

# Introduction à CSS

## TP : réalisation des pages pour le site d'une boutique de surf

CSS, ou *Cascading Style Sheets* (feuilles de style en cascade), est le langage qui permet de définir l'apparence visuelle des pages web. Alors que HTML structure le contenu, CSS le stylise en contrôlant les couleurs, les polices, les espacements et bien plus encore. En séparant le contenu de la présentation, CSS rend le code plus organisé et facilite la gestion des styles d'un site entier à partir d'un seul fichier.

Le CSS utilise des règles composées de sélecteurs et de propriétés. Les sélecteurs identifient les éléments HTML ciblés, et les propriétés définissent les caractéristiques visuelles appliquées (par exemple, color pour la couleur de texte, font-size pour la taille des polices, ou margin pour les espacements). Un seul fichier CSS peut transformer l'apparence de plusieurs pages en harmonisant les styles sur tout un site.

Avec l'évolution vers CSS3, des fonctionnalités avancées comme les animations, les effets de transition, et les transformations 3D sont devenues possibles, enrichissant l'expérience utilisateur. CSS est donc essentiel pour créer des sites à la fois esthétiques et adaptatifs, répondant aux exigences de design modernes et aux attentes des utilisateurs.

Dans ce TP, vous allez découvrir le CSS en réalisant des pages pour un site d'une boutique de surf fictive.

### Exercice 1 : Squelette de la page

Dans un fichier index.html, réalisez le squelette de votre site qui doit contenir comme sur l'image ci-dessous :

- un titre de page
- un titre principal
- des titres secondaires
- un paragraphe d'introduction
- une illustration
- une liste pour les différentes planches et accessoires disponibles
- une paragraphe « Copyright »
- un paragraphe mentions légales



## Bienvenue

### Qui sommes-nous ?

Notre boutique située sur la côte, vend les plus belles planches à surf depuis 1989.



### Nos produits

#### Nos planches

- Shortboard
- Longboard
- Paddleboard
- Bodyboard

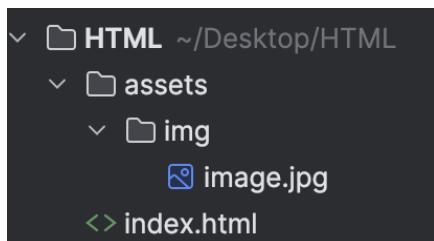
#### Accessoires

- Leash
- Wax
- Combinaison

Copyright©2024 Ma boutique de surf

Mentions légales

Pour mieux vous organiser, je vous conseille de ranger vos fichiers de cette manière :



## Exercice 2 : Zoning de la page

Nous venons d'ajouter des paragraphes, des titres et des listes.

Maintenant, il serait pertinent de structurer le contenu en zones afin de faciliter l'application de styles personnalisés par la suite.

Organisez la page en différentes sections :

- une zone pour l'« en-tête » (header)
- La section « produits » sera subdivisée en deux sous-sections.
  - une zone pour les « planches »,
  - une zone pour le « accessoires »
- une zone pour le « bas-de-page » (footer)

Utilisez les balises sémantiques HTML5 appropriées, et, si nécessaire, la balise div. Pour rappel, cette dernière permet de regrouper divers éléments, contrairement à un paragraphe qui ne peut contenir d'autres paragraphes, par exemple.

Dans l'en-tête, nous regrouperons les éléments allant de « Bienvenue » à l'image du surfeur.

Exemple de zoning de la page :

## Bienvenue

### Qui sommes-nous ?

Notre boutique située sur la côte, vend les plus belles planches à surf depuis 1989.



### Nos produits

#### Nos planches

- Shortboard
- Longboard
- Paddleboard
- Bodyboard

#### Accessoires

- Leash
- Wax
- Combinaison

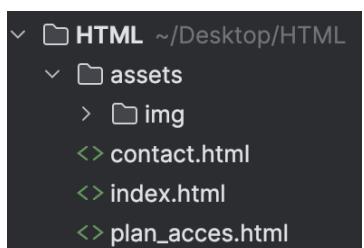
Copyright©2024 Ma boutique de surf

Mentions légales

## Exercice 3 : Navigation

Notre site doit contenir plusieurs pages, nous allons donc en ajouter d'autres et créer des liens pour permettre la navigation entre elles.

