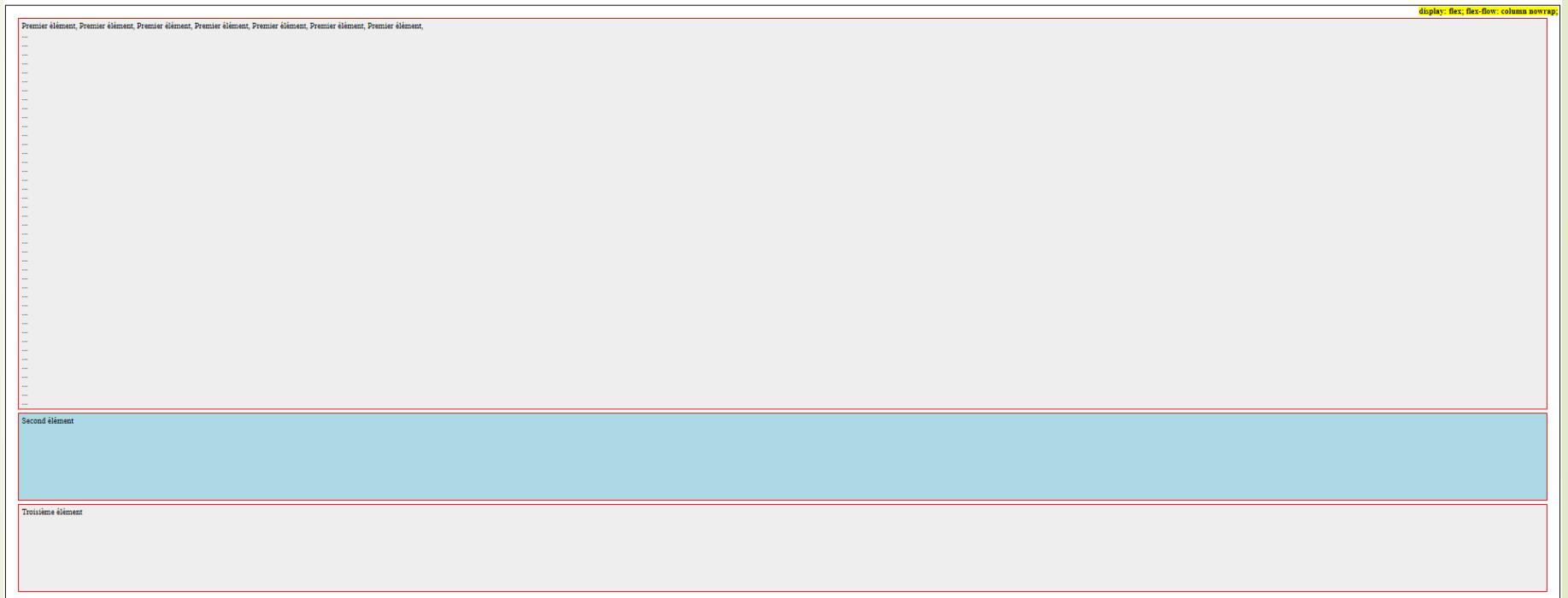
flex: 1 1 10em; Second élément

flex: 2 1 0em; Second élément

flex: 2 1 0em; Troisième élément Troisième élément
Troisième élément Troisième élément Troisième
élément Troisième élément Troisième élément
Troisième élément Troisième élément Troisième
élément Troisième élément Troisième élément
Troisième élément Troisième élément Troisième
élément

Usage de flex-direction: column

- ▶ Dans le cas des colonnes, les éléments de la même colonne ont normalement une même largeur, et remplissent la colonne
- ▶ Contrairement à la largeur d'une ligne, la hauteur d'une colonne n'est pas bornées par les limites de l'écran



- ▶ L'utilisation de column wrap est beaucoup plus compliquée, et ne fonctionne bien que si les éléments ne sont pas trop larges

Utilisation pour une mise en page complexe

- ▶ Flow permet de créer très facilement des mises en pages complexes et adaptatives (« responsive »)
- ▶ Dans l'exemple qui suit, en-tête et pied de page prennent toute la largeur de l'écran ; le reste de la page est sur trois colonnes, sauf s'il y a moins de 30em d'espace disponible, auquel cas, les éléments se placent les uns en dessous des autres

```
display: flex; flex-flow: row wrap;
```

flex 1 1 100%; En-tête		
flex 0 0 10em; Côté gauche.	flex 1 1 10em; Contenu de la page.	flex 0 0 10em; Troisième élément
flex 1 1 100%; Pied de page		

Autres propriétés

- ▶ **order**
 - ▶ Permet de modifier l'ordre des enfants d'un élément flex.
 - ▶ intéressant en combinaison avec les media-queries pour présenter les éléments dans des ordres différents selon la taille des supports
- ▶ **justify-content, align-items, align-content, align-self**
 - ▶ Permettent de régler la disposition des éléments les uns par rapport aux autres et leur espacement dans le conteneur.
- ▶ **inline-flex**
 - ▶ Un élément en display flex fonctionne extérieurement comme un bloc. Si on veut le placer à côté d'un autre élément, il faut utiliser inline-flex.



Le référencement

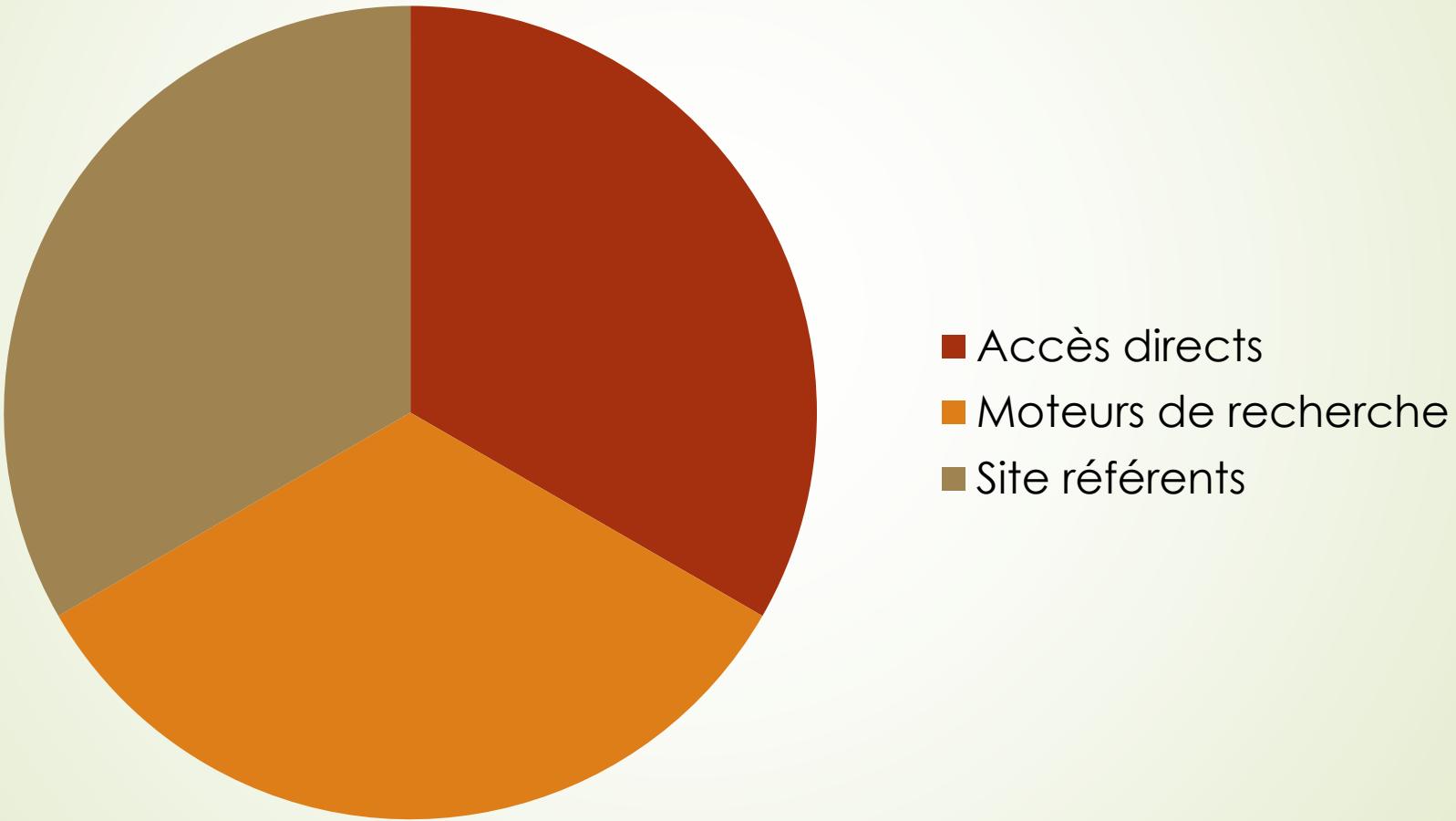
Source : Nassim BAHRI (Ecole Supérieure d'Economie Numérique)



Pourquoi ?

- ▶ Un site est créé pour être consulté
- ▶ Faire connaître son entreprise, ses services, son domaines d'activité, ses produits, etc..
- ▶ Aider les internautes à trouver votre site web
- ▶ Un site web qui ne génère pas de trafic = 0

Les trois piliers du trafic



Les trois piliers du trafic

1/Les accès directs

- ▶ Ceux qui mettent en marque-page votre site et y reviennent directement.
- ➔ Votre site fourni une réponse à leur besoin

Que faut-il analyser?

Qui revient, selon quelle fréquence, quelles pages,...

But : connaître les clients fidèles

Les trois piliers du trafic

2/Les moteurs de recherche

- ▶ Référencement naturel ou payant

- ➔ Votre site est bien référencé

Que faut-il analyser?

Les mots clés, taux de rebond, première visite?...

Apport réel à l'objectif du site (cliquer sur les pubs, vendre,...)

Les trois piliers du trafic

3/Les sites référents (Netlinking)

- ▶ Ces sites vous recommandent.
Ils vous détestent. Le résultat est le même.
Lien fort avec le ranking de votre site
➔ (rel=>nofollow)
Des liens sponsorisées que vous achetez.

Que faut-il analyser?

Ranking et autre liens de la page qui vous « linke »
Raison de la citation, mots clés des liens,..

Autres piliers du trafic

Ils sont multiples :

- ▶ Les campagnes emailing
- ▶ Les bloggeurs influents
- ▶ Affiliation (Afficher des produits en échange d'une commission sur les ventes)
- ▶ Nouveaux vecteurs

<http://bit.ly/1qUnEen>



Liens textuels et liens Produit

Créez un Lien vers un Produit Amazon ou encore d'un lien image unique

- Vos Liens intégreront votre site
- Vous pouvez y intégrer la publicité
- [Créez des Liens Produits](#)

Image uniquement

Texte et Image

Texte uniquement

Canon - Digital IXUS 80 IS appareil photo numérique - 8 MPix - Zoom optique 3x - stabilisé - argent

Achetez [amazon.fr](#)

Référencement naturel

SEO : Search Engine Optimization

- ▶ Optimiser son site pour les moteurs de recherche (généralement pour Google)
- ▶ Le but du référencement naturel est de se positionner au plus près du Top 10. Un site Internet (et ses pages) sur des mots-clés jugés utiles, pertinents et potentiellement rentables (gain de trafic, conversion en clients, ventes, etc.).

Référencement naturel

Etapes à suivre :

- ▶ Étudier et sélectionner avec soin les mots clés à cibler
- ▶ Venir optimiser la structure, le codage et le contenu du site dans le but qu'il contienne les mots clés retenus
- ▶ Augmenter le nombre de liens entrants vers le site visé et leurs qualités : c'est ce qu'on appelle le **netlinking**

Référencement naturel

Avantages :

- ▶ Les actions entreprises ont un effet sur le long terme.
- ▶ La visibilité est plus forte car multi supports (Google, Bing, Yahoo...).

Inconvénients :

- ▶ Les premiers résultats peuvent prendre quelques mois avant de se faire sentir.
- ▶ La modification du site Internet si celui-ci n'a pas été optimisé pour le référencement naturel au moment de sa conception.
- ▶ La recherche de liens entrants, et les actions SEO en général, demandent beaucoup de temps.

Référencement payant

SEA : Search Engine Advertising

- ▶ Dominé par Google avec Google AdWords, consiste tout simplement à acheter des mots clés auprès du moteur de recherche (ou de tout autre site).
- ▶ Cela permet donc, pour peu d'y mettre les moyens, d'être rapidement visible sur la première page de résultats d'un moteur de recherche.

Référencement payant

Etapes à suivre :



Référencement payant

Avantages :

- ▶ Vous pouvez être premier sur Google sur le mot clé de votre choix dès la mise en ligne de votre site.
- ▶ Vous avez un contrôle précis et optimisé de votre image et votre présence sur Google.
- ▶ Il est inutile (ou presque) d'optimiser le contenu de son site.

Référencement payant

Inconvénients :

- ▶ Le coût de sa mise en place.
- ▶ Si vous arrêtez de payer, vous disparaissez des pages de résultats (SERP). Il s'agit bien de publicité et non de référencement sur le long terme.
- ▶ Plus vous avez de concurrence, plus vous payez cher (principe des enchères).
- ▶ Vous devez analyser finement vos statistiques pour calculer le ROI (Retour Sur Investissement) de vos campagnes, les optimiser et les modifier en permanence, au risque de perdre de l'argent.

SEO ou SEA ?



Lequel choisir :

- ▶ Le SEO et le SEA ont tous les deux des avantages et des inconvénients. En fonction de vos contraintes ou de vos objectifs, vous pouvez choisir l'une ou l'autre solution.
- ▶ L'idéal sera donc de **combiner ces deux solutions** afin de bénéficier de leurs avantages respectifs.

SEO ou SEA ?



Exemple de stratégie à suivre :

- ▶ Commencer les actions de **netlinking** dès la mise en ligne du site Internet, dans le but d'améliorer son positionnement naturel sur les moteurs de recherche.
- ▶ Mettre en place une ou plusieurs campagnes **AdWords** pour obtenir rapidement du trafic qualifié sur le site.
- ▶ Au fur et à mesure que le référencement naturel porte ses fruits, diminuer les campagnes SEA jusqu'à trouver un équilibre entre SEO et SEA.

Référencement et positionnement

Référencement :

- Le référencement d'un site est l'action de signaler aux moteurs de recherches, aux annuaires, aux sites de réseaux sociaux, l'existence du site web.



Positionnement (classement) :

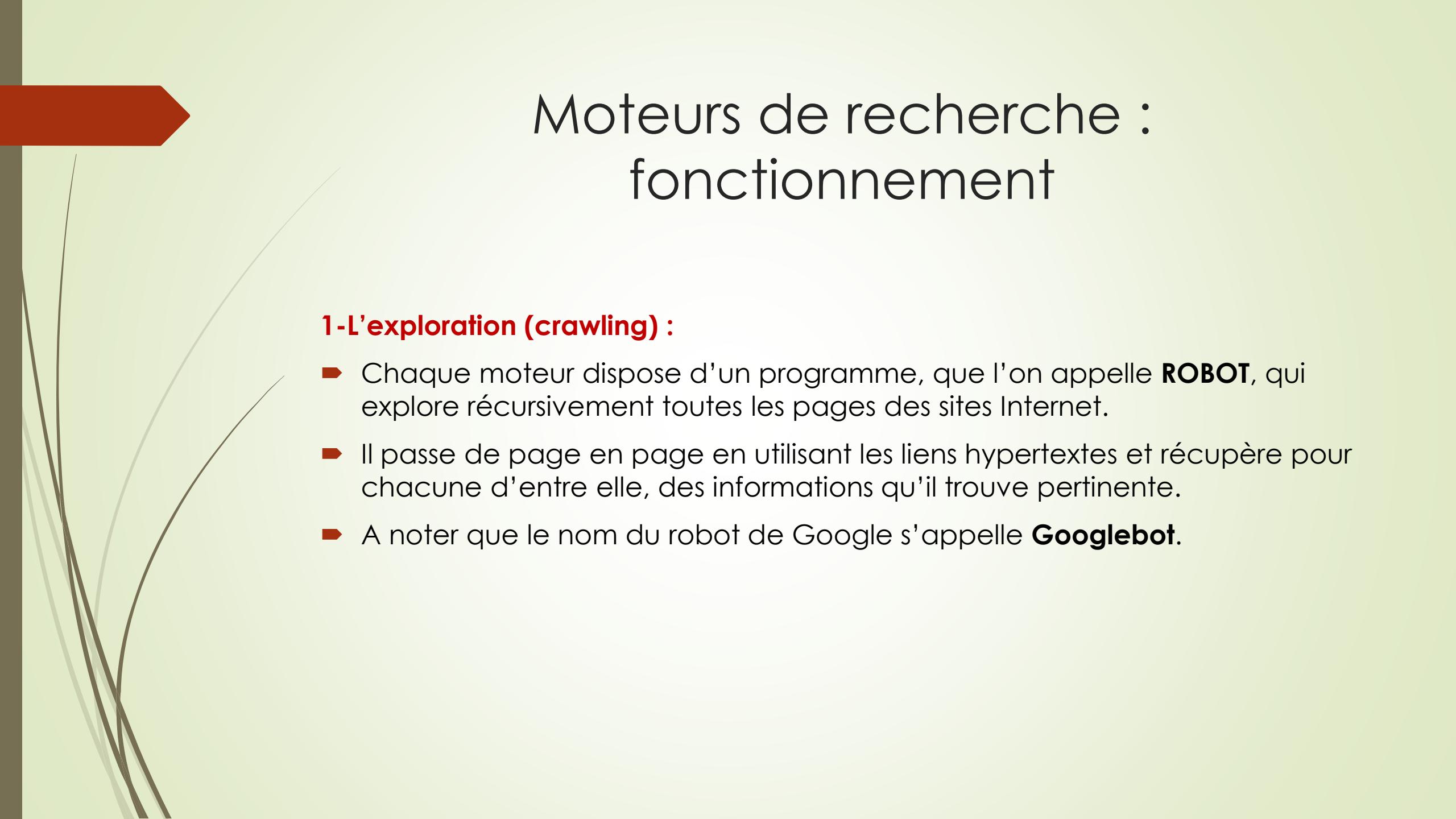
- Se désigne par l'action d'être dans les têtes de page des moteurs de recherche et sur des mots clés concurrentiels ou bien stratégiques afin d'assurer une visibilité maximale à un site Internet.

Nous avons pour habitude de parler du **référencement** d'un site internet, pour désigner en réalité le **positionnement** du site web en question.

Algorithme PageRank

- ▶ Larry Page et Sergey Brin deux étudiants développent « backrub » un moteur de recherche basé sur l'algorithme **PageRank**.
- ▶ Cet algorithme calcule pour chaque page un indice basé sur les liens entrants (0 à 10).
- ▶ $\text{PR}(\text{page } x) = \sum \text{PR}(\text{les pages qui amènent vers } x)$
- ▶ Principe : On mesure l'importance d'un article au nombre de fois qu'il est cité dans d'autres articles

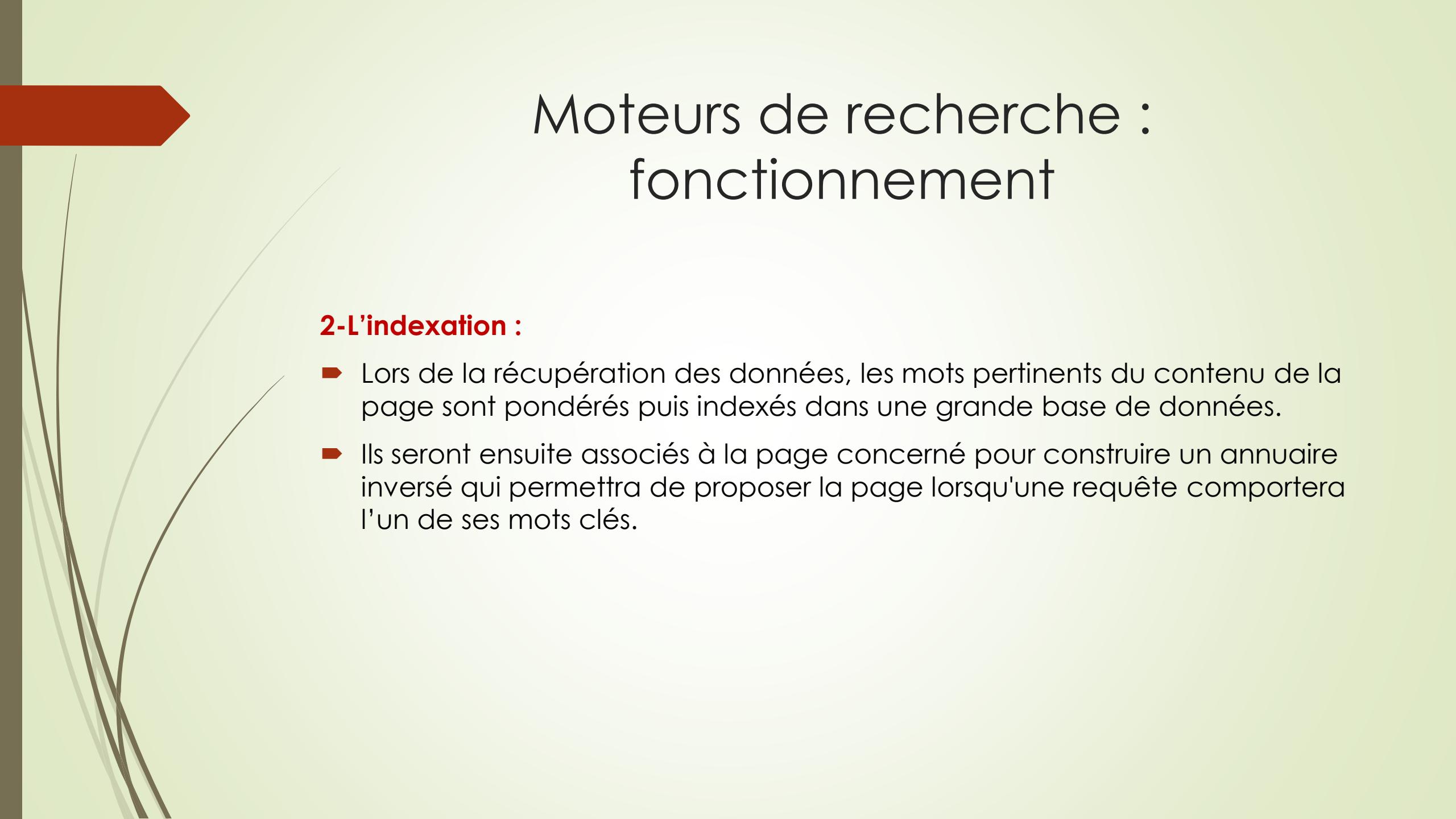
The Google logo is displayed in its signature multi-colored font. The letters are arranged in a bold, sans-serif style. The colors used are blue, red, yellow, and green.



Moteurs de recherche : fonctionnement

1-L'exploration (crawling) :

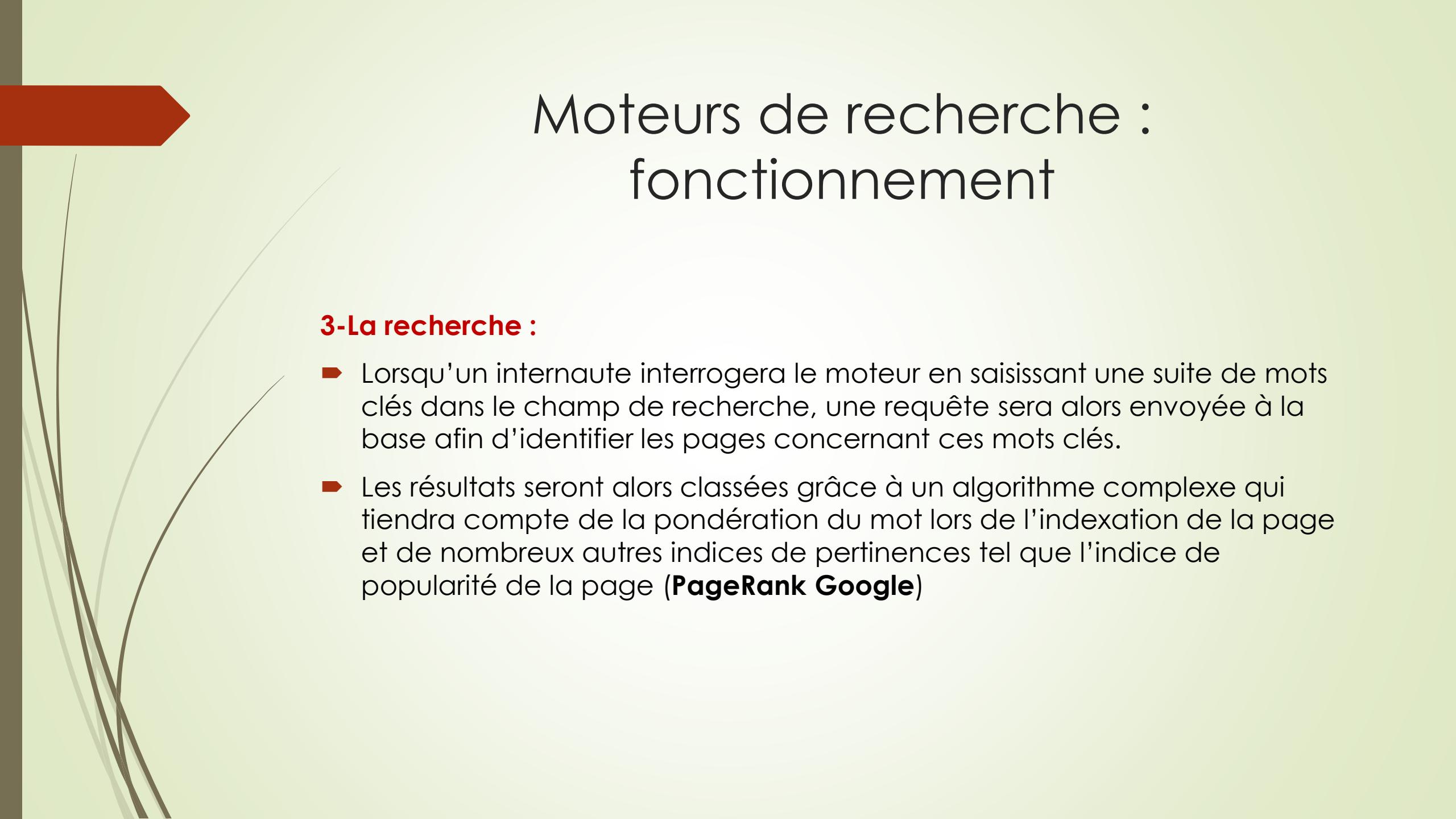
- ▶ Chaque moteur dispose d'un programme, que l'on appelle **ROBOT**, qui explore récursivement toutes les pages des sites Internet.
- ▶ Il passe de page en page en utilisant les liens hypertextes et récupère pour chacune d'entre elle, des informations qu'il trouve pertinente.
- ▶ A noter que le nom du robot de Google s'appelle **Googlebot**.