1. En prenant la première page comme modèle (avec les zones « en-tête » et « pied de page » identiques), créez deux nouvelles pages :
  - une page « plan d'accès »
  - une page « contact » (*vous pouvez mettre le formulaire réalisé dans le TP1 : introduction à HTML*)



2. Ajoutez une zone « menu » pour faciliter la navigation entre les différentes pages.

# Bienvenue

## Qui sommes-nous ?

Notre boutique située sur la côté, vend les plus belles planches à surf depuis 1989.



[Accueil](#) [Plan d'accès](#) [Contact](#)

## Nos produits

### Nos planches

- Shortboard
- Longboard
- Paddleboard
- Bodyboard

### Accessoires

- Leash
- Wax
- Combinaison

Copyright©2024 Ma boutique de surf

Mentions légales

## Modification du style de la page

Le CSS (Cascading Style Sheets) est un langage qui permet de modifier l'apparence des éléments d'une page, en ajustant par exemple la taille, le style, et la couleur du texte, la couleur de fond, les bordures, ou encore la position des éléments sur la page.

Il existe plusieurs façons d'ajouter des styles CSS : Chaque élément HTML possède un attribut style dans lequel on peut insérer des déclarations de style directement.

```
<div style="background-color:orange; border:1px solid black;  
color:yellow; font-size:150%; padding:1em;">  
  
    Cette balise div a du style !  
  
</div>
```

Il existe aussi la balise <style>, qui permet de regrouper plusieurs déclarations CSS dans la section <head> de la page.

```
<style>
  div {
    background-color: #339;
    color: #fff;
    padding: 15px;
    border-bottom: 5px solid red;
    margin-bottom: 15px;
  }
</style>
```

Une autre solution consiste à inclure un fichier externe contenant tout ou une partie des déclarations CSS (feuille de style), également placé dans le <head> de la page.

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

Cette troisième méthode est particulièrement avantageuse, car elle centralise les styles en un seul fichier, facilitant ainsi les modifications : il devient possible de changer un style commun à plusieurs pages sans devoir éditer chaque page individuellement.

Une déclaration CSS se compose d'un sélecteur, suivi d'une série de paires « propriété-valeur » : sélecteur { propriété1 : valeur1 ; propriété2 : valeur2 ; ... }

Exemple :

```
p {
  display: block;
  margin-top: 1em;
  margin-bottom: 1em;
}
```

## Comment appliquer un style aux balises HTML standard ?

Avec CSS, il est possible de modifier l'apparence de toutes les balises HTML standard en spécifiant simplement le nom de la balise et en définissant les styles souhaités. Placez ces définitions de style dans une feuille de styles externe ou dans la section <head> de la page.

Par exemple, le code suivant appliquera un style gras et un interligne de 1,3em à toutes les balises <p> de la page :

```
p {  
    font-weight: bold;  
    line-height: 1.3em;  
}
```

### Remarque :

em est une unité de mesure en CSS utilisée pour définir des tailles relatives, en général par rapport à la taille de la police de l'élément parent. D'autres unités couramment employées incluent le px (pixel), qui est une unité fixe, le %, qui ajuste la taille en fonction de l'élément conteneur, et rem, une unité relative à la taille de police de l'élément racine (souvent le <html>). Pour approfondir et découvrir comment utiliser ces unités efficacement, vous pouvez consulter cette [page](#).

## Utilisation des classes pour styliser des éléments spécifiques

Vous pouvez aussi attribuer des classes aux éléments HTML pour appliquer des styles personnalisés. Vous définissez le nom et les styles de chaque classe, qui remplaceront les styles par défaut des éléments concernés.

Pour créer une classe, définissez son nom précédé **d'un point** dans la feuille de styles. Assurez-vous que le nom de classe ne comporte pas d'espace pour éviter toute confusion.

```
.nom-de-la-classe {  
    color: green;  
}
```

Placez ces définitions de style dans une feuille de style ou dans le <head> de votre page. Ensuite, pour appliquer un style spécifique à un élément, ajoutez class="nom-de-la-classe" dans l'attribut de la balise :

```
<p class="nom-de-la-classe">Texte stylisé</p>
```

Les éléments HTML peuvent comporter plusieurs classes. Pour en ajouter plusieurs à un même élément, listez simplement les noms des classes en les séparant par un espace.

```
<p class="mon-style1 mon-style2">  
    Les styles des deux classes s'appliquent à ce paragraphe  
</p>
```

## Utilisation des "ID" pour appliquer un style

En plus des classes, un élément HTML peut également avoir un ID. Bien que les ID fonctionnent de manière similaire aux classes, il existe une différence importante :

- **Plusieurs** éléments peuvent partager la même classe.
- En revanche, chaque ID doit être **unique** dans la page : un seul élément peut avoir un ID donné.