Moteurs de recherche : fonctionnement

2-L'indexation :

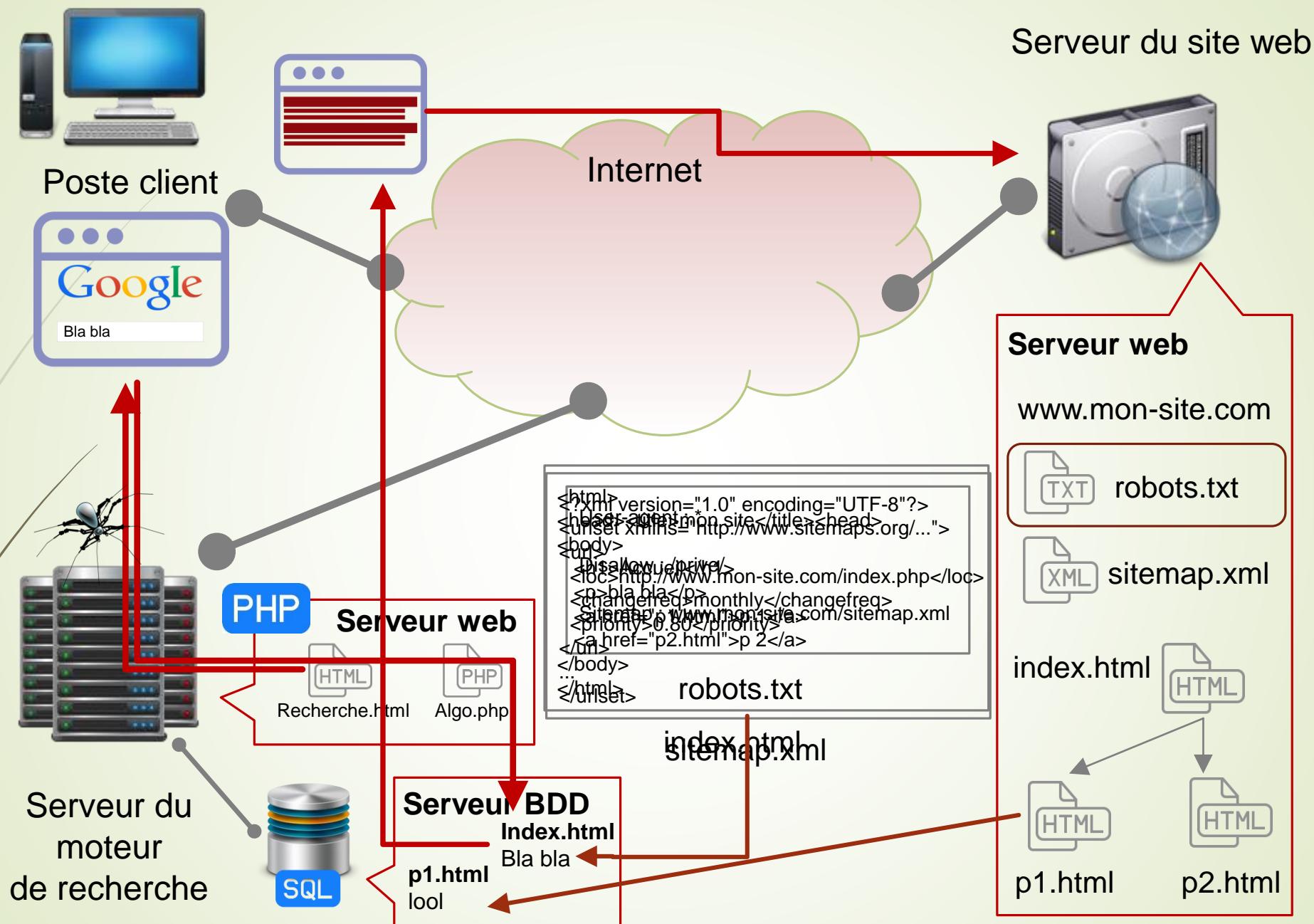
- ▶ Lors de la récupération des données, les mots pertinents du contenu de la page sont pondérés puis indexés dans une grande base de données.
- ▶ Ils seront ensuite associés à la page concerné pour construire un annuaire inversé qui permettra de proposer la page lorsqu'une requête comportera l'un de ses mots clés.



Moteurs de recherche : fonctionnement

3-La recherche :

- ▶ Lorsqu'un internaute interrogera le moteur en saisissant une suite de mots clés dans le champ de recherche, une requête sera alors envoyée à la base afin d'identifier les pages concernant ces mots clés.
- ▶ Les résultats seront alors classées grâce à un algorithme complexe qui tiendra compte de la pondération du mot lors de l'indexation de la page et de nombreux autres indices de pertinences tel que l'indice de popularité de la page (**PageRank Google**)



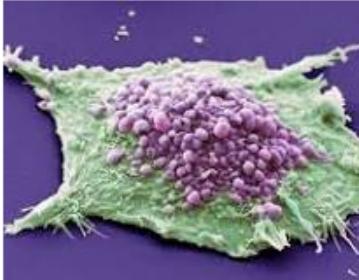
Page SERP de Google

Google c'est quoi le cancer?

Tous Actualités Images Vidéos Plus Paramètres Outils

Environ 1 650 000 résultats (0,51 secondes)

Le **cancer** est une maladie caractérisée par une prolifération cellulaire, ou tumeur maligne, anormalement importante formée à partir de la transformation par mutation ou instabilité génétique d'une cellule initialement normale. 17 févr. 2016



Cancer : définition, causes, types, qu'est-ce que c'est ?
www.maxisciences.com/cancer/cancer-definition-causes-types-qu-039-est-ce-que-c-039-...

À propos de ce résultat • Commentaires

Qu'est-ce que le cancer ? | Ligue contre le Cancer
[https://www.ligue-cancer.net/article/278_qu-est-ce-que-le-cancer-...](https://www.ligue-cancer.net/article/278_qu-est-ce-que-le-cancer-)

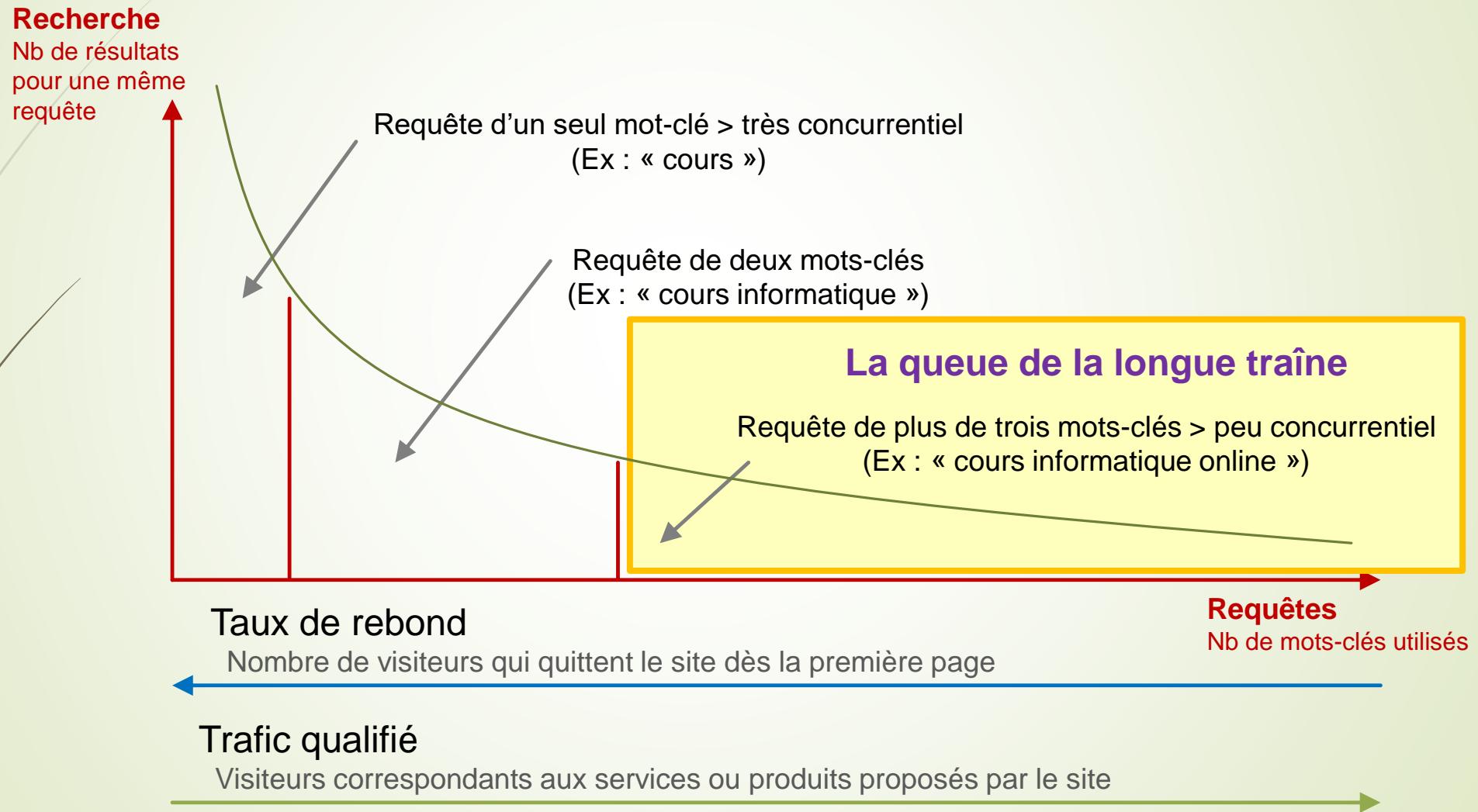
Retrouvez toutes les notions de bases sur les cancers, les cellules ... C'est une cellule qui devient totalement indisciplinée, suite à une agression ou un ...

Référencement naturel

Avant d'optimiser vos pages, il faut établir une stratégie de mots-clés afin de bien définir sur quelles requêtes de recherche vous désirez être visible.

- ▶ Connaître la théorie de la longue traîne,
- ▶ Utiliser des outils de choix des groupes de mots-clés,
- ▶ Vérifier la faisabilité d'utilisation d'un groupe de mots-clés dans le moteur,
- ▶ Terminer la procédure en finalisant une liste de 25 groupes de mots-clés.

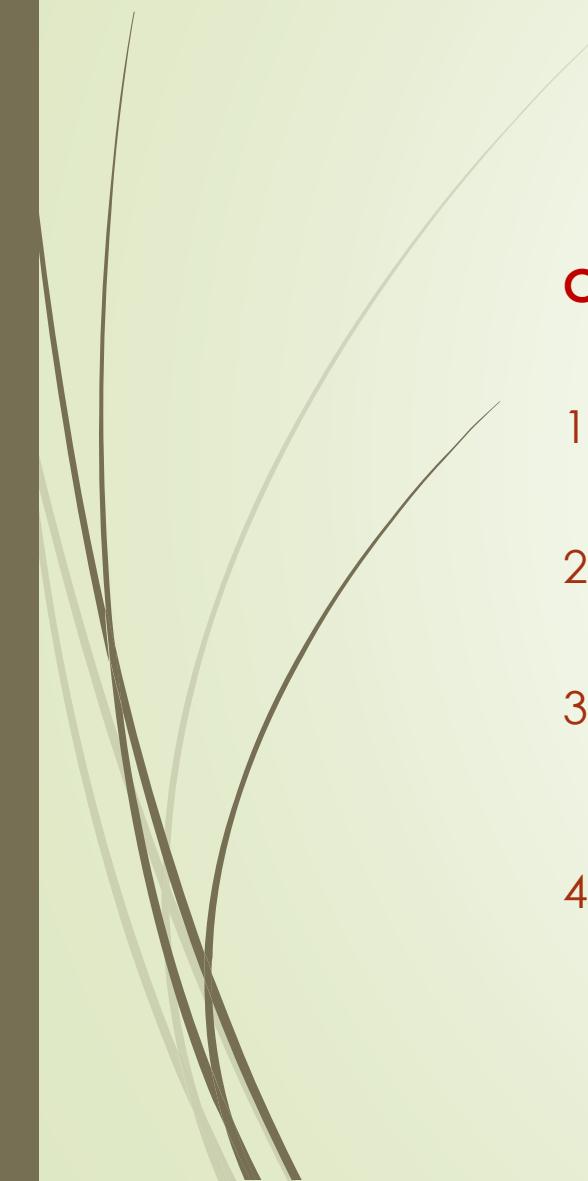
Théorie de la longue traîne





Choix des groupes de mots-clés

- ▶ Les moteurs de recherche comme Google, Bing et Yahoo tiennent compte de l'optimisation de votre site.
- ▶ Le premier travail pour mener à bien cette optimisation passe par la recherche des **bons mots-clés** qui ciblent votre secteur d'activité.
- ▶ Pour la recherche des mots-clés, le principe est de trouver les bon mots-clés qui correspondent à votre activité, vos produits et/ou services.
- ▶ Lorsque l'internaute va entrer ces mots-clés, il est susceptible d'arriver vers votre site si celui-ci a bien été optimisé avec ces mots-clés.



Choix des groupes de mots-clés

Comment procéder ?

1. Quelles questions l'internaute se pose t-il ?
2. .. Et donc : quelles recherches tape-t-il ?
3. Compiler les mots et phrases-clés correspondantes, chercher les variantes, le nombre de recherches ...
4. Les prioriser et les trier.

Outils de choix des groupes de mots-clés

- ▶ Comment trouver les bon mots-clés pour optimiser un site?

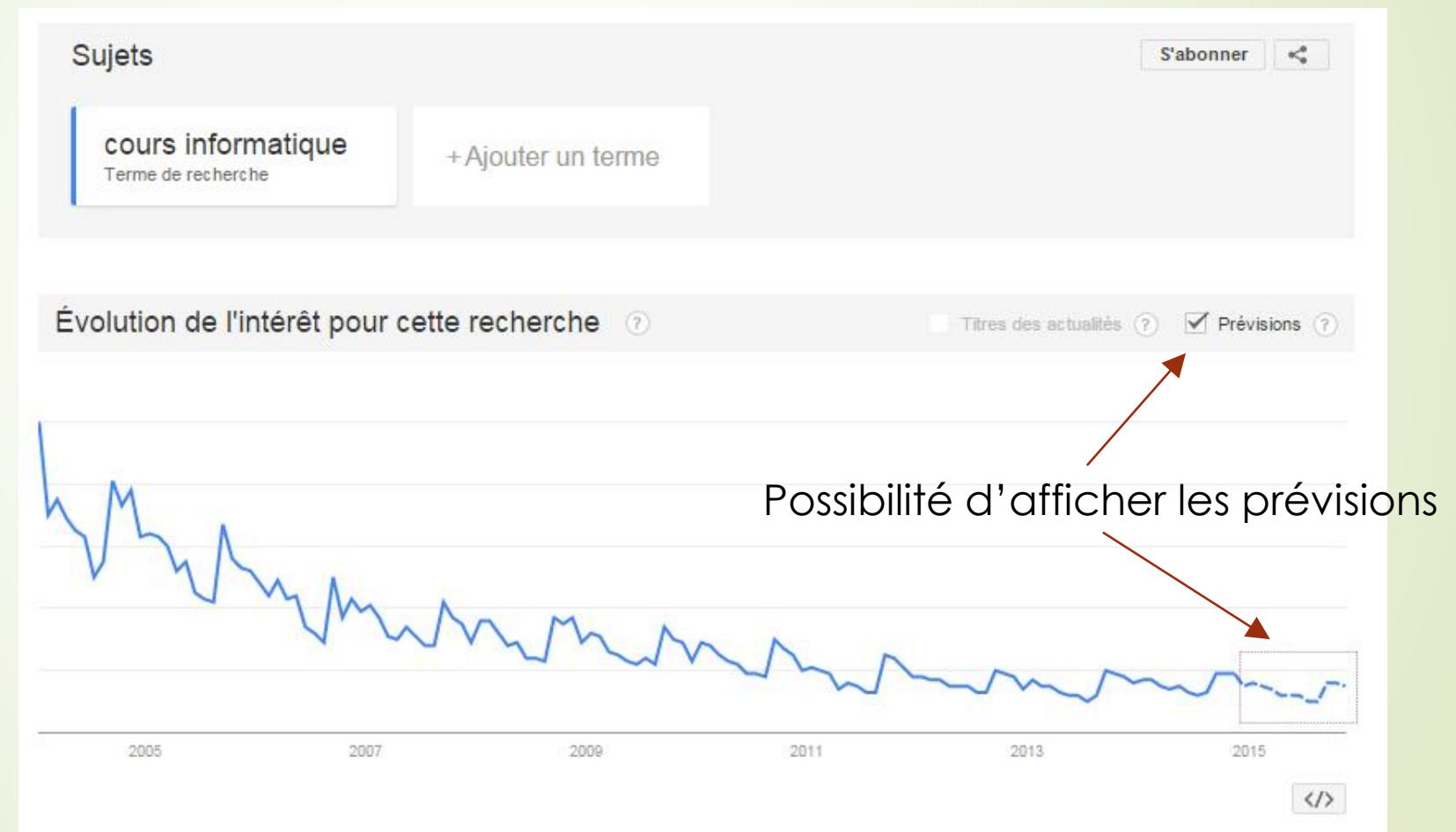


Google Tendances des recherches

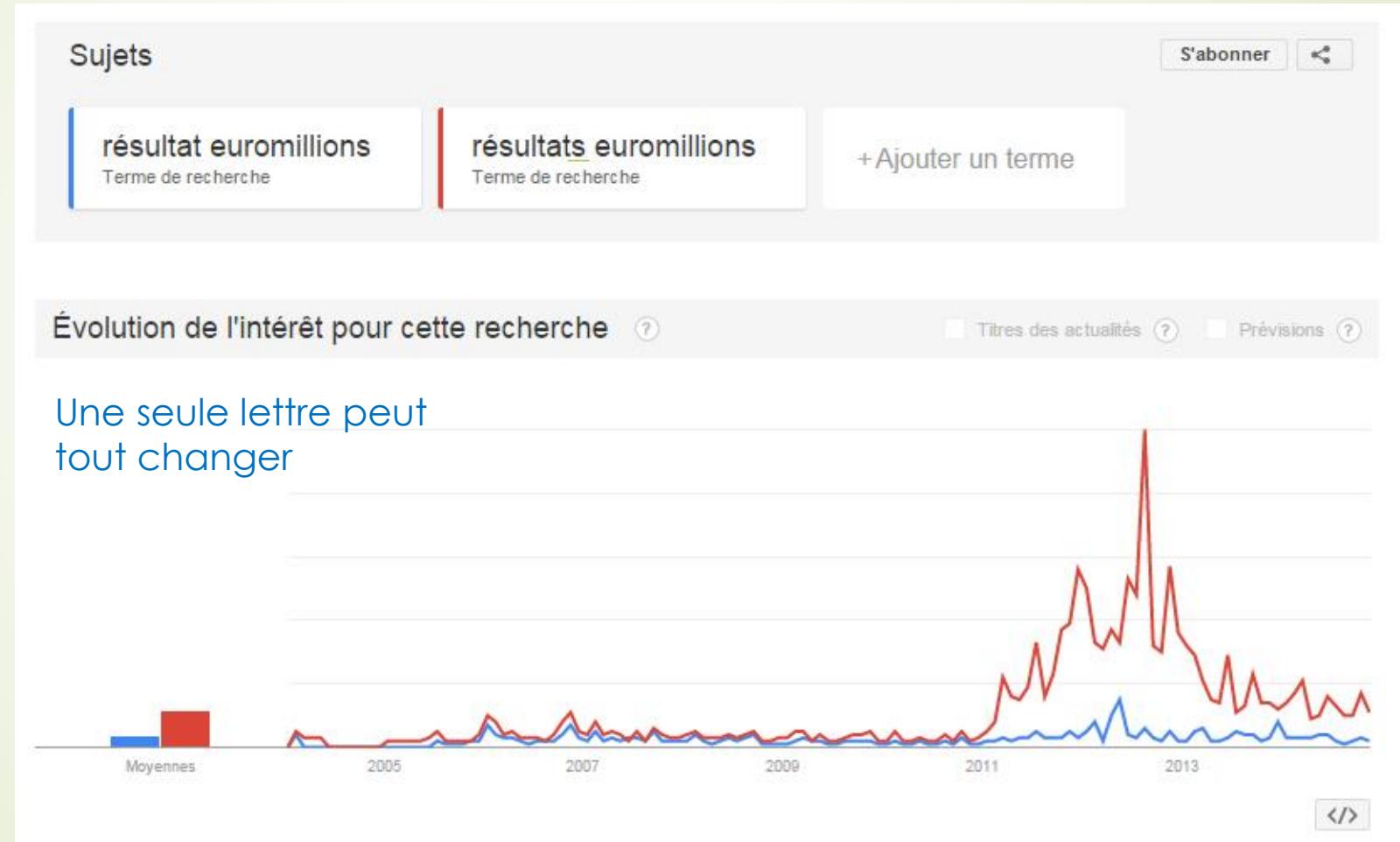
<http://www.google.com/trends/>

Utilité : Etude de marché en fonction des mots-clés

Google Trends



Google Trends



Google Trends

► Démonstration <http://www.google.com/trends/>



Outils de choix des groupes de mots-clés

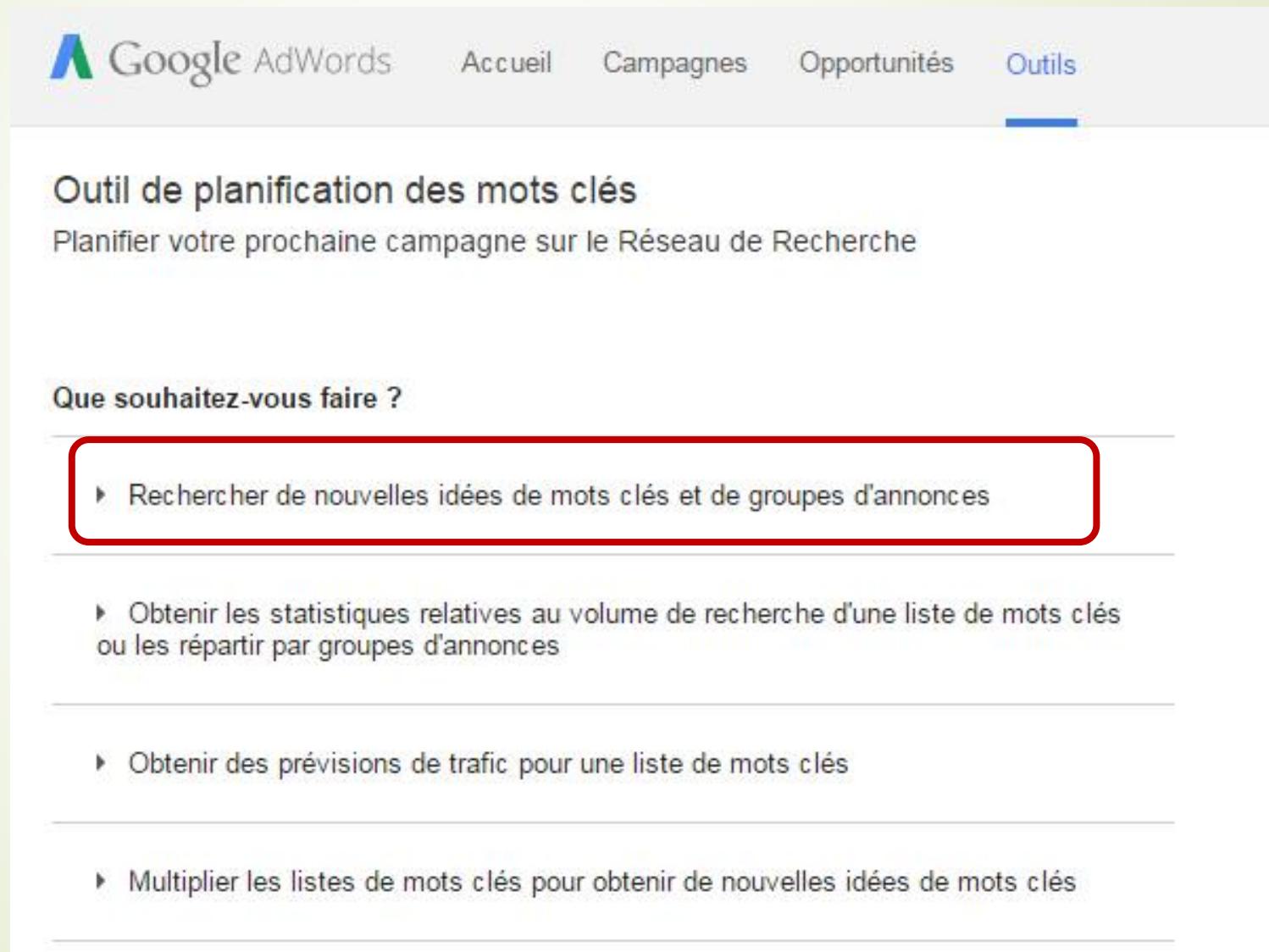
- ▶ Le générateur des mots-clés **Google AdWords**

<http://adwords.google.com/KeywordPlanner>

- ▶ Trouver des mots-clés connexes
- ▶ Analyser un site
- ▶ Voir quelle est la demande
- ▶ Analyser la concurrence



Google AdWords



The image shows a screenshot of the Google AdWords interface. At the top, there is a navigation bar with the Google AdWords logo and links for Accueil, Campagnes, Opportunités, and Outils. The 'Outils' link is underlined, indicating it is the active page. Below the navigation bar, the title 'Outil de planification des mots clés' is displayed, followed by the subtitle 'Planifier votre prochaine campagne sur le Réseau de Recherche'. A section titled 'Que souhaitez-vous faire ?' contains a list of five items, with the first item highlighted by a red rectangular border. The list items are:

- ▶ Rechercher de nouvelles idées de mots clés et de groupes d'annonces
- ▶ Obtenir les statistiques relatives au volume de recherche d'une liste de mots clés ou les répartir par groupes d'annonces
- ▶ Obtenir des prévisions de trafic pour une liste de mots clés
- ▶ Multiplier les listes de mots clés pour obtenir de nouvelles idées de mots clés

Google AdWords

The screenshot shows the Google AdWords Keyword Planner interface. At the top, there's a navigation bar with 'Accueil', 'Campagnes', 'Opportunités', and 'Outils' (selected). Below it, a search bar prompts 'Saisissez l'un ou plusieurs des éléments suivants :'. A blue box highlights the 'Votre produit ou service' field containing 'produit informatique' and 'matériels informatique'. To the right, a callout box says 'Tapez un ou plusieurs mots-clés' with a green arrow pointing to the search bar. The interface includes sections for 'Votre page de destination' (with 'www.example.com/page') and 'Votre catégorie de produit' (with a dropdown menu). On the left, a sidebar titled 'Autres filtres que vous pouvez utiliser' has a green bracket pointing to its filters: 'Ciblage' (with 'Tous les pays et territoires', 'Toutes les langues', 'Google', and 'Mots clés à exclure'), 'Période' (with 'Afficher le nombre moy. de recherches mensuelles pour : Les 12 derniers mois'), and 'Personnaliser votre recherche' (with 'Filtres de mots clés', 'Options relatives aux mots clés' (with 'Afficher les idées indirectement liées' and 'Masquer les mots clés dans mon compte'), and 'Mots clés à inclure'). At the bottom is a blue button labeled 'Obtenir des idées'.

Autres filtres que vous pouvez utiliser

Ciblage

- Tous les pays et territoires
- Toutes les langues
- Google
- Mots clés à exclure

Période

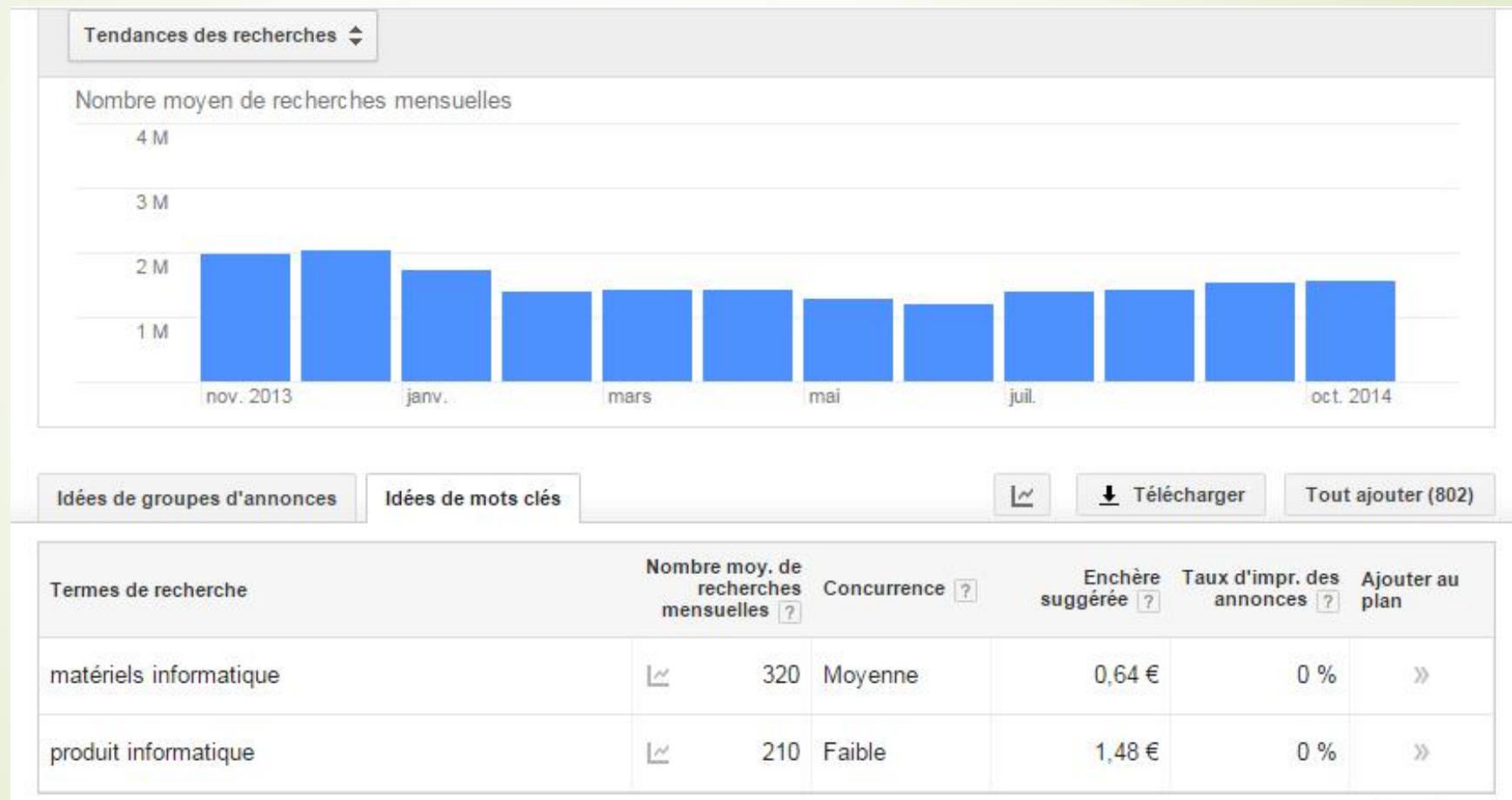
- Afficher le nombre moy. de recherches mensuelles pour : Les 12 derniers mois

Personnaliser votre recherche

- Filtres de mots clés
- Options relatives aux mots clés
 - Afficher les idées indirectement liées
 - Masquer les mots clés dans mon compte
- Mots clés à inclure

Obtenir des idées

Google AdWords

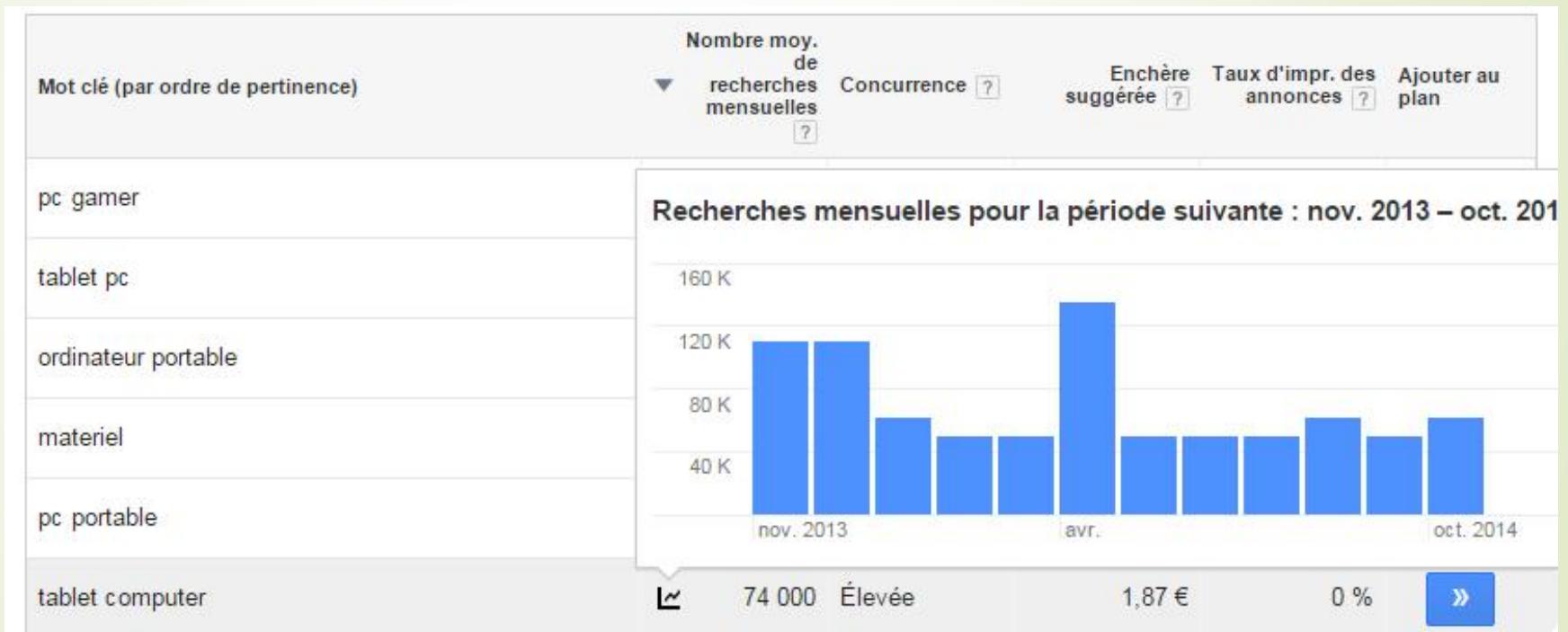


Google AdWords

Mot clé (par ordre de pertinence)	Nombre moy. de recherches mensuelles	Concurrence	Enchère suggérée	Taux d'impr. des annonces	Ajouter au plan
pc gamer	246 000	Moyenne	0,35 €	0 %	»
tablet pc	201 000	Élevée	1,32 €	0 %	»
ordinateur portable	165 000	Élevée	0,86 €	0 %	»
materiel	110 000	Faible	0,61 €	0 %	»
pc portable	74 000	Élevée	0,67 €	0 %	»
tablet computer	74 000	Élevée	1,87 €	0 %	»

Mettez le curseur de votre souris sur cette icône

Google AdWords



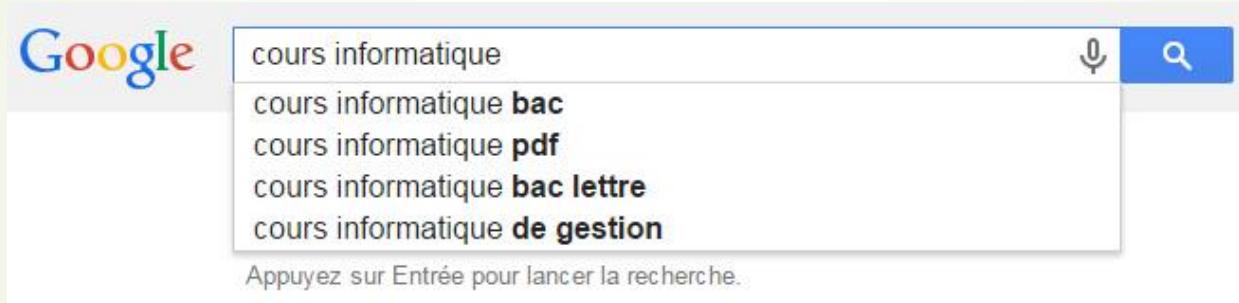
Google AdWords

- Démonstration <http://adwords.google.com/KeywordPlanner>



Autres techniques

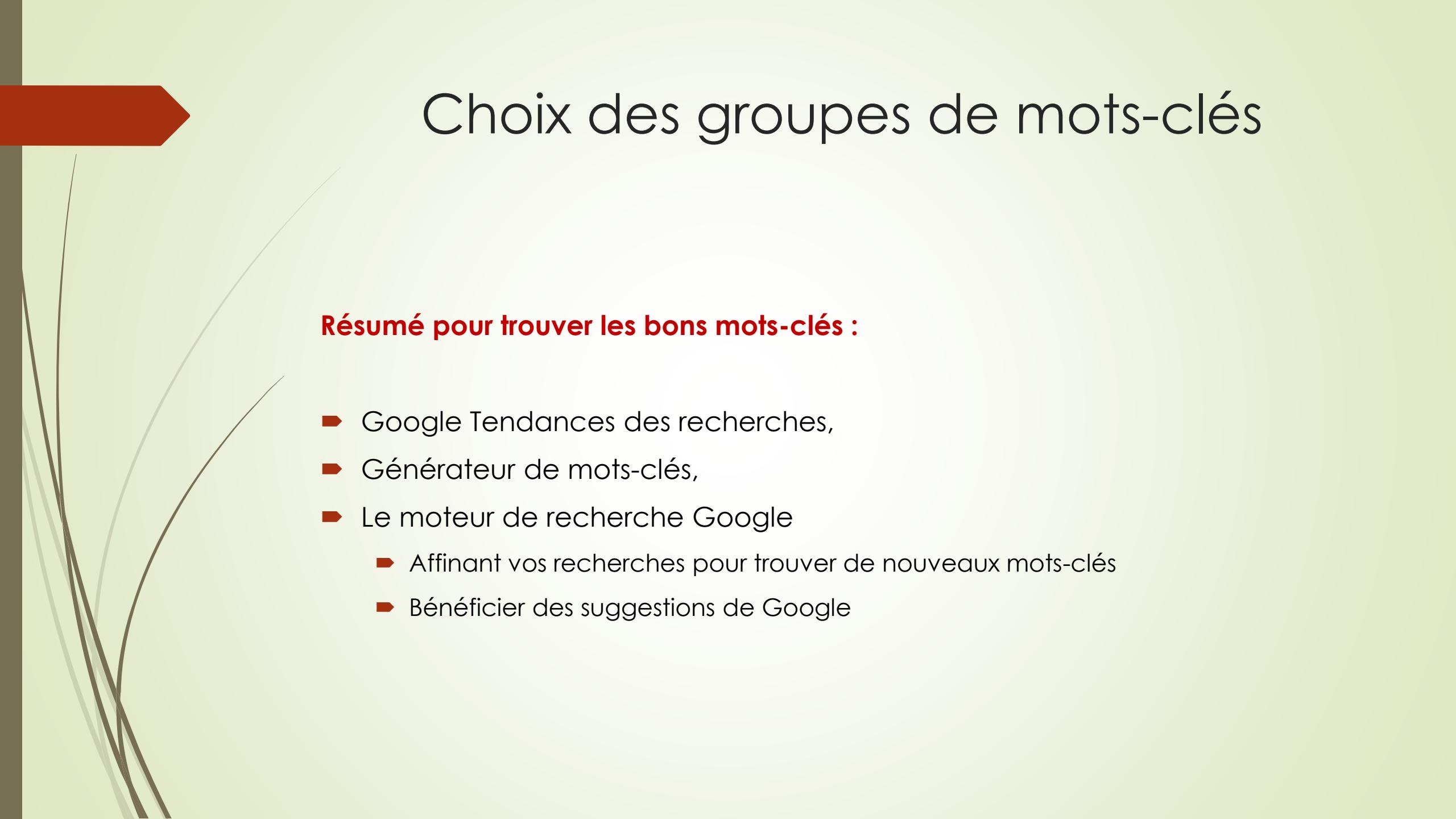
- ▶ Affiner vos recherches pour découvrir de nouvelles requêtes tapées dans votre thématique.



- ▶ Utiliser les suggestions de Google

Recherches associées à cours informatique

cours informatique pdf	telecharger cours informatique
cours informatique debutant	cours word
cours informatique gratuit	cours informatique en ligne
cours informatique excel	cours informatique video



Choix des groupes de mots-clés

Résumé pour trouver les bons mots-clés :

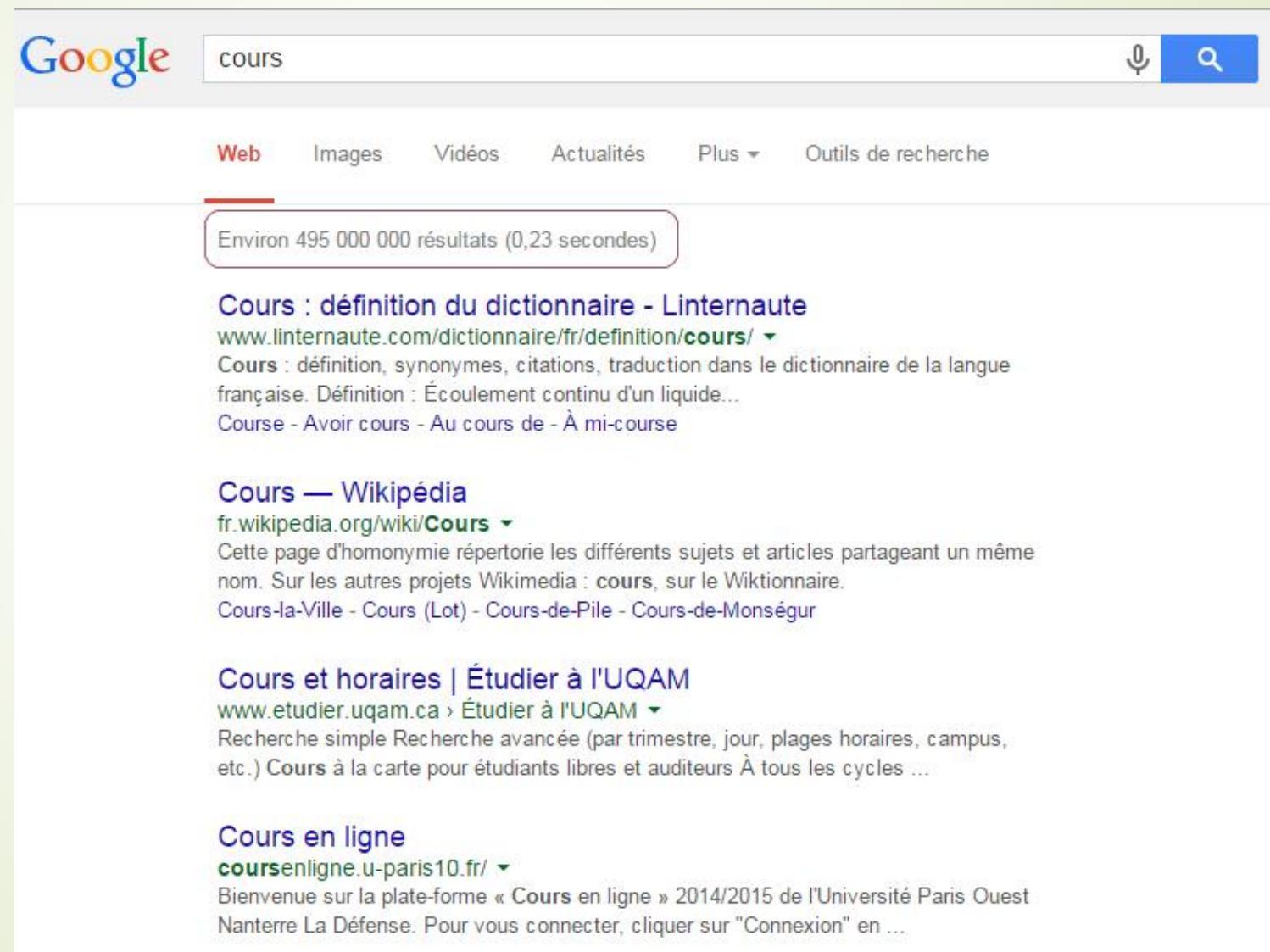
- ▶ Google Tendances des recherches,
- ▶ Générateur de mots-clés,
- ▶ Le moteur de recherche Google
 - ▶ Affinant vos recherches pour trouver de nouveaux mots-clés
 - ▶ Bénéficier des suggestions de Google



Faisabilité d'utilisation des groupes de mots-clés

- ▶ Il faut trouver un compromis entre une requête fréquemment utilisée et celle qui génère trop de concurrence pour avoir une chance d'être positionné.
- ▶ Pour être plus concret, voici des références de résultats pour une requête qui vous donneront une idée de vos chances d'être positionné :
 - ▶ **De 50 000 à 200 000 : Bon compromis pour être positionné**
 - ▶ **De 200 000 à 800 000 : Positionnement possible mais avec plusieurs optimisations successives**
 - ▶ **Plus de 800 000 : requête trop concurrentielle pour avoir des chances d'être bien positionné**

Exemple



Google cours

Web Images Vidéos Actualités Plus Outils de recherche

Environ 495 000 000 résultats (0,23 secondes)

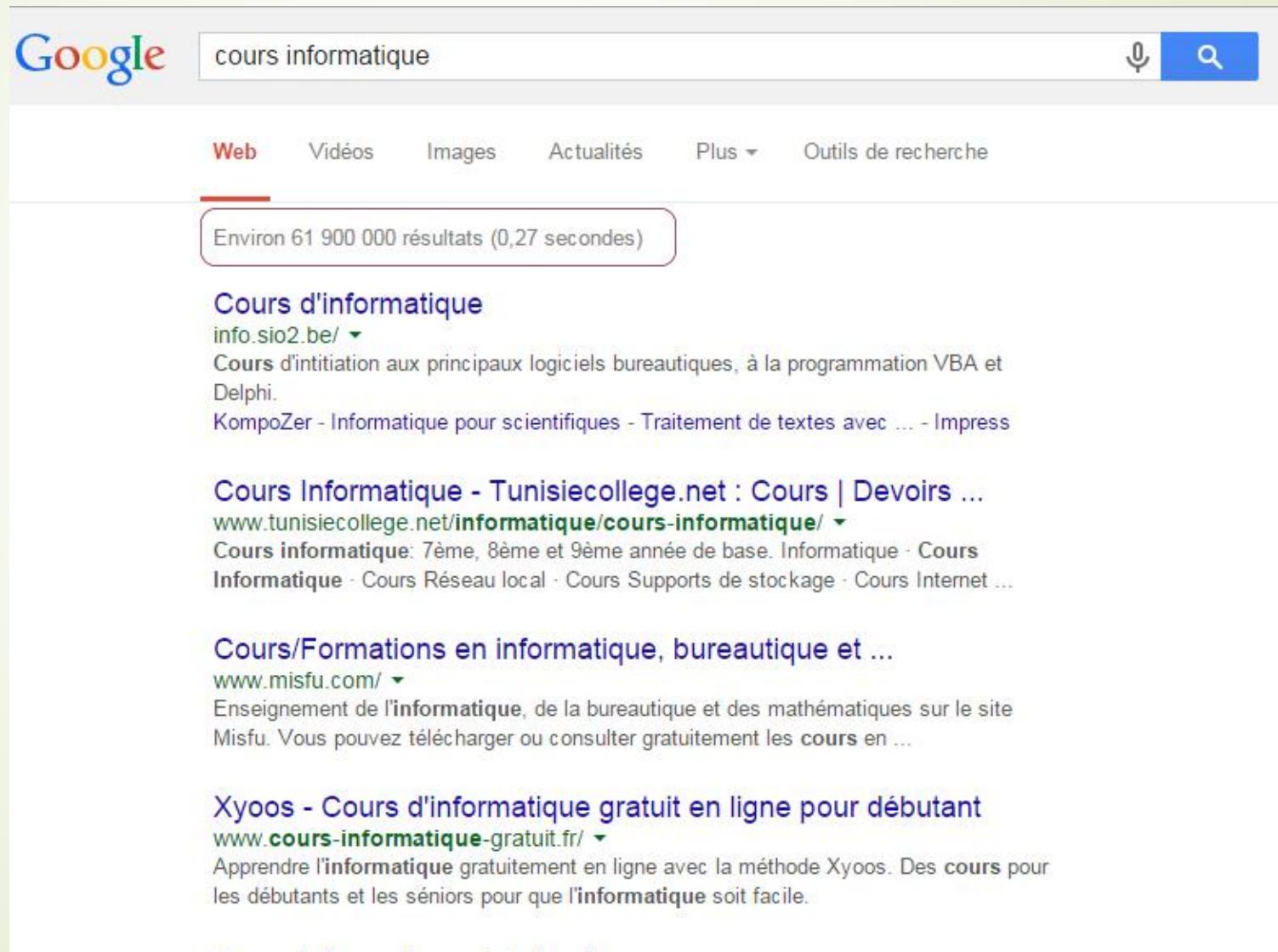
Cours : définition du dictionnaire - L'internaute
www.linternaute.com/dictionnaire/fr/definition/cours/ ▾
Cours : définition, synonymes, citations, traduction dans le dictionnaire de la langue française. Définition : Écoulement continu d'un liquide...
Course - Avoir cours - Au cours de - À mi-course

Cours — Wikipédia
fr.wikipedia.org/wiki/Cours ▾
Cette page d'homonymie répertorie les différents sujets et articles partageant un même nom. Sur les autres projets Wikimedia : cours, sur le Wiktionnaire.
Cours-la-Ville - Cours (Lot) - Cours-de-Pile - Cours-de-Monségur

Cours et horaires | Étudier à l'UQAM
www.etudier.uqam.ca › Étudier à l'UQAM ▾
Recherche simple Recherche avancée (par trimestre, jour, plages horaires, campus, etc.) Cours à la carte pour étudiants libres et auditeurs À tous les cycles ...