Pour créer un ID, placez son nom précédé d'un dièse. Assurez-vous que le nom d'ID ne contient pas d'espace pour éviter toute confusion.

```
#mon-id {  
    color: green;  
}
```

Pour appliquer le style d'un ID à un élément, ajoutez l'attribut id="mon-id" dans la balise :

```
<p id="mon-id">Texte avec style unique</p>
```

### Exercice 4 : Stylisons la page

1. Créez une feuille de style permettant de définir :
  - La couleur de fond du site de votre choix (*background-color*)
  - La couleur du titre de la page de votre choix (*color*)
2. Ajoutez dans vos pages html, la balise link mettant en relation votre feuille de style.

## Bienvenue

### Qui sommes-nous ?

Notre boutique située sur la côte, vend les plus belles planches à surf depuis 1989.



- [Accueil](#)
- [Plan d'accès](#)
- [Contact](#)

### Nos produits

#### Nos planches

- Shortboard
- Longboard
- Paddleboard
- Bodyboard

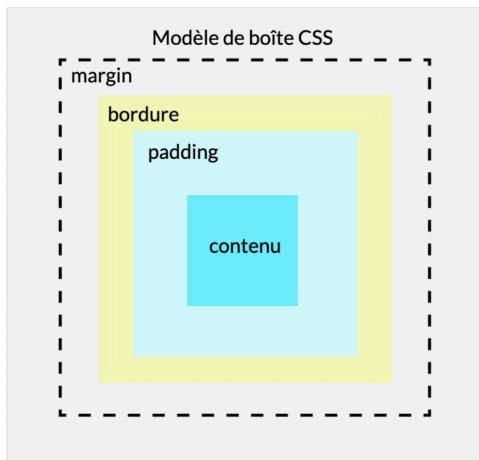
#### Accessoires

- Leash
- Wax
- Combinaison

Copyright©2024 Ma boutique de surf

## Marge extérieure et marge intérieure

Un élément peut posséder deux types de marges de part et d'autre de la bordure. Des marges extérieures (margin) des marges intérieures (padding).



## Exercice 5

Réalisez les modifications suivantes :

- dans le header (partie haute) :
  - avoir une bordure basse de 0.5px, solide, de la couleur de votre choix
  - une marge intérieure de 20px
  - modifiez la marge de l'image : marge haute : 40px, marge basse: 20 px, marge gauche: 25px
- dans le menu :

- mettre les éléments dans une liste
- les éléments de la liste ne doivent plus être précédés d'un point
- dans la liste des produits :
  - les éléments de la liste doivent être précédés d'un carré et plus d'un point
- dans le footer (partie basse) :
  - avoir une bordure haute de 1px, solide, de la couleur de votre choix
  - une marge intérieure gauche et droite de 20px
  - une marge extérieure haute et basse de 10px

## Bienvenue

### Qui sommes-nous ?

Notre boutique située sur la côté, vend les plus belles planches à surf depuis 1989.



---

[Accueil](#)  
[Plan d'accès](#)  
[Contact](#)

### Nos produits

#### Nos planches

- Shortboard
- Longboard
- Paddleboard
- Bodyboard

#### Accessoires

- Leash
- Wax
- Combinaison

---

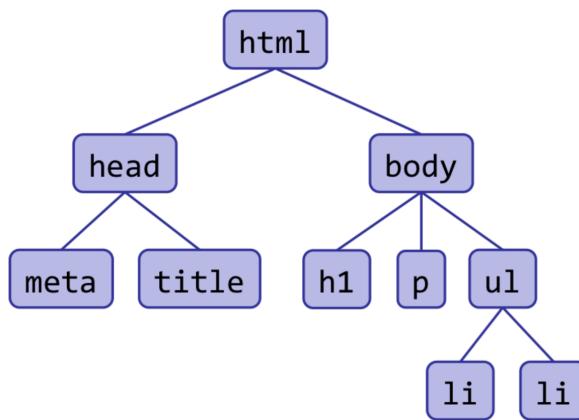
Copyright©2024 Ma boutique de surf  
Mentions légales

## Sélection par héritage

L'utilisation de nombreuses classes peut rapidement encombrer le code HTML. Pour alléger ce code, CSS permet de cibler uniquement les « descendants » d'un élément, avec la possibilité de restreindre la sélection aux « enfants directs » seulement.