Cours en ligne
coursenligne.u-paris10.fr/ ▾
Bienvenue sur la plate-forme « Cours en ligne » 2014/2015 de l'Université Paris Ouest Nanterre La Défense. Pour vous connecter, cliquer sur "Connexion" en ...

Exemple



Google cours informatique

Web Vidéos Images Actualités Plus Outils de recherche

Environ 61 900 000 résultats (0,27 secondes)

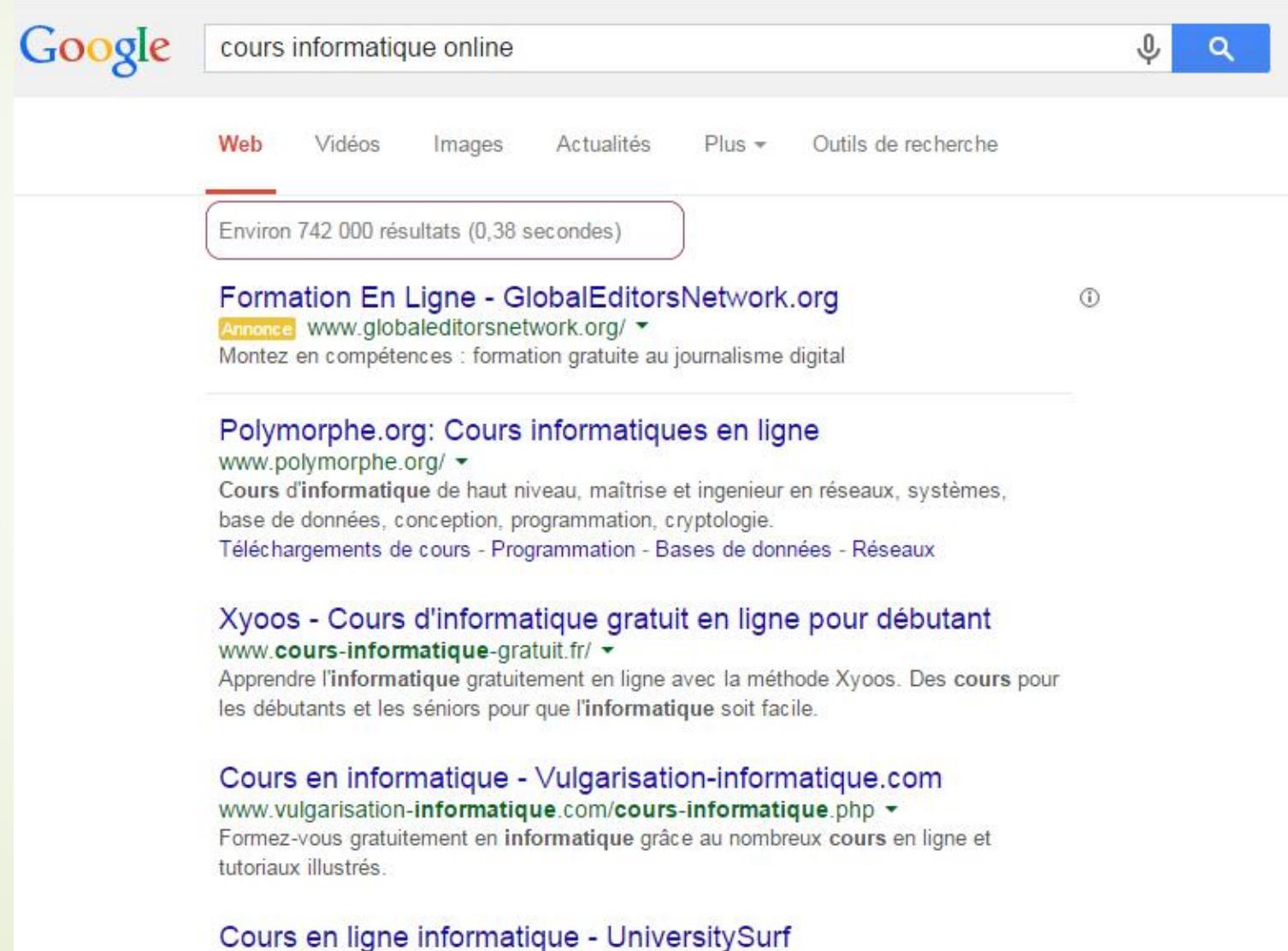
Cours d'informatique
info.sio2.be/ ▾
Cours d'introduction aux principaux logiciels bureautiques, à la programmation VBA et Delphi.
KompoZer - Informatique pour scientifiques - Traitement de textes avec ... - Impress

Cours Informatique - Tunisiecollege.net : Cours | Devoirs ...
www.tunisiecollege.net/informatique/cours-informatique/ ▾
Cours informatique: 7ème, 8ème et 9ème année de base. Informatique · Cours Informatique · Cours Réseau local · Cours Supports de stockage · Cours Internet ...

Cours/Formations en informatique, bureautique et ...
www.misfu.com/ ▾
Enseignement de l'informatique, de la bureautique et des mathématiques sur le site Misfu. Vous pouvez télécharger ou consulter gratuitement les cours en ...

Xyoos - Cours d'informatique gratuit en ligne pour débutant
www.cours-informatique-gratuit.fr/ ▾
Apprendre l'informatique gratuitement en ligne avec la méthode Xyoos. Des cours pour les débutants et les séniors pour que l'informatique soit facile.

Exemple



Google cours informatique online

Web Vidéos Images Actualités Plus Outils de recherche

Environ 742 000 résultats (0,38 secondes)

Formation En Ligne - GlobalEditorsNetwork.org
Annonce www.globaleditorsnetwork.org/ ▾
Montez en compétences : formation gratuite au journalisme digital

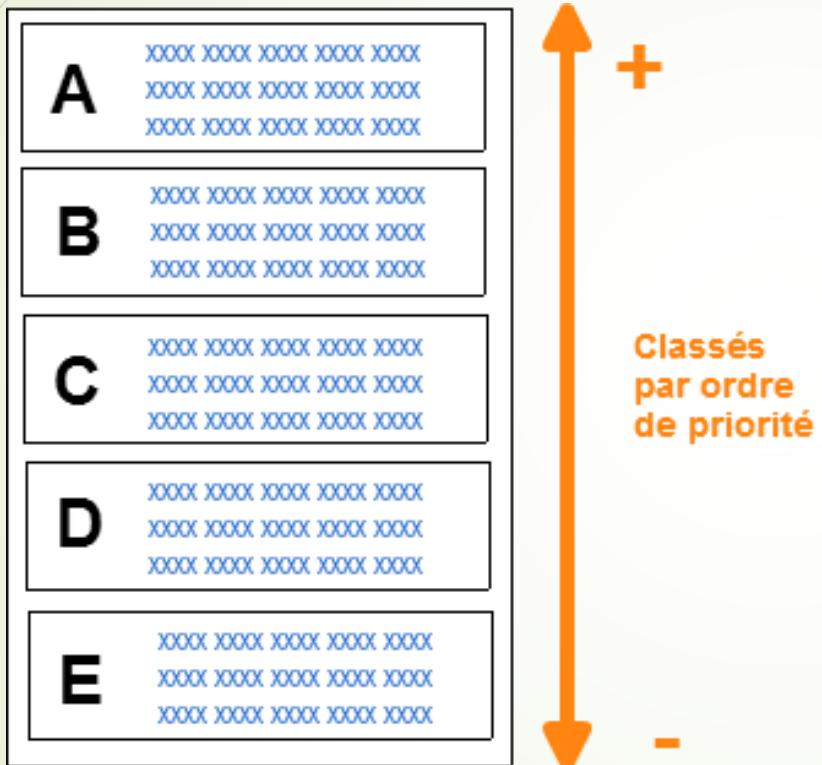
Polymorphe.org: Cours informatiques en ligne
www.polymorphe.org/ ▾
Cours d'informatique de haut niveau, maîtrise et ingénierie en réseaux, systèmes, base de données, conception, programmation, cryptologie.
Téléchargements de cours - Programmation - Bases de données - Réseaux

Xyoos - Cours d'informatique gratuit en ligne pour débutant
www.cours-informatique-gratuit.fr/ ▾
Apprendre l'informatique gratuitement en ligne avec la méthode Xyoos. Des cours pour les débutants et les seniors pour que l'informatique soit facile.

Cours en informatique - Vulgarisation-informatique.com
www.vulgarisation-informatique.com/cours-informatique.php ▾
Formez-vous gratuitement en informatique grâce aux nombreux cours en ligne et tutoriaux illustrés.

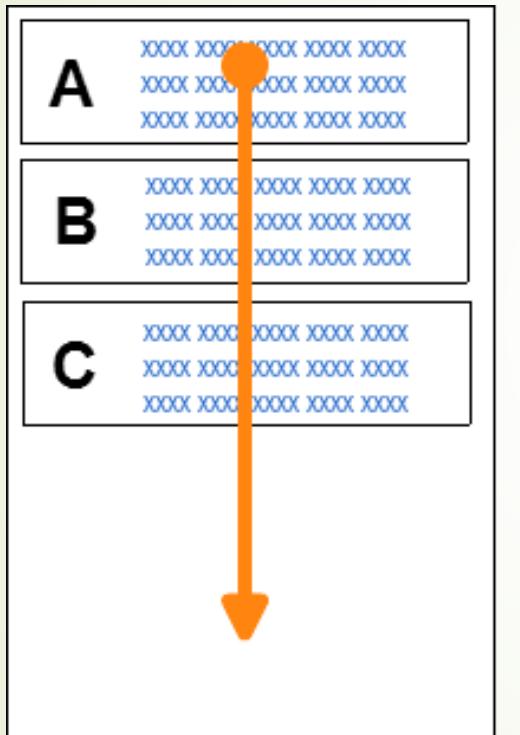
Cours en ligne informatique - UniversitySurf

En conclusion



- ▶ Après avoir utilisé les outils de choix et vous être assuré de la faisabilité d'utilisation de chaque groupe de mots-clés, vous devez obtenir une liste de 25 groupes classés par ordre de priorité et découpés en 5 blocs : A, B,...

En conclusion

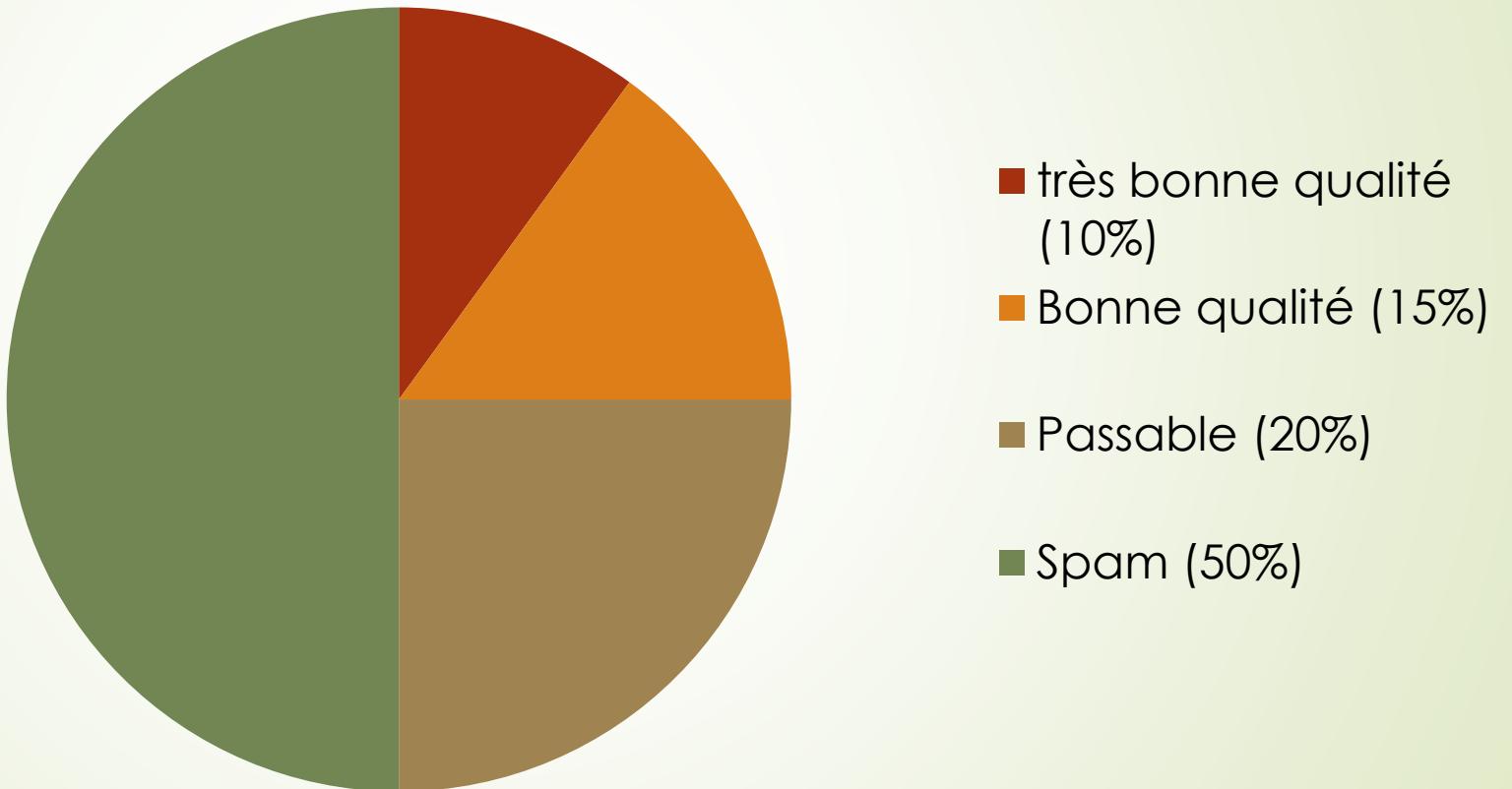


Cohérence verticale des groupes de mots-clés dans la page

- ▶ Ces différents blocs de groupes seront utilisés à différents endroits dans la code de la page à optimiser.
- ▶ Aussi pour conserver une cohérence verticale, vous devez toujours prélever les groupes dans cette liste en commençant par le bloc de priorité la plus élevée (donc le bloc A)

Contenu sur le web

Qualité du contenu sur le web



Problème : Comment les indexeurs évaluent le contenu du web

Le TrustRank

- ▶ Le **TrustRank** est l'un des nombreux critères de pertinence de Google.
- ▶ Il est traduit en français par « indice de confiance » ou « crédibilité d'un site »
- ▶ Méthode semi-automatique pour détecter les pages de spam = classification "spam ou pas spam" (Trust = confiance - Le terme TrustRank vient de Yahoo!).
- ▶ **Principe : une page « propre » ne propose pas de liens vers des pages de spam**



Le TrustRank

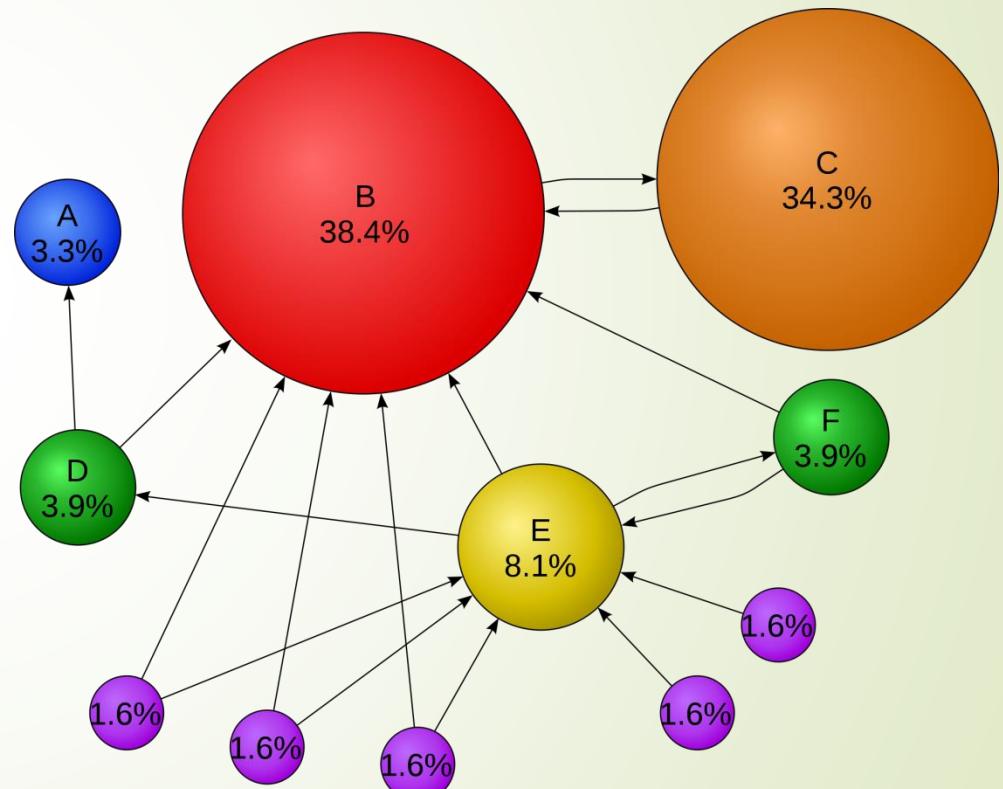
1. Amorçage : établir une liste de pages « propres » de référence
 - ▶ Après une analyse « humaine ».
 - ▶ On n'a pas forcément besoin d'une grande liste (ex 200 sites).
2. Suivi récursif des liens de la liste d'amorçage
3. Degré de confiance que l'on peut attribuer à la page : un indice
 - ▶ Plus les liens sont forts avec des pages de référence, plus leur degré de confiance est élevé
 - ▶ C'est le TrustRank (ou TR), indice entre 0 (=spam) et 1 (=page de référence)

Le TrustRank

- ▶ Google prend en considération dans son algorithme TrustRank l'audience d'un site.
 - ▶ Un site ayant un taux de rebond très élevé de son audience naturelle recevrait une note basse.
 - ▶ Inversement, un site possédant un faible taux de rebond et une forte fidélisation recevrait une note élevée.
- ▶ Tester le TrustRank d'une page :
<http://www.seomastering.com/trust-rank-checker.php>

La PageRank

- ▶ « Vote » d'une page pour une autre
 $PR(C) > PR(E)$ même avec moins de liens
- ▶ Échelle logarithmique 0-10
La barre d'outil Google permet d'en visualiser une approximation,
Avoir une PR 3-4 c'est déjà beaucoup
Extension : <http://bit.ly/1pwMrYK>
- ▶ Un processus de calcul récursif
→ pour éviter que le PR $\rightarrow \infty$, il faut un amortissement



La PageRank reste un secret

- ▶ Un nombre important de facteurs est pris en compte dans le PageRank :
 - ▶ Ces facteurs sont très nombreux (plus de 200 d'après Google).
 - ▶ Leur nature et leur pondération sont secrets pour limiter les chances de manipulations (et la concurrence des autres moteurs de recherche).
 - ▶ Le terme "PageRank" est une marque déposée et a été l'objet d'un brevet.

Quelques paramètres de calcul du PR

1. Sur la page (**onpage**)
 - ▶ Ancienneté / fréquence d'actualisation
 - ▶ Texte visible sur la page / code, les métas qui ne sont pas visibles
2. Sur le site (**onsite**)
 - ▶ Lien internes, Arborescence, fil d'Ariane,...
 - ▶ Paramétrage sur Google outils pour les webmasters (Sitemap..)
3. Hors du site (**offsite**)
 - ▶ Les liens entrants (calcul du TrustRank et le PageRank)
 - ▶ Réseaux sociaux
 - ▶ Publication et tweeting

PageRank : la formule

$$\text{PR}(B) = (1-d) + d \times (\text{PR}(A_1) / N(A_1) + \dots + \text{PR}(A_n) / N(A_n))$$

- ▶ Soient A_1, A_2, \dots, A_n : n pages pointant vers une page B.
- ▶ Notons $\text{PR}(A_k)$ le Page Rank de la page A_k
- ▶ $N(A_k)$ le nombre de liens sortants présents sur la page A_k
- ▶ 'd' est un coefficient d'amortissement choisi arbitrairement entre 0 et 1 (généralement 0.85), permettant de limiter l'importance de l'addition de tous les facteurs $\text{PR}(A_n)/N(A_n)$.
- ▶ Le facteur $(1-d)$ permet de garantir que la moyenne des Page Rank de l'ensemble des pages du web sera de 1.



Le SERP Rank

- ▶ **SERP = Search Engine Results Page**
- ▶ C'est l'ordre de présentation des liens lorsque l'on entre des mots-clés dans un moteur de recherche
 - ▶ La page de résultats présente une liste ordonnée de liens vers des pages/images/vidéos associés à des textes courts (snippets)
- ▶ Le SERP Rank est fonction du PageRank mais aussi de facteurs liés aux mots-clés.

Un petit essai

- ▶ Saisissons la requête suivante dans le moteur de recherche Google : « achat fleurs »
- ▶ Selon vous, le premier résultat affiché dans la page SERP possède-t-il le plus grand PageRank par rapport au autres résultats ? Pourquoi ?

Un petit essai

Google achat fleurs

Web Images Vidéos Plus Outils de recherche

Environ 5 480 000 résultats (0,16 secondes)

Acheter Bulbes De Fleurs - info.com
Annonce www.info.com/ Obtenez des infos de plusieurs moteurs de recherche!

Bulbes de fleurs, bulbes de printemps, achat/vente de ...
www.willemsefrance.fr/plantes/bulbes-a-fleurs.html Vente de bulbes de fleurs au meilleur prix chez Willemse, spécialiste de la vente de plantes et graines. Paiement sécurisé. Livraison rapide.

Jardinerie en ligne Willemse : achat/vente de plantes pour ...
www.willemsefrance.fr/ Vente en ligne de plantes, fleurs, graines & accessoires pour le jardin. Fiches conseils & astuces pour réussir votre jardinage. Demandez gratuitement le ... Arbres & arbustes - Plantes vivaces - Bulbes à fleurs pas cher - Rosiers

Livraison de bouquets de fleurs a domicile
www.aquarelle.com › Fleurs & Bouquets Aquarelle, leader de la livraison de fleurs, réalise et livre vos bouquets de fleurs à partir de 26,50 €. Commandez un bouquet de fleurs et recevez la photo de ...

Pourquoi?

3

4

4

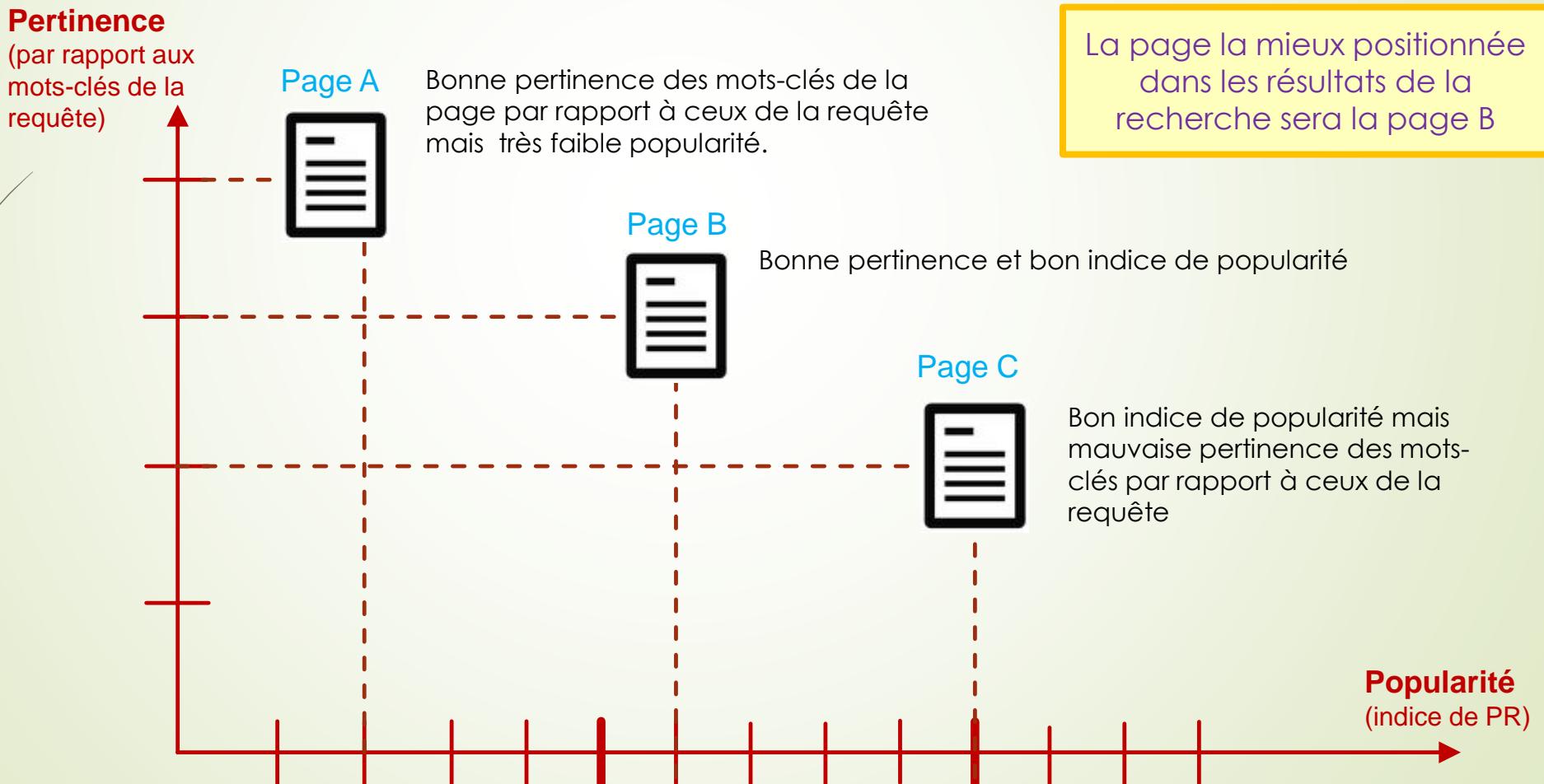
Référencement naturel

Le positionnement d'une page dépend de deux facteurs :

- ▶ **La popularité de la page** : c'est ce qu'on appelle pour Google, la « PageRank » d'une page. La PR permet de mesurer l'indice de popularité d'une page sur une échelle de 0 à 10.
- ▶ **La pertinence de la page** : c'est l'adéquation des différents mots-clés associés à la page lors de son indexation par le robot du moteur avec les mots-clés saisis dans la requête par l'internaute.

Ces deux facteurs sont complètement indépendant l'un de l'autre.

Positionnement de pages selon les deux critères



Optimisation d'une page

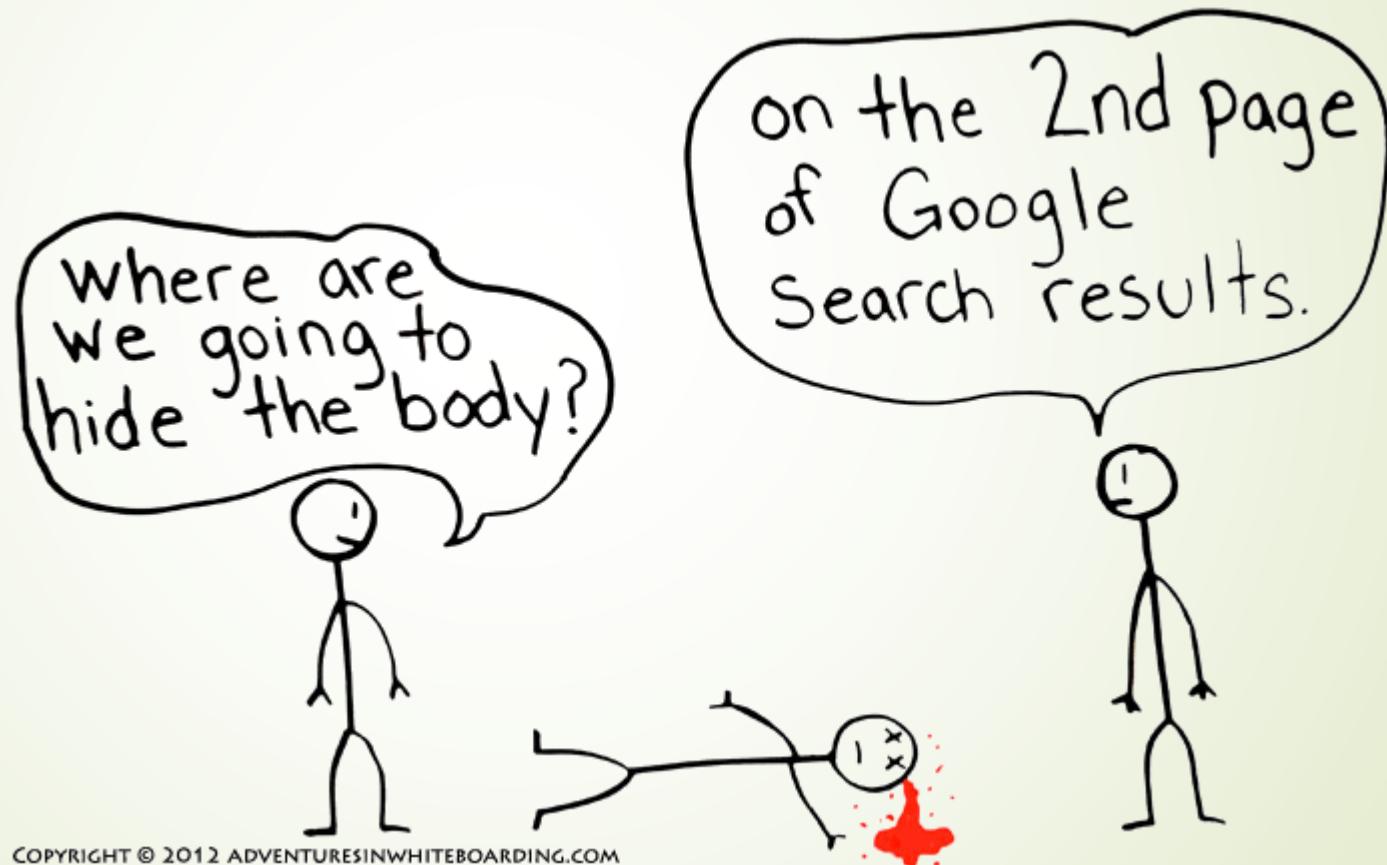
- ▶ Comment améliorer la pertinence et la popularité d'une page ?
- ▶ **PERTINENCE :**
 - ▶ Optimisation interne à chaque page (balises **title** et **meta**, structure et contenu de la page, images,...)
 - ▶ Optimisation globale du site (nom de domaine, nommage des URL, sitemap,...)
- ▶ **POPULARITE :**
 - ▶ Optimisation externe à chaque page (indice de popularité, stratégies de liens,...)

Préparation au référencement

80% des internautes ne dépassent pas la seconde page de résultats lors de leurs recherche sur un moteur

- L'objectif est donc d'être positionné dans ces deux premières pages si vous ne désirez pas vous partager les 20% d'internautes restants avec des milliers d'autres sites

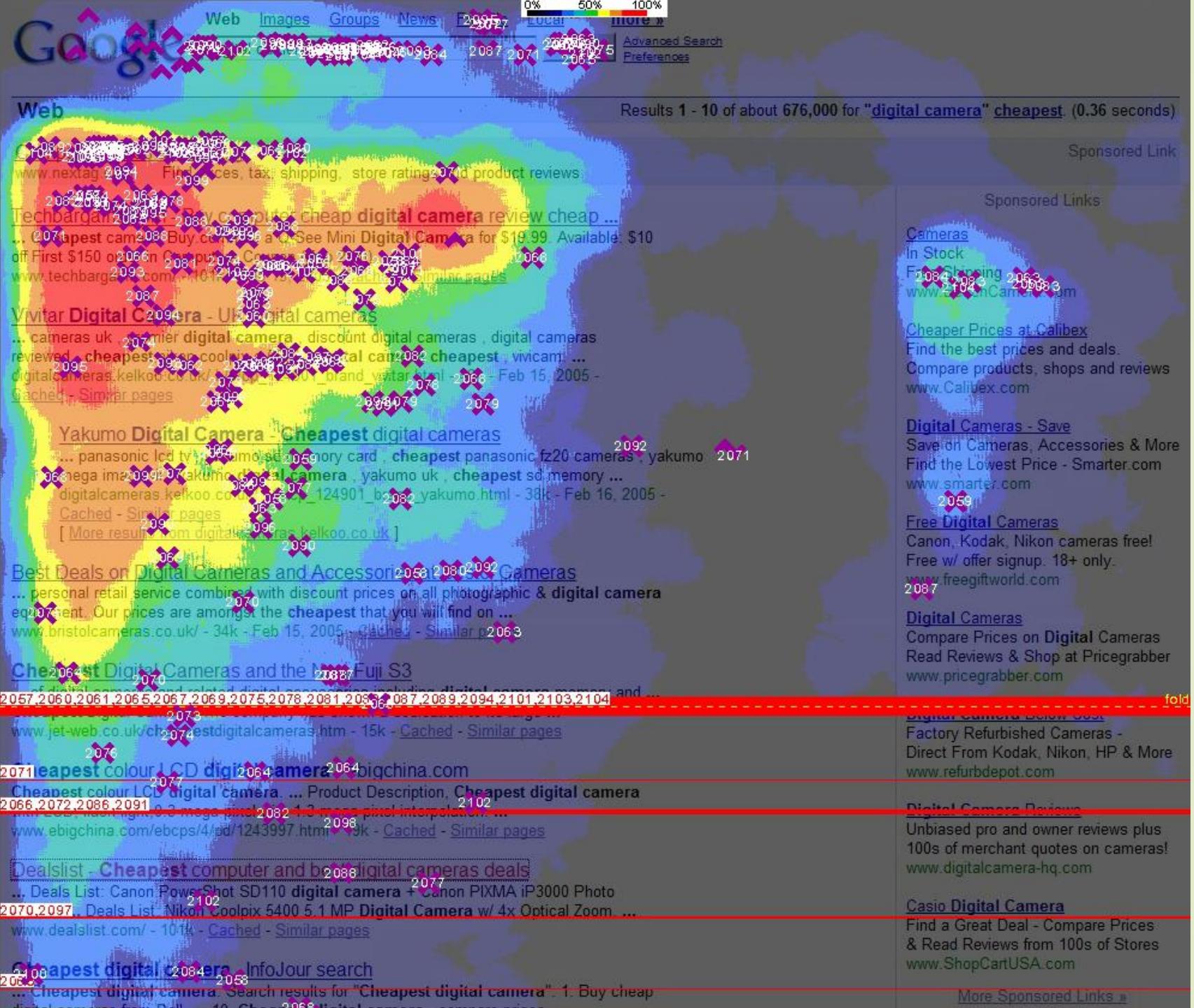
Préparation au référencement



Eye-tracking

- ▶ Selon une étude menée par les sociétés Enquiro et Dit-It.com en collaboration avec la société EyeTool, spécialisée dans les systèmes d'eye-tracking (analyse des mouvements de l'oeil), l'oeil de l'internaute explore en priorité un « triangle d'or », situé en haut à gauche des pages de résultats de Google.

Eye-tracking





Eye-tracking

- ▶ Ainsi, il est possible d'indiquer un taux de visibilité pour chaque rang des liens proposés par le moteur
 - ▶ positions 1, 2 et 3 : 100 % ;
 - ▶ position 4 : 85 % ;
 - ▶ position 5 : 60 % ;
 - ▶ positions 6 et 7 : 50 % ;
 - ▶ positions 8 et 9 : 30 % ;
 - ▶ position 10 : 20 %.



70% of the links search users click on are organic.

imFORZA.com



75% of users never scroll past the first page of search results.

imFORZA.com



70-80% of users ignore the paid ads, focusing on the organic results.

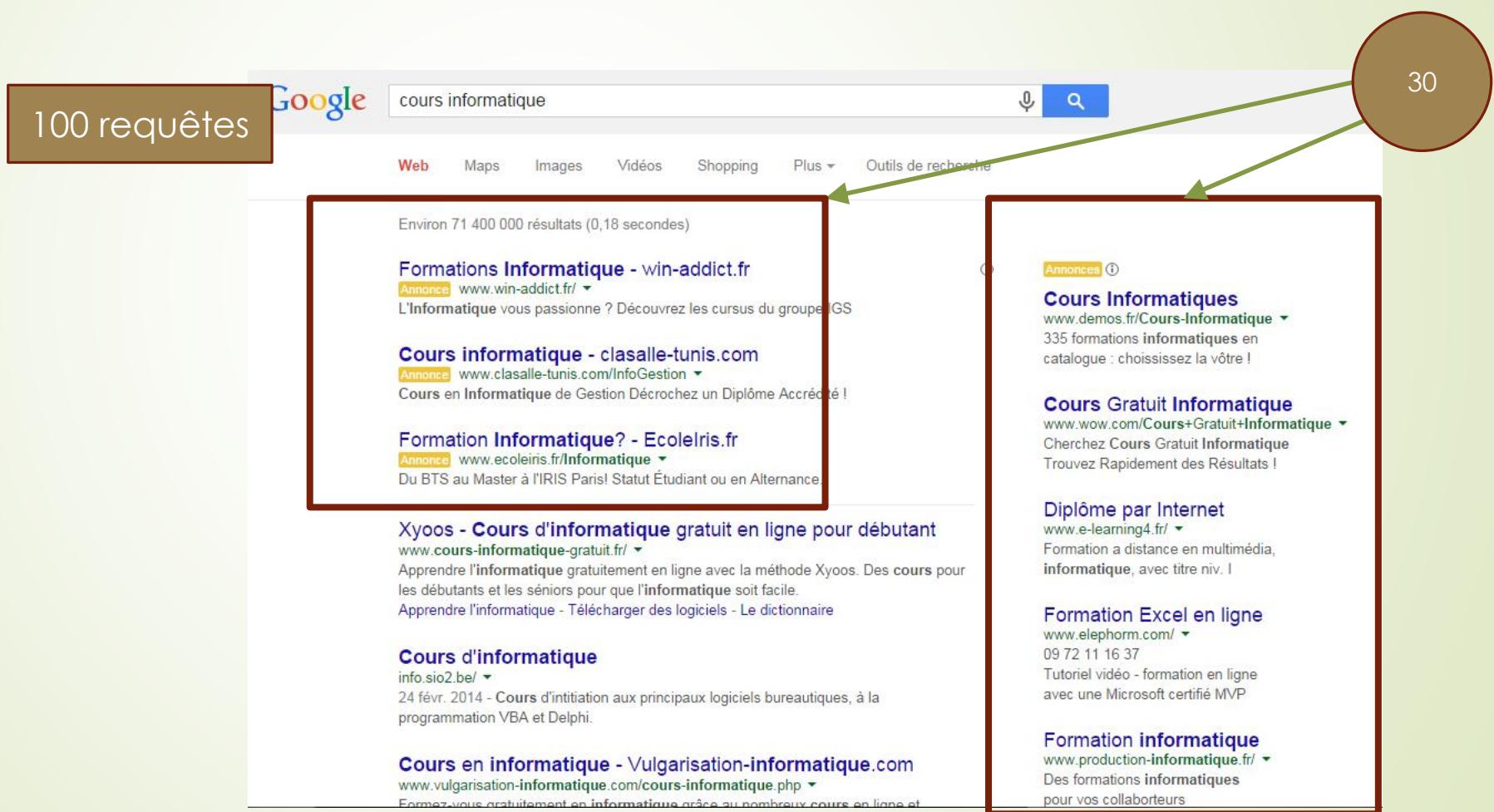
imFORZA.com



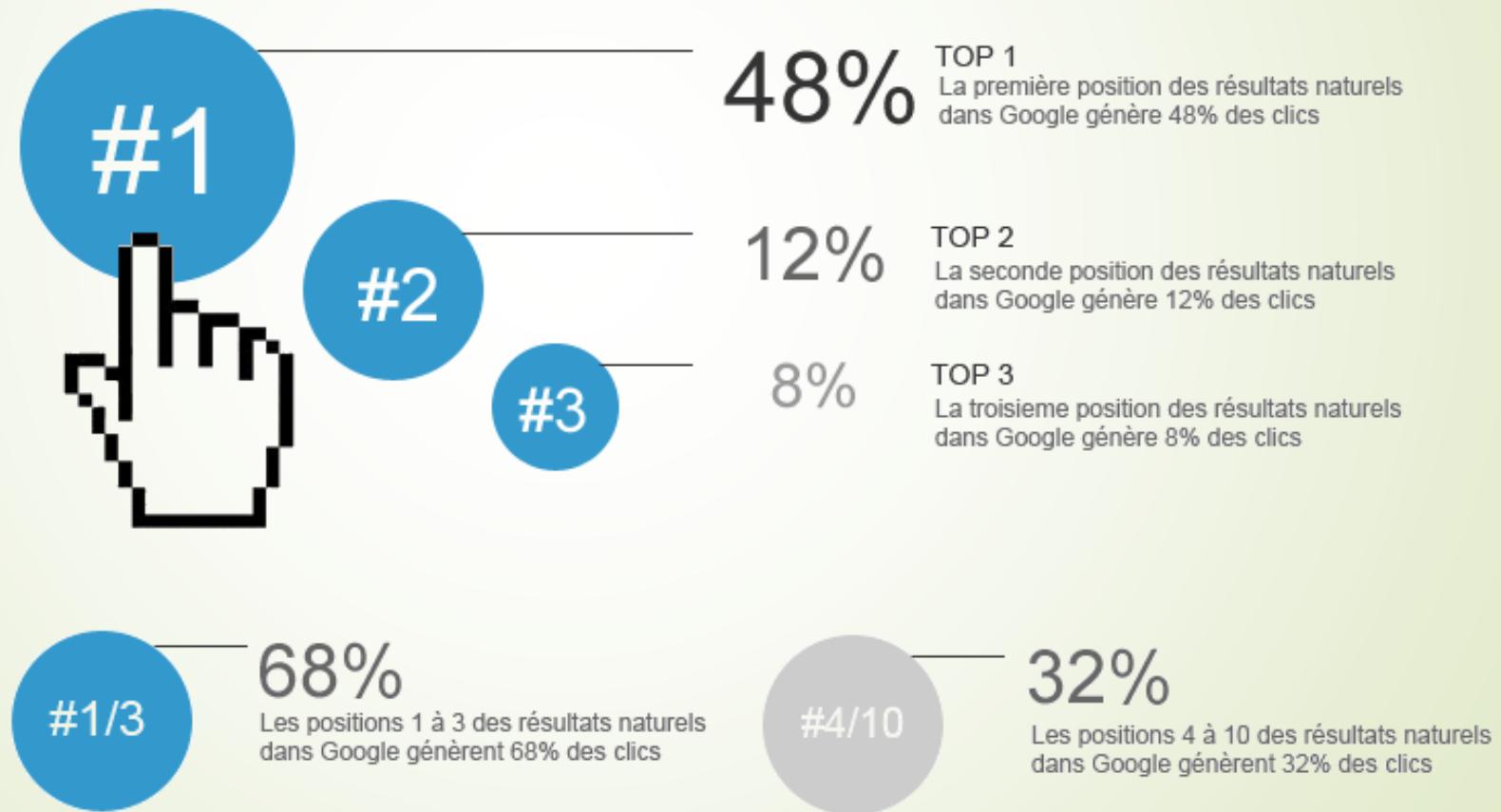
Search is the #1 driver of traffic to content sites, beating social media by more than 300%

imFORZA.com

Répartition des clicks sur la page SERP de Google



Répartition des clicks sur la page SERP de Google





Optimisation interne à chaque page

ON PAGE

Balise « title »

- ▶ Son contenu est repris en gras dans les résultats affichés par le moteur
- ▶ Son contenu s'affiche dans ma barre de titre du navigateur
- ▶ Son contenu est enregistré comme étiquette dans les favoris du navigateur
- ▶ Son contenu est repris dans l'historique du navigateur

Ordinateurs portables et tablettes professionnelles | Dell ...

www.dell.com/fr/entreprise/p/laptops ▾

À la recherche du meilleur ordinateur portable ou de la meilleure tablette pour entreprise? Optez pour un ordinateur portable ou une tablette professionnelle ...

- ▶ Son contenu est essentiel pour optimiser le positionnement d'une page car il est fortement pondéré lors de l'indexation de la page par le robot des moteurs

Balise « title »

TAILLE

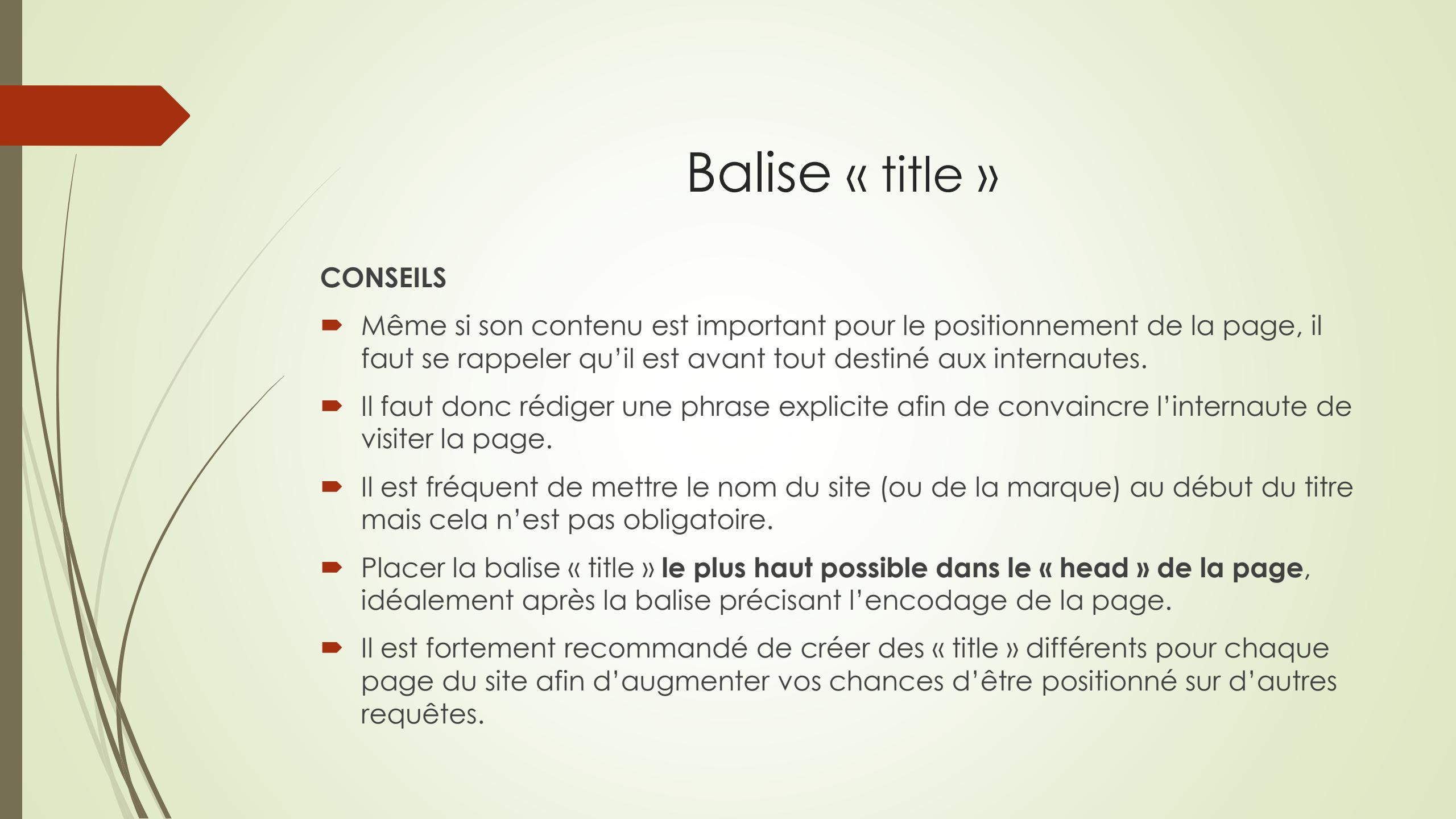
- ▶ 6 à 7 mots avec 60 caractères maximum (espaces inclus)

CONTENU

- ▶ Construire une phrase décrivant la page en y incluant les premiers mots les plus important de votre liste de mots-clés (le bloc A).
- ▶ Les mots placés au début auront plus d'importance que ceux à la fin.