## Le modèle Parent-Enfant

En CSS, l'héritage repose sur le modèle Parent-Enfant : chaque élément Enfant hérite automatiquement des styles définis pour son élément Parent.



Par exemple, la balise <html> est le parent de <body>, et <body> est le parent de <ul>, qui est lui-même le parent des <li>.

Cet héritage est très utile pour éviter des répétitions inutiles : lorsqu'une propriété est appliquée à un élément parent (par exemple, font-size: 1.5em), elle est transmise à tous ses enfants, y compris aux sous-niveaux d'enfants, etc.

**Précision :** l'élément enfant n'héritera que des propriétés CSS transmissibles, car certaines, comme margin et padding (qui concernent les blocs), ne se transmettent pas automatiquement.

## Le sélecteur de descendance

Le sélecteur de descendance, représenté par un espace entre deux sélecteurs, permet de combiner deux sélecteurs pour cibler des éléments correspondant au second sélecteur uniquement s'ils sont descendants du premier.

```
/* Les éléments <li> qui sont des descendants d'un <ul>
class="truc">*/

ul.truc li {
  margin: 2em;
}
```

## Le sélecteur d'enfants directs (X > Y)

```
#container > ul {
  border: 1px solid black;
}
```

Le sélecteur X > Y cible uniquement les enfants directs de X, contrairement au sélecteur de descendance standard X Y, qui cible tous les niveaux de descendants. Par exemple, dans la structure suivante :

```
<div id="container">
  <ul>
    <li>
      <ul></ul>
    </li>
  </ul>
</div>
```

Le sélecteur #container > ul ne sélectionnera que les `<ul>` directement enfants de la `div` avec l'ID `container` et ignorerá les `<ul>` qui se trouvent plus profondément dans la hiérarchie, comme celui à l'intérieur du `<li>`.

Il existe d'autres types de sélecteurs, les plus curieux d'entre vous peuvent les consulter dans cette [documentation](#).

## Exercice 6

En utilisant la sélection en héritage, faire en sorte que les titres des listes de produits soient en italique dans la couleur de votre choix et avec une marge intérieure gauche de 5 pixels.

### Nos produits

#### *Nos planches*

- Shortboard
- Longboard
- Paddleboard
- Bodyboard

#### *Accessoires*

- Leash
- Wax
- Combinaison

## Premiers pas vers le positionnement des éléments en CSS

CSS propose plusieurs méthodes de positionnement, notamment le positionnement par défaut en flux, ainsi que des options comme le positionnement absolu, fixe, flottant, relatif, en grille, tabulaire, ou à l'aide de flexbox. Chacune de ces méthodes a ses propres règles et spécificités. Une bonne compréhension de ces règles est essentielle pour éviter des comportements inattendus et faciliter le débogage.

Avec la diversité des supports (smartphones, tablettes, grands écrans) et les exigences d'accessibilité (comme éviter que des éléments se chevauchent lors de zooms), on priviliege aujourd'hui un affichage dynamique, ajusté aux caractéristiques de la zone d'affichage (*responsive design*).

## Notion de flux

Le flux d'un document décrit la disposition naturelle des éléments d'une page selon l'ordre de leur déclaration dans le HTML. Dans ce flux :

- Les éléments de type *bloc* (comme `<h1>`, `<p>`, `<ul>`, `<table>`) occupent toute la largeur de leur conteneur, permettent des marges verticales, et peuvent avoir une hauteur et une largeur définies.
- Les éléments de type *inline* (comme `<a>`, `<img>`, `<strong>`) se placent en ligne les uns après les autres.

Un élément peut être « reclassé » de bloc à inline (ou inversement) via la propriété `display`. Lorsque le positionnement naturel du flux ne suffit plus, d'autres options de positionnement peuvent être utilisées.

## La propriété position

### Position relative

La position relative (`position: relative`) permet de décaler un élément par rapport à sa position dans le flux, sans affecter les autres éléments. Elle est utile pour :

- Servir de repère pour des éléments enfants positionnés en absolu.
- Permettre l'utilisation de la propriété `z-index` pour gérer des superpositions, ce qui est impossible pour des éléments en flux.

### Position absolue

La position absolue (`position: absolute`) détache un élément du flux et lui permet de se positionner indépendamment de l'ordre HTML. Contrairement aux éléments flottants, un élément en position absolue se réfère au premier ancêtre positionné (c'est-à-dire un élément avec `position` défini). Tant qu'une propriété (`top`, `bottom`, `left`, ou `right`) n'est pas spécifiée, l'élément reste à sa position d'origine.