EXEMPLE

```
<head>
    <meta charset="UTF-8" />
    <title>Ma boutique, PC portable Dell inspiron </title>
</head>
```



Balise « title »

CONSEILS

- ▶ Même si son contenu est important pour le positionnement de la page, il faut se rappeler qu'il est avant tout destiné aux internautes.
- ▶ Il faut donc rédiger une phrase explicite afin de convaincre l'internaute de visiter la page.
- ▶ Il est fréquent de mettre le nom du site (ou de la marque) au début du titre mais cela n'est pas obligatoire.
- ▶ Placer la balise « title » **le plus haut possible dans le « head » de la page**, idéalement après la balise précisant l'encodage de la page.
- ▶ Il est fortement recommandé de créer des « title » différents pour chaque page du site afin d'augmenter vos chances d'être positionné sur d'autres requêtes.

Balise « title »

ERREURS A EVITER

- ▶ Les symboles

» 9 [Tests de QI gratuits](#)
[www.mon-qI.com/tests-de-qI.php](http://www.mon-qi.com/tests-de-qI.php)

Exclusif : mesurez votre **QI**, quotient intellectuel, gratuitement grâce à nos **tests de QI**
... **Tests** très complets, Correction détaillée, Situation par rapport à la ...

- ▶ Ne pas mettre une balise title dans la page

[Untitled Document](#)
www.cgmp-provence.org/bagnards.html

Muni de ces renseignements, le chercheur pourra obtenir d'autres précisions en
consultant le **document** original au Service Historique de la Marine à Toulon, ...

- ▶ Utiliser des titres non optimisés de type « ma page d'accueil » ou bien
« bienvenu sur notre site web »

[bienvenue sur mon site](#)

lucasnico.monsite-orange.fr/

expo galerie de toiles et ardoises peinte et illustrations BD.

Balise « description »

- ▶ Son contenu doit décrire ce que contient la page et sera affiché en dessous du titre dans les résultats du moteur.
- ▶ A défaut de balise « description », le moteur choisira arbitrairement le texte placé au début du contenu de la page.
- ▶ Son contenu est de moins en moins utilisé dans le positionnement de la page.

[Ordinateurs portables et tablettes professionnelles | Dell ...](#)

www.dell.com/fr/entreprise/p/laptops ▾

À la recherche du meilleur ordinateur portable ou de la meilleure tablette pour entreprise? Optez pour un ordinateur portable ou une tablette professionnelle ...

- ▶ Même si les mots-clés qu'il contient sont faiblement pondérés lors de l'indexation de la page par le robot, il est fortement recommandé de toujours mettre une balise « description » pour contrôler le texte de la description qui s'affichera dans le résultats du moteur.

Balise « description »

TAILLE

- ▶ 15 à 25 mots avec 165 caractères maximum (espaces inclus)

CONTENU

- ▶ Construire une ou deux phrases décrivant la page en y incluant les mots-clés les plus importants de votre liste (bloc A). Ces mots devront aussi être cohérents avec les mots contenus dans le texte de la page afin de ne pas risquer d'être considéré comme spameur. La description doit être donc un développement du titre et un résumé du texte de la page.

EXEMPLE

```
<head>
    <meta charset="UTF-8"/>
    <title>Ma boutique, PC portable Dell inspiron </title>
    <meta name="description" content="À la recherche du meilleur ordinateur
portable ou de la meilleure tablette pour entreprise? Optez pour un ordinateur
portable ou une tablette"/>
</head>
```



Balise « description »

CONSEILS

- ▶ Même s'il est conseillé de faire un description spécifique à chaque page, il peut être toléré d'utiliser la même description pour plusieurs pages abordant le même sujet.
- ▶ Vous pouvez ainsi découper votre site en plusieurs parties thématiques afin de leurs associer une description commune.
- ▶ Comme pour le contenu du titre, rappelez-vous que le texte est avant tout destiné aux internautes.
- ▶ Il faut donc rédiger des phrases explicites afin de convaincre l'internaute de visiter la page.
- ▶ Evitez de répéter les mêmes mots-clés de manière trop rapprochée dans le texte des phrases.
- ▶ Si un mot-clé est répété, essayez d'en espacer le plus possibles les occurrences.



Balise « keywords »

- ▶ Son contenu n'est **plus utilisé par** les moteurs de recherche **Google** et **Bing**
- ▶ **Seul Yahoo!** semble lui attacher un peu d'importance (avant son accord avec Bing)
- ▶ Plus de détails : <http://bit.ly/1FR4CKZ> / <http://bit.ly/1vRk5bo>
- ▶ Suite au nombreux abus dans l'usage de cette balise par les webmasters pour surclasser leur site, les principaux moteurs de recherche actuels ne tiennent plus compte des mots-clés qu'elle contient pour positionner une page dans les résultats.

Balise « keywords »

TAILLE

- 20 à 100 mots avec 1000 caractères maximum (espaces inclus)

CONTENU

- Saisissez tous les groupes de mots-clés de votre liste en séparant chaque groupe par une virgule (utilisez tous les blocs de A à E)

EXEMPLE

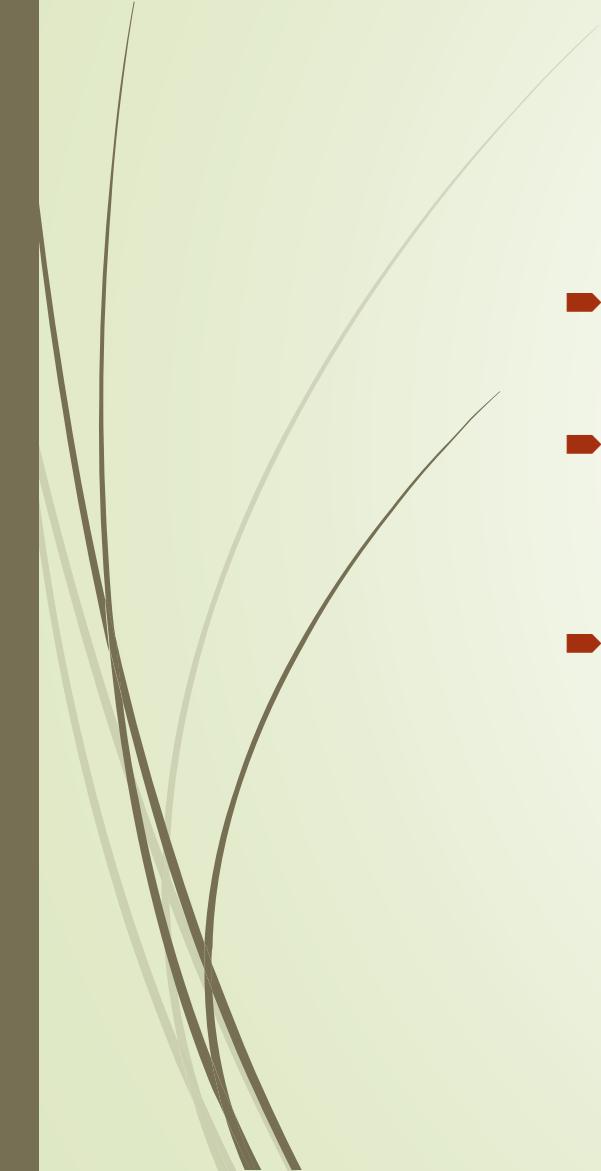
```
<head>
    <meta charset="UTF-8"/>
    <title>Ma boutique, PC portable Dell inspiron </title>
<meta name="description" content="À la recherche du meilleur
ordinateur portable ou de la meilleure ..."/>
<meta name="keywords" content="pc portable dell, ordinateur dell,
dell inspiron prix bas, ordinator portable dell,..."/>
</head>
```



Balise « keywords »

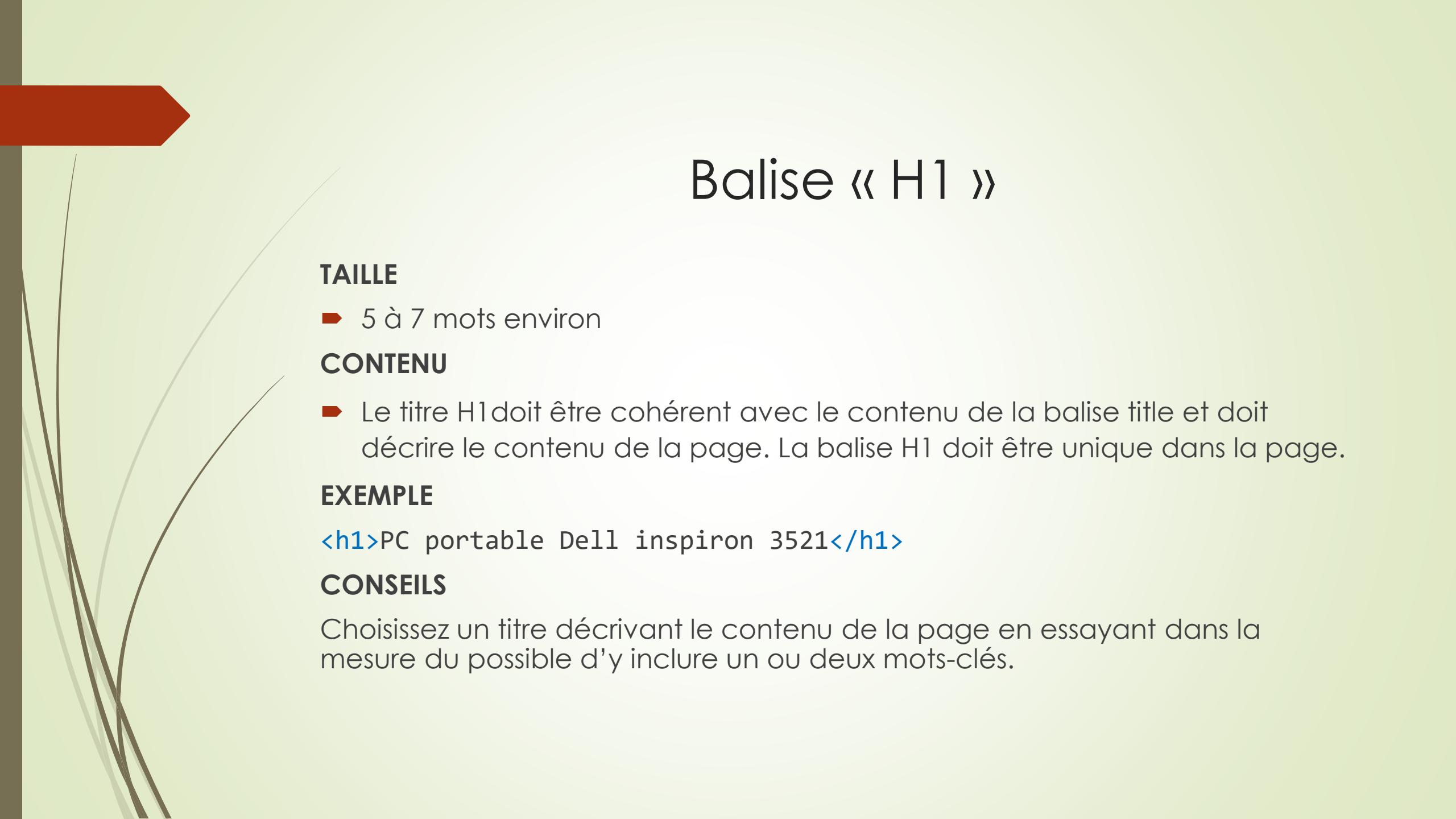
CONSEILS

- ▶ Proposez plusieurs déclinaisons des groupes de mots-clés en variant la casse, l'accentuation et les mettant aussi au pluriel.
- ▶ Vous pouvez aussi ajouter les principales fautes d'orthographe que les internautes peuvent faire en saisissant les mots-clés dans la balise.
- ▶ Prenez soin de bien mettre les groupes de forte priorité en début de liste dans la balise.



Balise « H1 »

- ▶ Son contenu est **fortement pondéré par les moteurs** de recherche car il est sensé représenter le titre principal de la page.
- ▶ Comme le contenu de la balise H1 est obligatoirement visible par les internautes, il doit toujours être cohérent avec le reste du contenu de la page.
- ▶ Il y a donc peu de risque de spam par les webmasters et il est à ce titre bien considéré par les moteurs de recherche.



Balise « H1 »

TAILLE

- ▶ 5 à 7 mots environ

CONTENU

- ▶ Le titre H1 doit être cohérent avec le contenu de la balise title et doit décrire le contenu de la page. La balise H1 doit être unique dans la page.

EXEMPLE

`<h1>PC portable Dell inspiron 3521</h1>`

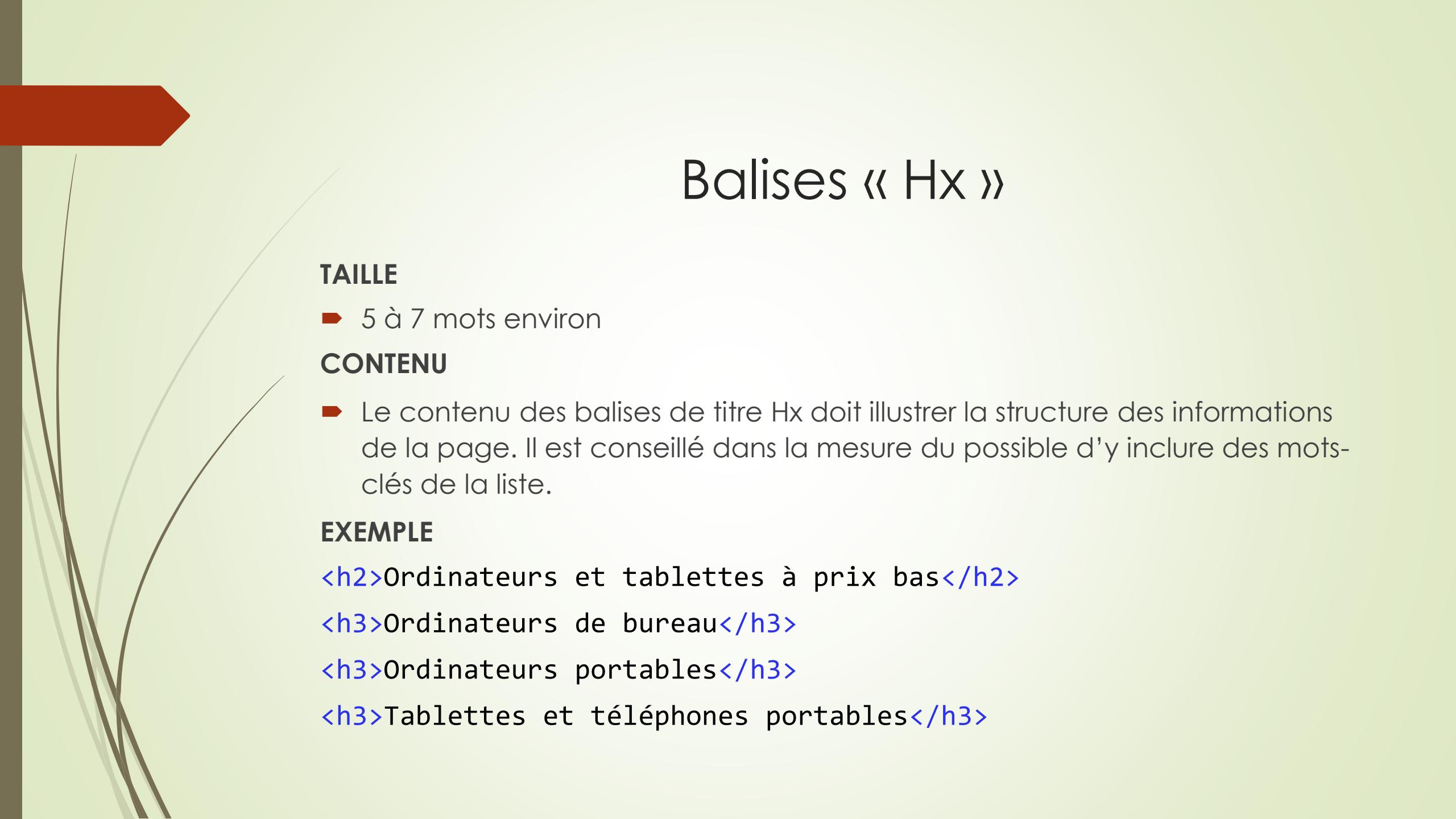
CONSEILS

Choisissez un titre décrivant le contenu de la page en essayant dans la mesure du possible d'y inclure un ou deux mots-clés.



Balises « Hx »

- ▶ Chaque contenu de balise de titre sera pondéré en rapport avec le niveau du titre (H1, H2,...).
- ▶ Contrairement à la **balise H1 qui doit être unique**, les autres balises de titre peuvent être répétées plusieurs fois dans la page selon la structure des informations à afficher.
- ▶ Le fait de pouvoir utiliser plusieurs balises de même niveau dans la structure du contenu de la page, permet d'y placer facilement des mots-clés importants par rapport à l'objectif de positionnement de la page.



Balises « Hx »

TAILLE

- ▶ 5 à 7 mots environ

CONTENU

- ▶ Le contenu des balises de titre Hx doit illustrer la structure des informations de la page. Il est conseillé dans la mesure du possible d'y inclure des mots-clés de la liste.

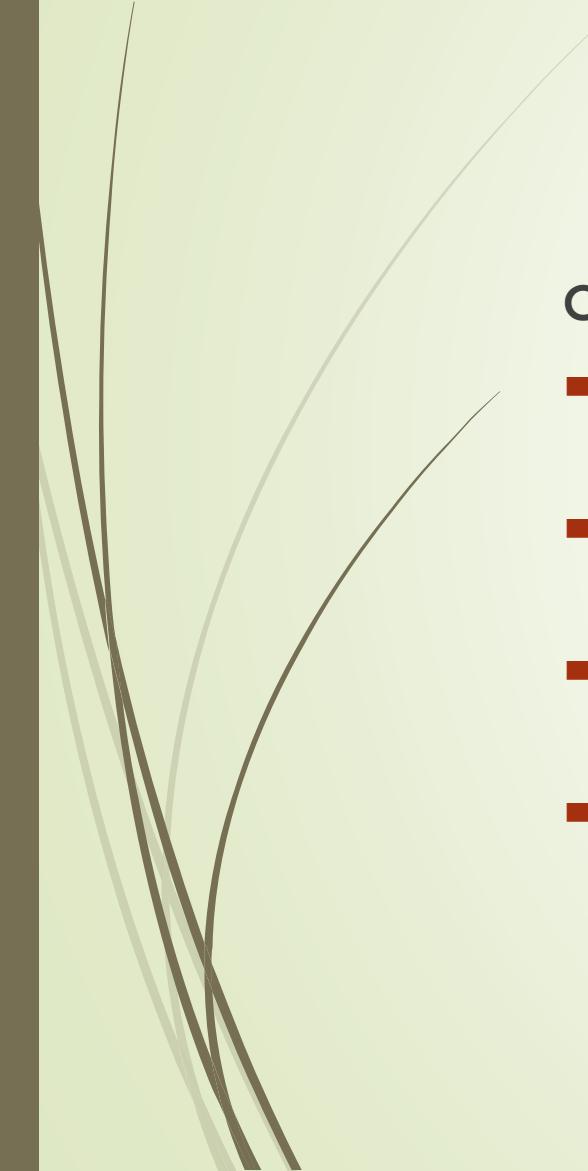
EXEMPLE

`<h2>Ordinateurs et tablettes à prix bas</h2>`

`<h3>Ordinateurs de bureau</h3>`

`<h3>Ordinateurs portables</h3>`

`<h3>Tablettes et téléphones portables</h3>`



Balises « Hx »

CONSEILS

- ▶ Le contenu des balises de titre Hx doit illustrer la structure des informations de la page.
- ▶ Il est conseillé dans la mesure du possible d'y inclure des mots-clés de la liste.
- ▶ Il est également conseillé d'avoir au moins trois niveaux de balises de titre (de H1 à H3).
- ▶ Vous pouvez redéfinir si besoin la mise en forme des balises de titre à l'aide de feuille de styles en rapport avec la charte graphique du site.



Les images

- ▶ L'attribut **ALT** (texte alternatif de l'image) de la balise image est lu par les moteurs de recherche et peut donc permettre d'améliorer le positionnement de la page contenant l'image si des mots-clés sont placés dans cet attribut.
- ▶ De même, le nom du fichier de l'image apparaissant aussi dans le code, doit aussi être nommé en fonction de ce que représente l'image.
- ▶ Si votre page comporte des images, prenez soin de bien configurer la balise ALT de chaque image et de nommer le fichier d'image avec un texte indiquant ce que représente l'image et comportant si possible des mots-clés issus de la liste.

Les images

TAILLE

- ▶ 2 à 5 mots environ

CONTENU

- ▶ La valeur de l'attribut ALT doit contenir un texte alternatif à l'image dans le cas où celle-ci ne s'affiche pas. Il est conseillé dans la mesure du possible d'y inclure aussi des mots-clés issus de la liste.

EXEMPLE

```

```



Les images

CONSEILS

- ▶ L'attribut ALT est aussi utilisé pour l'accessibilité et notamment les non-voyants qui naviguent à l'aide du lecteur d'écran dans la page web.
- ▶ Il faut donc que le texte de cet attribut reste cohérent avec ce que représente l'image.



Les balises de mise en forme

Les balises: `` / `` / ``

- ▶ Les balises `` et `` sont également d'une importance capitale dans votre code HTML, simplement parce qu'elles permettent de mettre en valeur des expressions que vous jugez importantes, et qui constituent par extension l'intérêt de votre page web.
- ▶ Utilisez ces balises avec modération, et ne mettez pas en valeur 95 % de vos textes avec celles-ci.
- ▶ **Googlebot** est très perfectionné : s'il voit une balise ``, il va considérer l'information qu'elle contient comme importante.
- ▶ Ne faites donc pas n'importe quoi avec ces balises.

Optimisation du contenu

THE CONTENT IS KING

- ▶ Le contenu c'est le texte visible pour les visiteurs de la page.
- ▶ Mais il englobe aussi le contenu média (images, vidéos), la facilité de navigation, l'ergonomie du site,...

Symptômes d'une faiblesse de contenu :

- ▶ Taux de rebond très élevé,
- ▶ Taux des nouveaux visiteurs élevé,
- ▶ Peu d'entrée directe (voir les trois piliers de trafic)

A éviter !!!

LE CLOAKING

- ▶ Texte caché visible pour le robot indexeur, mais invisible par le lecteur
- ▶ Techniques utilisées : Meta tags, identification des images, texte blanc sur fond blanc, caché derrière une image, display:none, etc.
- ▶ Les robots détectent de plus en plus les données cachées
- ▶ Dans cas les robots baissent la TrustRank des pages concernées.



Optimisation externe à chaque page

OFFSITE

L'indice de popularité

- ▶ L'indice de popularité (nommé **PageRank** pour Google) permet d'évaluer le nombre et la qualité des liens qui pointent sur la page (backlinks).
- ▶ Il est noté de 0 à 10 selon une échelle logarithmique.
- ▶ L'indice de popularité associé à l'indice de pertinence de la page permettra d'améliorer le positionnement de la page dans les résultats du moteur.

CONSEILS

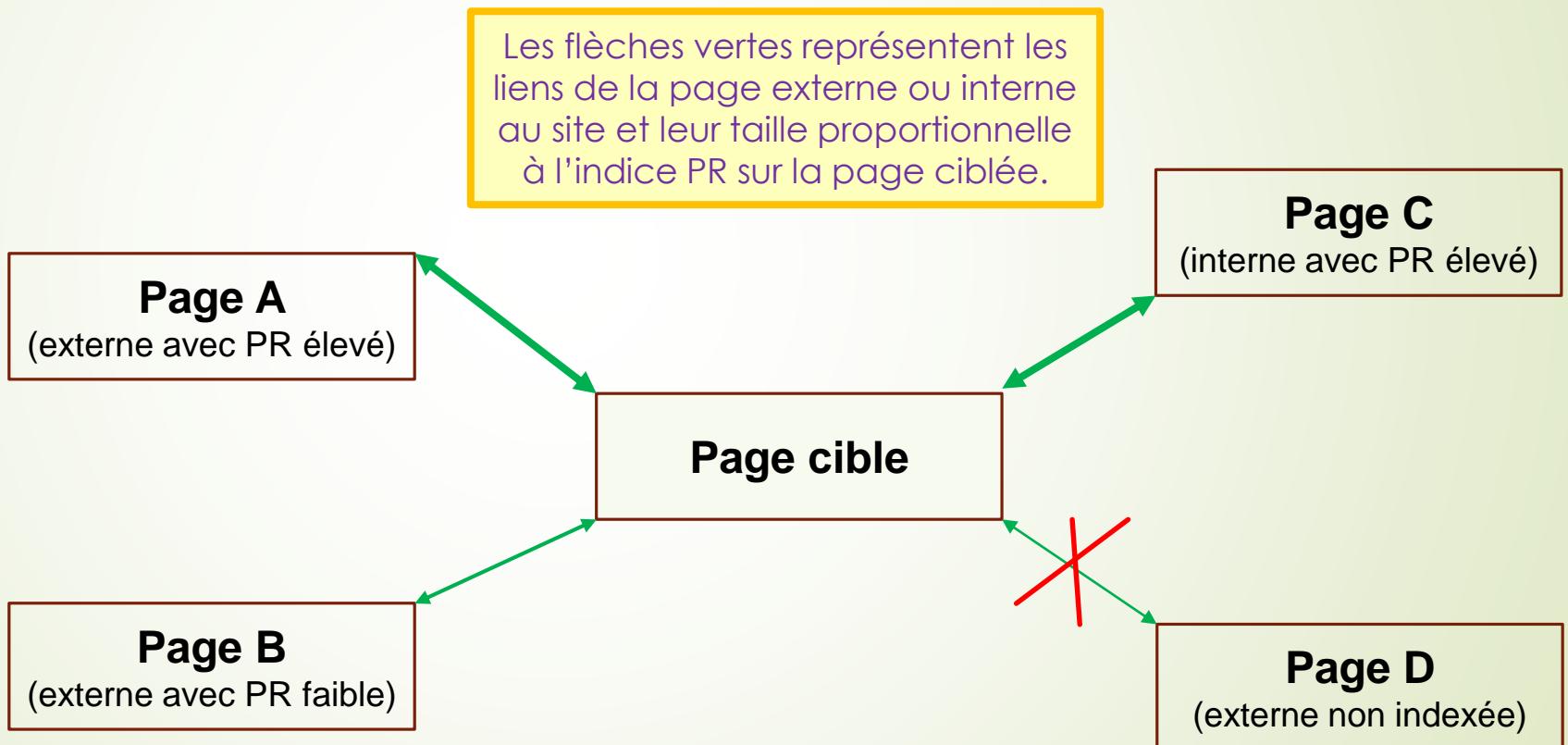
- ▶ Une page est la cible des liens hypertextes, plus son indice de popularité sera important.
- ▶ Les liens externes au site seront mieux pondérés que les liens inter-pages d'un même site.

L'indice de popularité

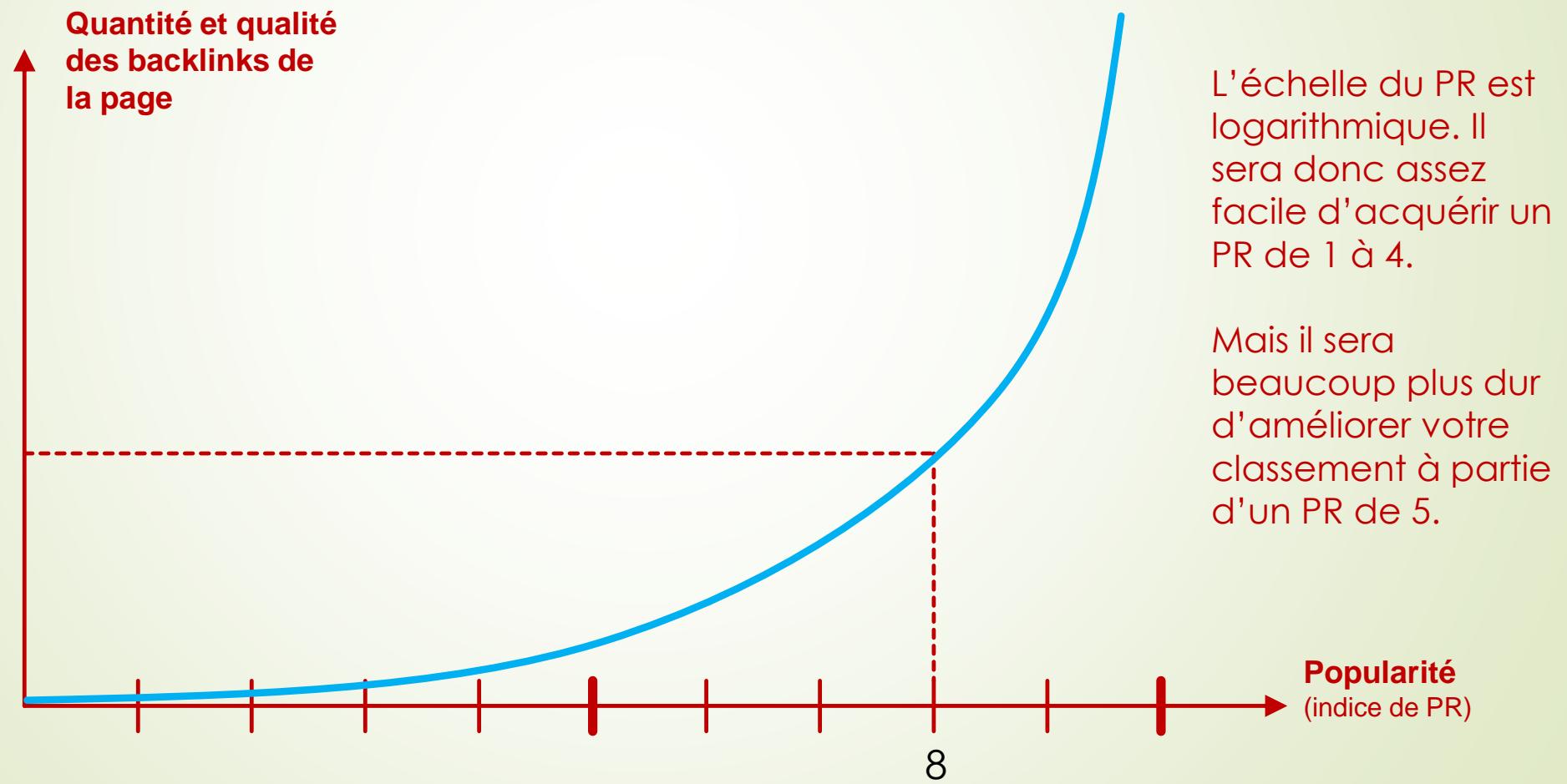
CONSEILS

- ▶ Les liens issus d'une page ayant un bon PR seront aussi mieux pondérés.
- ▶ Plus une page aura des liens sortant, moins l'indice de chaque lien aura d'importance pour l'indice de popularité de la page cible.
- ▶ Pour qu'un lien issu d'une page soit pris en compte, elle doit être indexée dans le moteur concerné.
- ▶ Le texte des liens devra contenir des mots-clés en rapport avec le contenu de la page ciblée (évitez « cliquez ici » par exemple).
- ▶ Les liens devront être réalisés en HTML (à bannir les liens en JavaScript ou en flash)

L'indice de popularité



L'indice de popularité



L'indice de popularité

Sur Google

- ▶ Afficher les backlinks d'une page (Google, Bing)
link:<http://www.monsite.com/mapage.html>
- ▶ L'extension « Google ToolBar » permet d'afficher le PR d'une page (Firefox et Internet Explorer)
- ▶ Sur Google chrome <http://bit.ly/1pwMrYK>

Comment augmenter son indice de popularité?

- ▶ En réalisant des échanges de liens avec des sites ayant un fort PR,
- ▶ En s'inscrivant sur les annuaires (généralistes ou thématiques),
- ▶ En diffusant du contenu de qualité sur d'autres sites.



Optimisation globale du site

ONSITE



Le nom de domaine

- ▶ Inclure un mot clé dans le nom de domaine favorise son positionnement dans les résultats pour les requêtes contenant ce mot clé.
- ▶ Il n'est pas toujours facile d'agir sur le nom de domaine car ce dernier doit d'abord satisfaire les exigences du marketing.
- ▶ Néanmoins il est possible d'y inclure un mot clé, cela ne pourra être que positif pour votre visibilité sur Internet.
- ▶ Si le nom de domaine est constitué de deux mots, utilisez alors un tiret pour les séparer (exemple **mon-site.com**).
- ▶ Utilisez un seul nom de domaine pour votre site.
- ▶ Si vous avez un second domaine, faites une redirection permanente (ou **redirection 301**) vers le domaine principal.

Les URL

- ▶ Inclure un mot clé dans le nom des répertoires contenant les pages web favorise leur positionnement dans les résultats pour les requêtes contenant ce mot-clé.

EXEMPLE

- ▶ Moins efficace :
 - ▶ <http://mon-site.com/produit.php?idproduit=213&categorie=11>
- ▶ Efficace :
 - ▶ <http://mon-site.com/disque-dur-externe-213.html>
- ▶ Encore plus efficace :
 - ▶ <http://mon-site.com/accessoires/disque-dur-externe-213.html>



Le sitemap

- ▶ La présence d'un sitemap à la racine **facilite l'indexation** de votre site.
- ▶ Le sitemap est un **fichier XML** qui représente l'architecture du site.
- ▶ La présence d'un fichier sitemap.xml sur votre site n'augmente pas son positionnement mais favorisera le travail du robot.
- ▶ Il permet aussi de lui communiquer la structure des pages que le robot devra indexer en précisant l'importance de chaque page, la date de la dernière actualisation, et la fréquence de ses mises à jour.

Le sitemap

CONSEILS

- ▶ Actualisez le fichier sitemap.xml à chaque changement de la structure du site.
- ▶ Si le site est dynamique, il est alors conseillé de générer automatiquement un fichier sitemap à chaque ajout de données dans la base.
- ▶ Il existe de nombreux logiciels pour générer un sitemap.
- ▶ Certains d'entre eux peuvent être utilisés en ligne depuis un simple navigateur : <http://www.xml-sitemaps.com>
- ▶ Pour déclarer un sitemap à Google, vous devez créer un compte de webmaster sur le site de Google puis le déclarer en ligne dans votre espace privé.

Le sitemap

CONSEILS

- ▶ Pour déclarer automatiquement un sitemap pour tous les moteurs (Google, Bing et Yahoo), il faut ajouter une instruction qui indique la localisation du fichier dans le fichier robots.txt :

sitemap: <http://mon-site.tn/sitemap.xml>
- ▶ Il est aussi judicieux de créer une version HTML du sitemap pour la mettre à la disposition des internautes.



Le fichier robots.txt

- ▶ Le fichier **robots.txt** ne contribue pas au référencement mais au contraire, il permet de bloquer l'indexation de certaines pages du site.
- ▶ Si vous ne souhaitez pas que certaines pages du site se retrouvent dans les résultats des moteurs car elles sont confidentielles ou simplement en construction, vous pouvez alors configurer le fichier robots.txt en conséquence.
- ▶ Comme nous l'avons vu déjà précédemment, vous pouvez aussi ajouter dans ce fichier une instruction qui indiquera aux robots où se trouve le fichier sitemap.

Le fichier robots.txt

- ▶ **User-agent : ***

- ▶ La première ligne permet d'indiquer à quel robot sont destinées les instructions qui suivent. Mettre une étoile pour les destiner à tous les robots.

- ▶ **Disallow : /private/**

- ▶ La seconde ligne désigne le répertoire ou la page qui ne doit pas être visitée par le robot.

- ▶ **Allow : /public/**

- ▶ Cette ligne (utilisée dans des cas particulier) désigne le répertoire ou la page qui doit être visitée par le robot.

- ▶ **Sitemap : <http://mon-site.com/sitemap.xml>**

- ▶ La dernière ligne doit être ajoutée si vous avez un fichier sitemap.xml afin de préciser au robot où il se trouve

Le fichier robots.txt

CONTENU

```
user-agent : *
Disallow : /admin/
Disallow : /fichier-prive.html
Sitemap : http://mon-site.com/sitemap.xml
```

CONSEILS

- ▶ Le fichier robots.txt est un fichier texte (ascii) qui doit être placé à la racine du site (attention de bien l'écrire en minuscule et avec un s).
- ▶ Ne laissez aucune ligne vide dans le fichier.
- ▶ Vous pouvez dupliquer la seconde instruction autant de fois que vous aurez de répertoires ou de pages spécifiques à interdire.
 - ▶ Pour exclure toutes les pages du site, utilisez l'instruction **disallow** : /
 - ▶ Pour n'exclure aucune page, utilisez l'inscruction **disallow** : (sans paramètre)



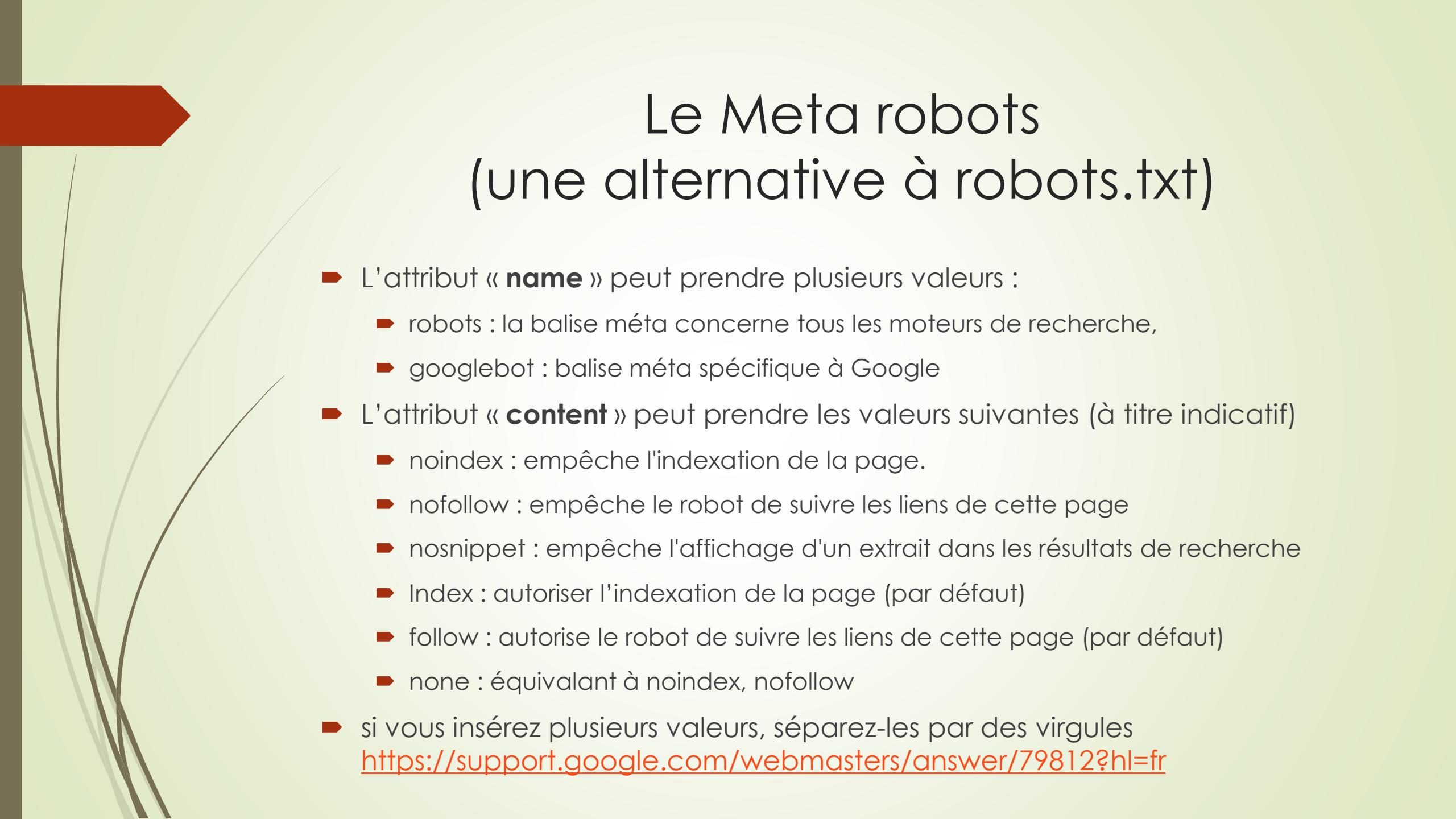
Le Meta robots (une alternative à robots.txt)

- ▶ Il existe une autre méthode que le fichier robots.txt pour interdire l'indexation des vos pages par le robot des moteurs. Il faut utiliser la **balise meta robots** possédant le forme suivante :

```
<meta name="robots" content="index,follow" />
```

- ▶ Cette balise interdira aux robots l'indexation de votre page. Tout comme robots.txt, il est possible d'interdire l'accès à quelques robots uniquement. La balise meta robots s'écrira de cette façon :

```
<meta name="nom du robot" content="action,..." />
```



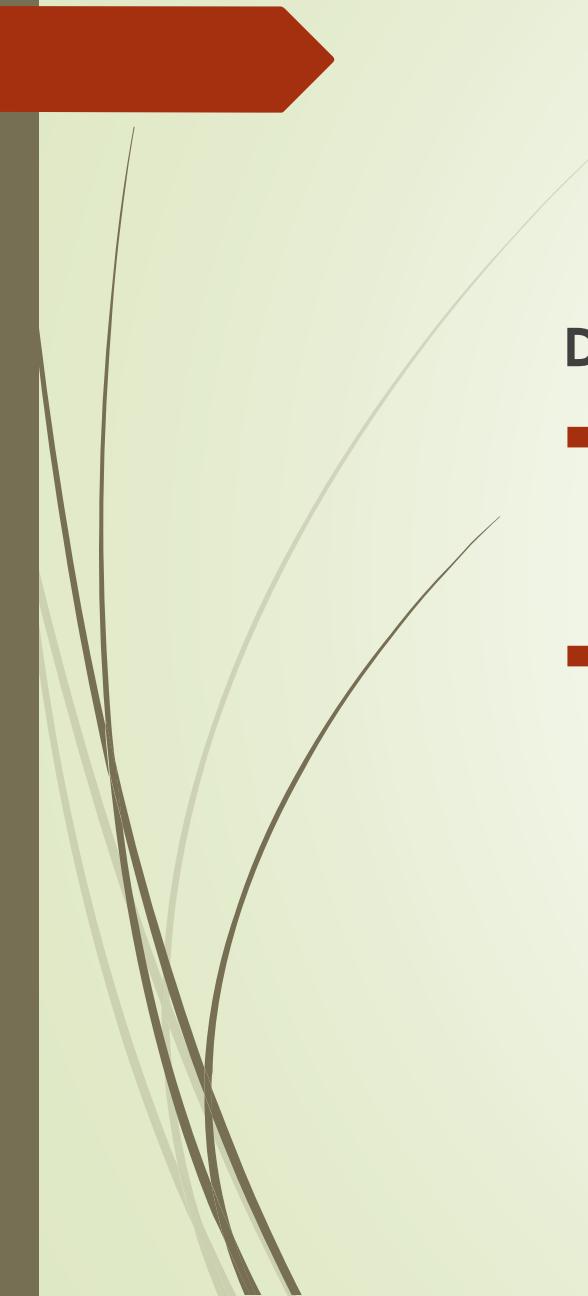
Le Meta robots (une alternative à robots.txt)

- ▶ L'attribut « **name** » peut prendre plusieurs valeurs :
 - ▶ robots : la balise méta concerne tous les moteurs de recherche,
 - ▶ googlebot : balise méta spécifique à Google
- ▶ L'attribut « **content** » peut prendre les valeurs suivantes (à titre indicatif)
 - ▶ noindex : empêche l'indexation de la page.
 - ▶nofollow : empêche le robot de suivre les liens de cette page
 - ▶ nosnippet : empêche l'affichage d'un extrait dans les résultats de recherche
 - ▶ Index : autoriser l'indexation de la page (par défaut)
 - ▶ follow : autorise le robot de suivre les liens de cette page (par défaut)
 - ▶ none : équivalent à noindex,nofollow
- ▶ si vous insérez plusieurs valeurs, séparez-les par des virgules
<https://support.google.com/webmasters/answer/79812?hl=fr>



Le contenu dupliqué (duplicate content)

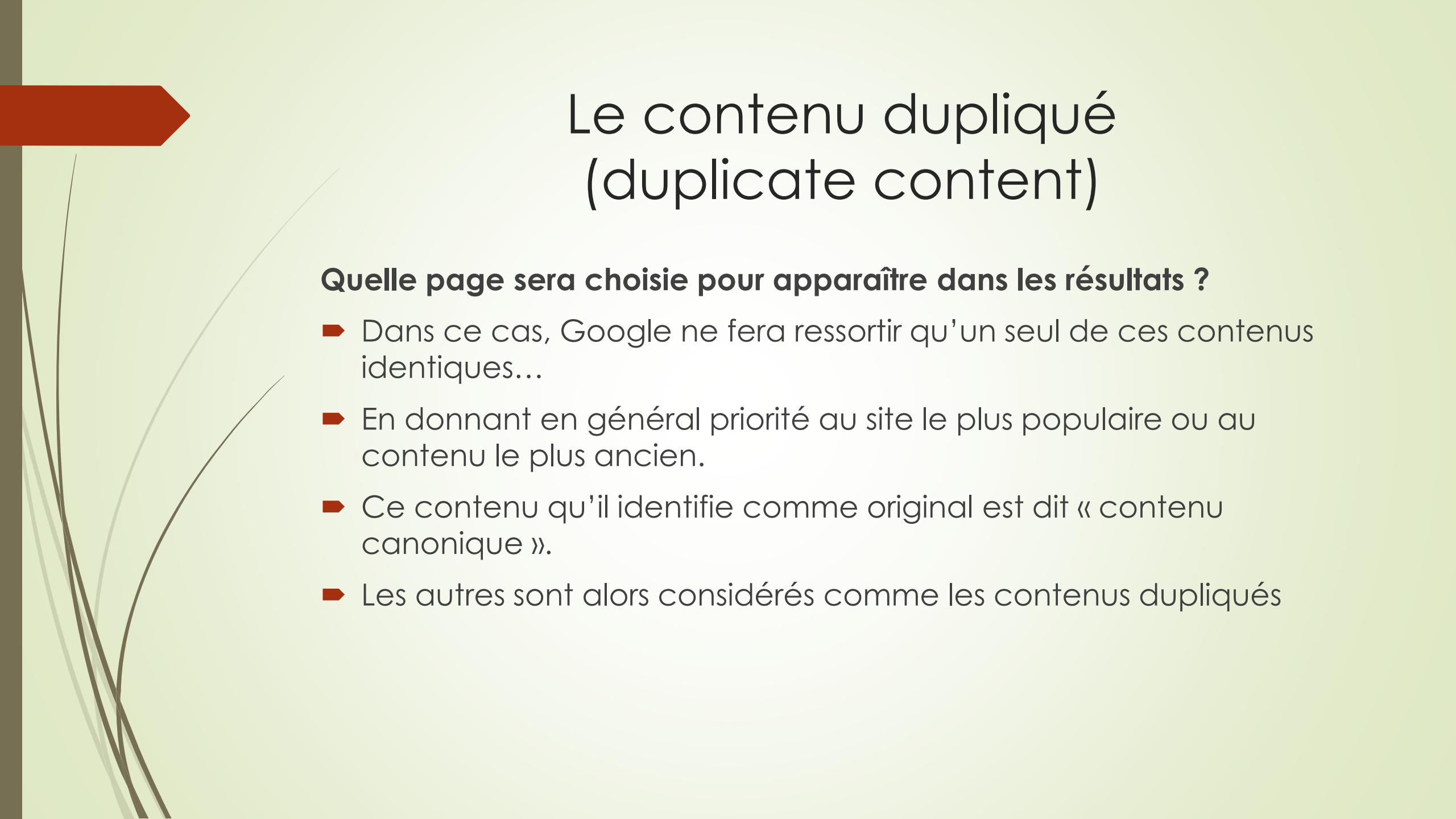
- ▶ Le problème des contenus dupliqués (**duplicate content**) est un des sujets les plus récurrents dans le domaine du référencement.
- ▶ Le contenu dupliqué est quelque chose de naturel, ne serait-ce que par le fait que nous sommes amené à faire des citations, à rapporter des propos de personnalités ou encore à faire mention à des textes officiels.
- ▶ Toutefois, une trop grande proportion de contenu dupliqué sur un site **peut conduire à une pénalité algorithmique**, il est donc conseillé de bloquer un tel contenu à l'aide d'un fichier robots.txt ou une balise meta robots ayant pour valeur « noindex »



Le contenu dupliqué (duplicate content)

Duplicate content externe

- ▶ Il arrive parfois que des sites e-commerce concurrents possèdent de nombreuses pages quasi identiques, à cause de la fameuse question du « descriptif fabriquant ».
- ▶ Si les pages descriptives des produits manufacturés que vous vendez (chaussures, vêtements, ...) se bornent à reprendre la description inscrite sur la notice ou sur l'étiquette, il y a fort à parier que certains de vos concurrents auront fait la même erreur et que le contenu de vos pages sera donc exactement le même que celui de plusieurs autres sites.



Le contenu dupliqué (duplicate content)

Quelle page sera choisie pour apparaître dans les résultats ?

- ▶ Dans ce cas, Google ne fera ressortir qu'un seul de ces contenus identiques...
- ▶ En donnant en général priorité au site le plus populaire ou au contenu le plus ancien.
- ▶ Ce contenu qu'il identifie comme original est dit « contenu canonique ».
- ▶ Les autres sont alors considérés comme les contenus dupliqués



Le contenu dupliqué (duplicate content)

Duplicate content interne

- ▶ **DUST : Duplicate URL Same Text**
- ▶ Plusieurs URL qui pointent sur la même page (donc même contenu).

DUST

Des URL dynamiques existent pour la même page relative à un produit et proviennent de sessions utilisateurs et/ou de préférences de recherche.	<code>http://www.example.com/produits?catégorie=robes&couleur=vert</code> <code>http://example.com/robes/cocktail?gclid=ABCD</code> <code>http://www.example.com/robes/vert/robeverte.html</code>
Votre système de blog enregistre automatiquement plusieurs URL si vous placez le même article sous plusieurs sections.	<code>http://blog.example.com/robes/les-robdes-vertes-sont-magnifiques/</code> <code>http://blog.example.com/chose-vertes/les-robdes-vertes-sont-magnifiques/</code>
Votre serveur est configuré pour diffuser le même contenu pour le sous-domaine www et le protocole https .	<code>http://example.com/robdes-vertes</code> <code>https://example.com/robdes-vertes</code> <code>http://www.example.com/robdes-vertes</code>
Le contenu de ce blog destiné à la syndication sur d'autres sites est répliqué partiellement ou intégralement sur ces domaines.	<code>http://news.example.com/robdes-vertes-pour-tous-les-jours-155672.html</code> (article syndiqué) <code>http://blog.example.com/robes/les-robdes-vertes-sont-magnifiques/3245/</code> (article d'origine)

Source : <https://support.google.com/webmasters/answer/139066?hl=fr>

Duplicate content (DUST) : Solution

Définissez votre domaine favori :

- ▶ Indiquez quelle version de l'URL de votre site vous préférez pour votre domaine :
 - ▶ <http://www.mon-site.com>
 - ▶ <http://mon-site.com>
- ▶ Vous pouvez indiquer votre URL favori soit dans le fichier de configuration de votre serveur, soit en plaçant un fichier « .htaccess » sur le racine de votre site :

```
Rewritecond %{HTTP_HOST} ^mon-site.com$
```

```
Rewriterule ^(.*) http://www.mon-site.com/$1 [QSA,L,R=301]
```

Duplicate content (DUST) : Solution

Utilisez le **rel="canonical"**

- ▶ Prenons l'exemple suivant :
 - ▶ Une page A considérée comme page principale que Googlebot doit traiter en priorité,
 - ▶ Une page B au contenu identique à la page A, nous souhaitons que Googlebot ne se focalise pas dessus.
- ▶ Nous ajoutons dans le « head » de la page B la ligne suivante :

```
<link rel="canonical" href="http://www.mon-site.com/a.html" />
```

Duplicate content (DUST) : Solution

Utilisez le rel="canonical"

- ▶ La page détail d'un produit X est accessible par le biais de deux liens
 - ▶ <http://www.mon-site.com/details.php?cat=11&produit=123>
 - ▶ <http://www.mon-site.com/categorie/mon-produit-123.html>
- ▶ Nous ajoutons dans le « head » de la page détail le lien préféré (que nous souhaitons afficher aux internautes) :

```
<link rel="canonical" href="http://www.mon-  
site/com/categorie/mon-produit-123.html" />
```

Duplicate content (DUST) : Solution

Utilisez le rel="canonical"

- ▶ D'une manière générale, il est conseillé de faire figurer dans vos pages une balise canonical avec l'URL de la page en cours.
- ▶ Cela permet notamment de limiter la déperdition liée aux paramètres inutiles dans l'URL :
 - ▶ <http://www.mon-site.com/produits.html>
 - ▶ <http://www.mon-site.com/produits.html?page=1>

Le fichier « .htaccess »

Les erreurs 404 :

- ▶ C'est ce que trop de webmasters ne font pas, et c'est également ce qui massacre un référencement acquis, en plus de rapporter des pénalités.
- ▶ Changer l'emplacement d'un fichier ou le renommer entraîne obligatoirement un changement au niveau de l'URL d'accès de celui-ci.
- ▶ Seulement, Googlebot n'est pas devin.
- ▶ Au moment où il reviendra sur votre page web, il empruntera le lien qu'il avait visité la dernière fois.
- ▶ Or, ce lien n'étant plus valide, le Googlebot tombera sur une belle erreur 404.

Le fichier « .htaccess »

Les erreurs 404:

S'il y a bien une chose que Googlebot déteste par-dessus tout et qui vous rapportera énormément de pénalités, ce sont les erreurs 404.

- ▶ La solution sera donc la création d'une page 404 personnalisée, sur laquelle tous les utilisateurs, dont le Googlebot, seront redirigés dès qu'une 404 sera levée.

```
ErrorDocument 404 http://www.site.com/erreur.html
```

Le fichier « .htaccess »

Les redirections 301 :

- Cependant, ça ne suffit pas. Si l'internaute ou le Googlebot emprunte l'ancien lien de votre page et que celle-ci se trouve à un autre endroit, vous aimeriez naturellement que celui-ci soit redirigé sur la nouvelle page plutôt que sur la page d'erreur.

```
RedirectPermanent /old-url.html http://www.site.com/new-url.html
```

- Soit ajouter dans l'ancienne page l'en-tête suivant:

```
<?php  
header ("status: 301 Moved Permanently");  
header ("location: http://www.site.com/new-url.html");  
exit ();  
?>
```

Le fichier « .htaccess »

Remarque :

- ▶ Il ne faut pas oublier que le fichier « .htaccess » est un fichier texte.
- ▶ Ce qui fait qu'il est accessible en lecture par les internautes.
- ▶ Il faut un moyen pour sécuriser l'accès à ce type de fichier.
- ▶ C'est pourquoi nous ajoutons dans ce fichier l'entête suivante :

```
<Files .htaccess>  
order allow,deny  
deny from all  
</Files>
```

Autres facteurs qui entrent en jeu

L'ancienneté

- ▶ Google prend en compte l'ancienneté du nom de domaine.
- ▶ Plus un nom de domaine est ancien, plus Google aura confiance en lui.

L'hébergement

- ▶ Le temps de chargement des pages étant un des nombreux critères de l'algorithme des moteurs de recherche, il est important de choisir un hébergeur qui a une qualité correcte afin qu'il ne vous pénalise pas.
- ▶ En effet, si votre site est souvent hors service ou trop lent, Google pourrait le pénaliser dans ses résultats.
- ▶ D'ailleurs, un internaute quitte souvent un site web si celui-ci est trop lent.

Autres facteurs qui entrent en jeu

Ergonomie du site

- ▶ La facilité de navigation et de lecture du site par les internautes.
- ▶ L'utilisation des fils d'Ariane (Breadcrumbs)

Code source soigné et valide

- ▶ Prenez soins de bien structurer le code source de vos pages et assurer-vous qu'il sera valide c'est-à-dire qu'il respecte la norme W3C (au moins la page d'accueil du site).

Autres facteurs qui entrent en jeu

Maillez vos pages

- ▶ Une architecture claire
- ▶ Des liens internes et externes
- ▶ Des liens vers les derniers articles, meilleures ventes, articles populaires,...

MEILLEURES VENTES		
1		Tablette AIKUN 7'' 8Go 95,000 DT
2		Téléphone Portable EverFancy D40 Blanc 189,000 DT
3		PC Portable HP 255 Dual Core 2Go 489,000 DT
4		Mixeur plongeant Pimmy 881 39,000 DT
5		Tablette Acer Iconia B1-720 Iron 169,000 DT

Les erreurs à éviter

- ▶ Eviter les liens en JavaScript

`page`

→ Googlebot suit ce lien

`page`

→ Cette page ne sera pas visitée par les spiders

- ▶ Eviter l'utilisation des **widgets** (des réseaux sociaux) **dans la page d'accueil** ce qui ralentit le temps de chargement de la page.

Google Analytics

A quoi ça sert :

- ▶ Suivre la provenance des visiteurs
- ▶ Améliorer la conception et le contenu du site
- ▶ Optimiser les campagnes marketing
- ▶ Comparer les indicateurs...



Google Analytics

► Démonstration www.google.tn/intl/fr/analytics/



Outil génération de lien

- ▶ Démonstration <https://support.google.com/analytics/answer/1033867?hl=fr>



Google Webmaster Tools

- ▶ Permet de suivre l'indexation de votre site
- ▶ Soumettre le sitemap
- ▶ Liste des erreurs, liens morts,....
- ▶ Contrôler les performances du site dans les résultats de recherche Google (par rapport au mots-clés tapés par les internautes).
- ▶ Afficher les liens vers son site (backlink)



Google Webmaster Tools

► Démonstration <https://www.google.com/webmasters/tools/>





En conclusion

Stratégie de référencement

1. Audit

- ▶ Evaluer les sites des concurrents, élaborer une stratégie, examiner les mots-clés, trouver sa valeur ajoutée

2. Conception

- ▶ Développer le site, optimiser le site (onpage)

3. Référencement

- ▶ Soumission, stratégie efficace de netlinking

4. Suivi et optimisation

- ▶ Un processus continu de web analytics, de mise à jour et d'amélioration

ON-THE-PAGE FACTORS

These elements are in the direct control of the publisher

CONTENT

Cq	QUALITY	Are pages well written & have substantial quality content?
Cr	RESEARCH	Have you researched the keywords people may use to find your content?
Cw	WORDS	Do pages use words & phrases you hope they'll be found for?
Ce	ENGAGE	Do visitors spend time reading or "bounce" away quickly?
Cf	FRESH	Are pages fresh & about "hot" topics?
Vt	THIN	Is content "thin" or "shallow" & lacking substance?
Va	ADS	Is your content ad-heavy, especially "above the fold?"
HTML		
Ht	TITLES	Do HTML title tags contain keywords relevant to page topics?
Hd	DESCRIPTION	Do meta description tags describe what pages are about?
Hh	HEADERS	Do headlines & subheads use header tags with relevant keywords?
Hs	STRUCTURE	Do pages use structured data to enhance listings?
Vs	STUFFING	Do you excessively use words you want pages to be found for?
Vh	HIDDEN	Do colors or design "hide" words you want pages to be found for?
ARCHITECTURE		
Ac	CRAWL	Can search engines easily "crawl" pages on site?
Ad	DUPLICATE	Does site manage duplicate content issues well?
As	SPEED	Does site load quickly?
Au	URLS	Are URLs short & contain meaningful keywords to page topics?
Am	MOBILE	Does your site work well for mobile visitors, on smartphones and tablets?
Vc	CLOAKING	Do you show search engines different pages than humans?

THE PERIODIC TABLE OF SEO SUCCESS FACTORS

Search engine optimization — SEO — seems like alchemy to the uninitiated. But there's a science to it. Below are some important "ranking factors" and best practices that can lead to success with both search engines and searchers.

ON-THE-PAGE SEO

OFF-THE-PAGE SEO

CONTENT	HTML	ARCHITECTURE	LINKS	TRUST	SOCIAL	PERSONAL
Cq Quality	Ht Titles	Ac Crawl	Lq Quality	Ta Authority	Sr Reputation	Pc Country
Cr Research	Hd Description	Ad Duplicate	Lt Text	Th History	Ss Shares	Pl Locality
Cw Words	Hh Headers	As Speed	Ln Numbers	Ti Identity		Ph History
Ce Engage	Hs Structure	Au URLs	Vp Paid	Vd Piracy		Ps Social
Cf Fresh	Vs Stuffing	Am Mobile		VI Spam		
Vt Thin	Vh Hidden	Vc Cloaking				
Va Ads						

FACTORS WORK TOGETHER

All factors on the table are important, but those marked 3 carry more weight than 1 or 2. No single factor guarantees top rankings or success, but having several favorable ones increases the odds. Negative "violation" factors shown in red harm your chances.

OFF-THE-PAGE FACTORS

Elements influenced by readers, visitors & other publishers

LINKS

Lq	QUALITY	Are links from trusted, quality or respected web sites?
Lt	TEXT	Do links pointing at pages use words you hope they'll be found for?
Ln	NUMBER	Do many links point at your web pages?
Vp	PAID	Have you purchased links in hopes of better rankings?
VI	SPAM	Have you created many links by spamming blogs, forums or other places?

TRUST

Ta	AUTHORITY	Do links, shares & other factors make site a trusted authority?
Th	HISTORY	Has site or its domain been around a long time, operating in same way?
Ti	IDENTITY	Does site use means to verify its identity & that of authors?
Vd	PIRACY	Has site been flagged for hosting pirated content?

SOCIAL

Sr	REPUTATION	Do those respected on social networks share your content?
Ss	SHARES	Do many share your content on social networks?

PERSONAL

Pc	COUNTRY	What country is someone located in?
Pl	LOCALITY	What city or local area is someone located in?
Ph	HISTORY	Has someone regularly visited your site or socially favored it?
Ps	SOCIAL	Have your friends socially favored the site?

Written By: [Search Engine Land](#)

Learn More: <http://selind.com/seotable>

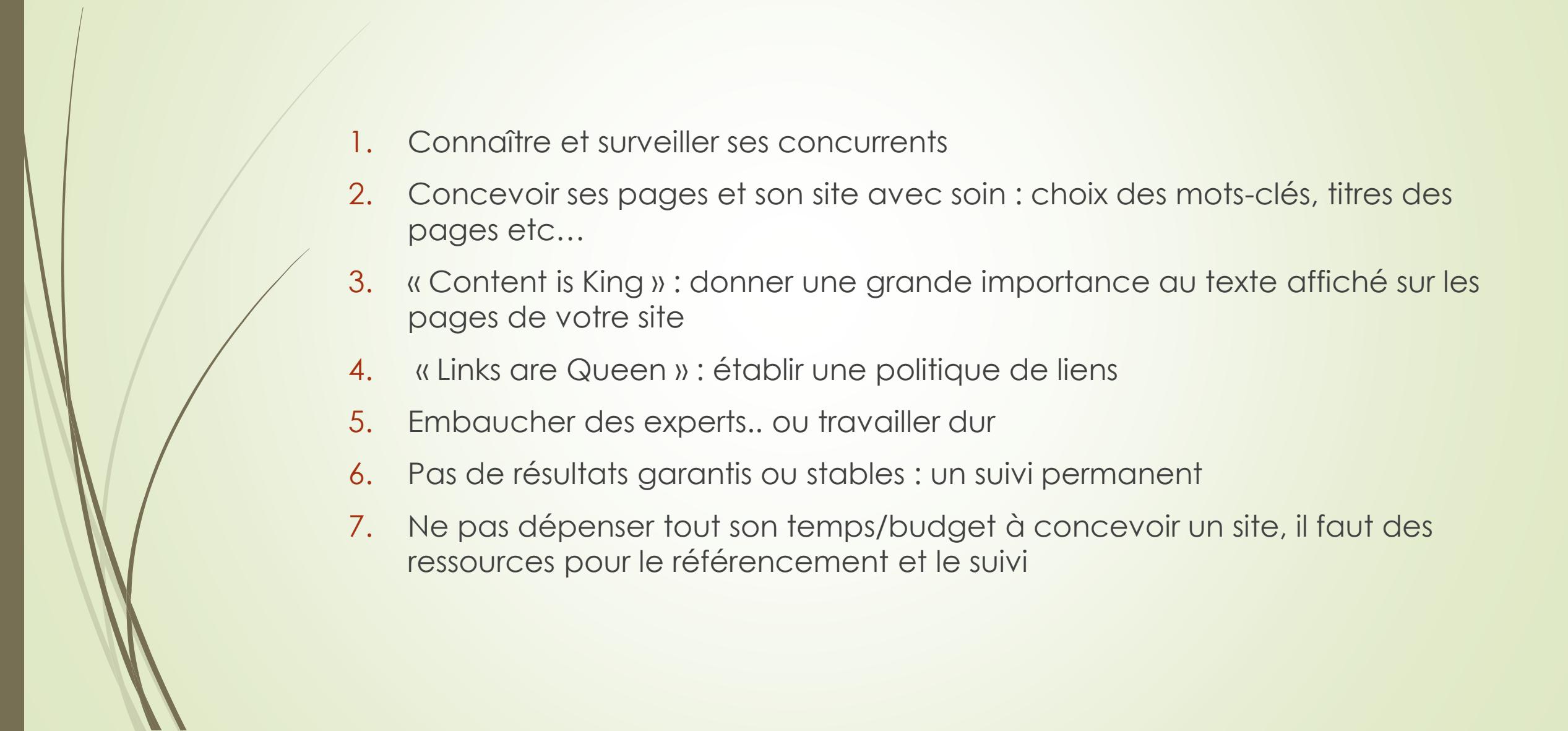
Design By: [COLUMN FIVE](#)

Copyright Third Door Media

Source : <http://searchengineland.com/seotable>



Les commandements du SEO

- 
1. Connaître et surveiller ses concurrents
 2. Concevoir ses pages et son site avec soin : choix des mots-clés, titres des pages etc...
 3. « Content is King » : donner une grande importance au texte affiché sur les pages de votre site
 4. « Links are Queen » : établir une politique de liens
 5. Embaucher des experts.. ou travailler dur
 6. Pas de résultats garantis ou stables : un suivi permanent
 7. Ne pas dépenser tout son temps/budget à concevoir un site, il faut des ressources pour le référencement et le suivi

Qu'est ce que la réussite?



On dit qu'on a réussi lorsque :

1. Les gens viennent sur le site (Référencement, netlinking ...)
2. Restent pour le consulter (Taux de rebond raisonnable)
3. Agissent de la manière attendue = conversion

Taux de conversion =

taux de conversion online = Achat en ligne, inscription, abonnement à une newsletter, téléchargement...

+ taux de conversion offline = recherche du magasin le plus proche appels téléphoniques, fréquentation d'un magasin, achat...

4. Sont fidélisés = Mettent l'adresse du site dans leurs favoris, reviennent, s'abonnent (Mise à jour de l'information, pull/push, RSS...).
5. Le facteur principal de promotion d'un site est son utilité et son contenu : « content is King »



Search Engine Marketing



SEM = SEO + SEA + SMO



Optimisation pour le web mobile

Bonnes pratiques

Source : www.opquast.com



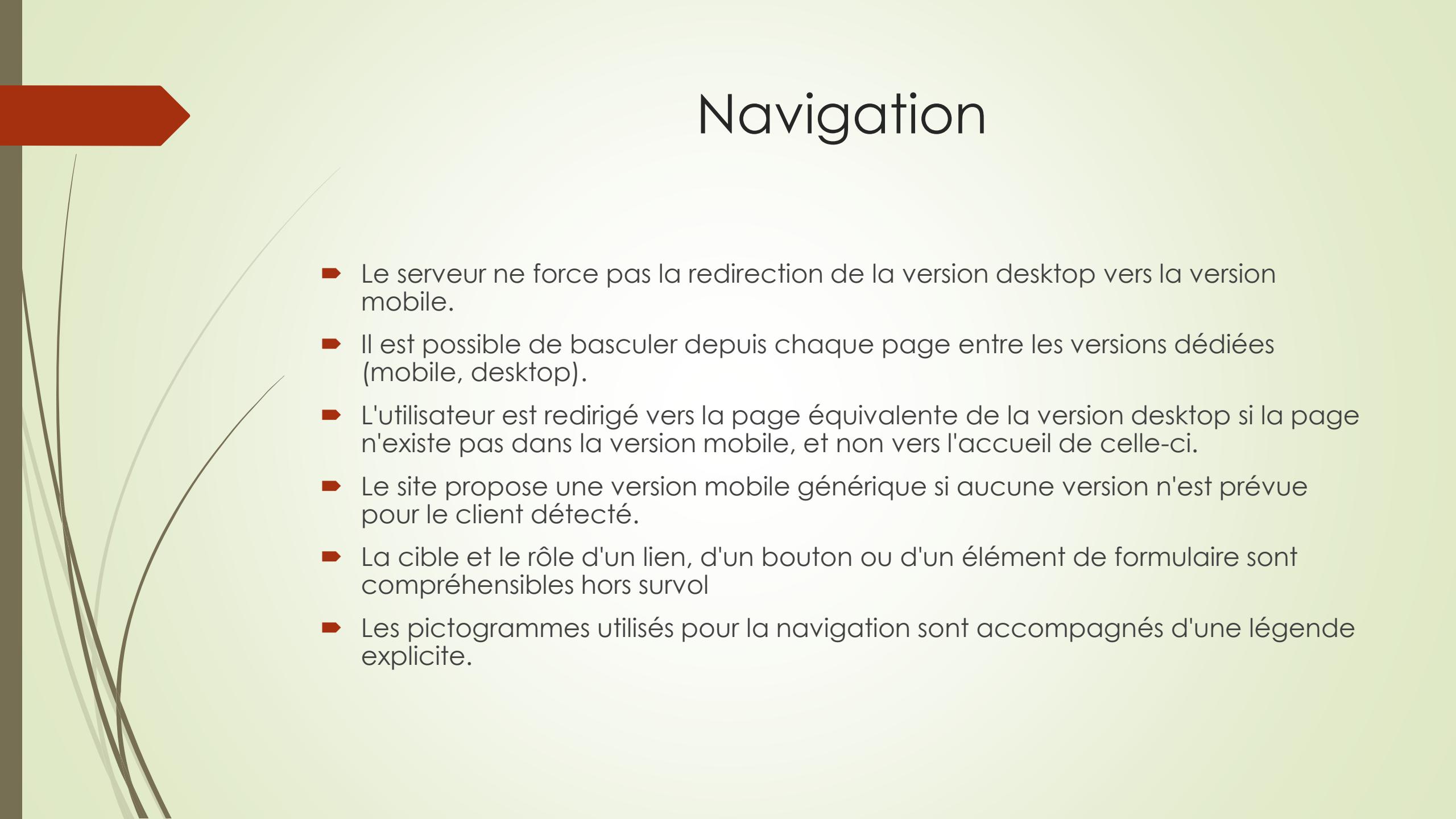
Interactions

- ▶ Les champs de saisie de type mail, URL, téléphone, nombre, recherche, mots de passe, heure et date sont dotés du type approprié.
- ▶ Les numéros de téléphone sont activables via le protocole approprié.
- ▶ Chaque champ de formulaire peut être activé via une action sur son étiquette.
- ▶ Chaque zone d'interaction tactile est de taille suffisante.
- ▶ Les espaces entre les zones d'interaction tactiles sont suffisants.
- ▶ Les espaces entre les zones d'interaction tactile et les bords de la zone d'affichage du navigateur sont suffisants.



Multimédia

- ▶ Les contenus audiovisuels sont accompagnés d'une transcription textuelle équivalente.
- ▶ Les animations et contenus multimédia utilisent les éléments HTML5 appropriés.
- ▶ Les contenus multimédia ne sont téléchargés qu'à la demande explicite de l'utilisateur.



Navigation

- ▶ Le serveur ne force pas la redirection de la version desktop vers la version mobile.
- ▶ Il est possible de basculer depuis chaque page entre les versions dédiées (mobile, desktop).
- ▶ L'utilisateur est redirigé vers la page équivalente de la version desktop si la page n'existe pas dans la version mobile, et non vers l'accueil de celle-ci.
- ▶ Le site propose une version mobile générique si aucune version n'est prévue pour le client détecté.
- ▶ La cible et le rôle d'un lien, d'un bouton ou d'un élément de formulaire sont compréhensibles hors survol
- ▶ Les pictogrammes utilisés pour la navigation sont accompagnés d'une légende explicite.

Présentation

- ▶ Le site ne bloque pas les fonctionnalités de zoom du navigateur.
- ▶ Le site n'impose pas de rafraîchissement automatique de la page entière.
- ▶ L'utilisation du zoom du navigateur ne provoque pas de superpositions de contenus.
- ▶ Les alertes javascript invitant à l'installation d'une application mobile ne se produisent qu'une seule fois par session.
- ▶ La promotion de l'application mobile ne recourt ni aux alertes javascript ni aux splashscreens.
- ▶ Le contenu des pages s'adapte via les styles aux caractéristiques de l'appareil de l'utilisateur.



Performances

- ▶ Les ressources CSS sont minifiées.
- ▶ Les ressources javascript sont minifiées.
- ▶ Les animations CSS ne modifient pas le positionnement ni les dimensions.
- ▶ Le serveur transmet des contenus compressés aux clients qui les acceptent.
- ▶ Le serveur envoie les informations permettant la mise en cache des contenus.