### Position fixe

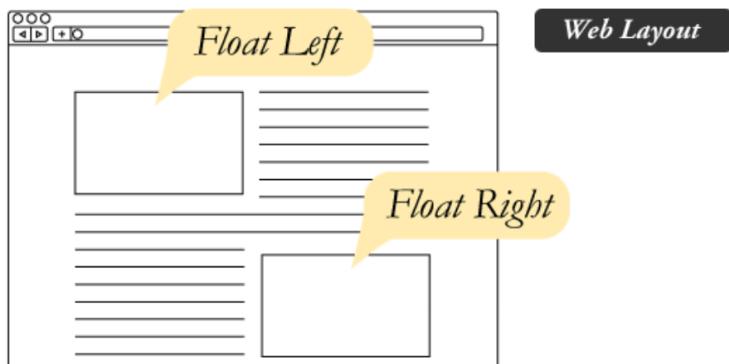
Le positionnement fixe (`position: fixed`) fonctionne comme le positionnement absolu, mais se réfère toujours à la fenêtre du navigateur. Ainsi, un élément en position fixe reste en place même lors du défilement de la page.

### Position statique

La position statique (`position: static`) est le positionnement par défaut. Elle place l'élément dans le flux naturel, sans le décaler. Elle n'est généralement précisée que pour remettre un élément dans le flux.

### Position flottante (float)

La propriété float permet de déplacer un élément sur le côté gauche (float: left) ou droit (float: right) de son conteneur.



Elle est souvent utilisée pour créer un habillage autour de l'élément flottant. Les éléments qui suivent dans le flux entourent alors l'élément flottant. Si deux éléments flottent dans la même direction, ils se placent côte à côte.

**Nettoyage des flottants** : La propriété clear est utilisée conjointement avec float pour éviter que les éléments suivants subissent l'habillage. En définissant clear sur left, right, ou both, on place l'élément en dessous des éléments flottants.

Vous pouvez consulter cette [documentation](#) si vous souhaitez en apprendre davantage.

### Exercice 7 :

1. En utilisant le positionnement relatif et le positionnement absolu, faire que l'image se positionne dans le coin supérieur droit de l'en-tête en déclarant :
  - la zone d'en-tête en positionnement relatif,
  - l'image en positionnement absolu avec une distance à zéro du haut et une distance à zéro du bord droit du bloc référent.
2. Que peut-il se passer lorsque la zone d'affichage diminue en largeur ? (testez en modifiant la taille de votre fenêtre).

The screenshot shows a light blue header section of a website. On the left, the word 'Bienvenue' is displayed in a dark teal font. Below it, the question 'Qui sommes-nous ?' is asked in a bold black font. A horizontal line separates this text from a descriptive paragraph. To the right of the text, there is a photograph of a surfer riding a wave, with the surfer's body and the white foam of the wave visible against the blue water. The overall design is clean and modern.

Bienvenue

Qui sommes-nous ?

Notre boutique située sur la côté, vend les plus belles planches à surf depuis 1989.

## Introduction à Flexbox

Flexbox (ou modèle de boîte flexible) est un système de disposition CSS conçu pour faciliter l'alignement, la disposition et la distribution de l'espace entre les éléments d'un conteneur, même si leur taille est dynamique ou inconnue.

L'objectif de Flexbox est d'offrir au conteneur la capacité d'adapter la taille, la largeur et l'ordre de ses éléments pour occuper au mieux l'espace disponible, ce qui le rend particulièrement utile pour des interfaces responsives.

Un conteneur flex ajuste la taille de ses éléments pour qu'ils remplissent l'espace disponible ou se rétractent pour éviter tout débordement.

Guide : <https://css-tricks.com/snippets/css/a-guide-to-flexbox/>

## Propriétés du conteneur Flexbox

**display** : active le mode flex pour le conteneur.

**flex-direction** : détermine l'axe principal et la direction des éléments (row par défaut).

**flex-wrap** : contrôle le retour à la ligne des éléments.

- nowrap (valeur par défaut) : tous les éléments flex sont disposés sur une seule ligne.
- wrap : les éléments sont disposés sur plusieurs lignes, de haut en bas.
- wrap-reverse : les éléments sont disposés sur plusieurs lignes, de bas en haut.

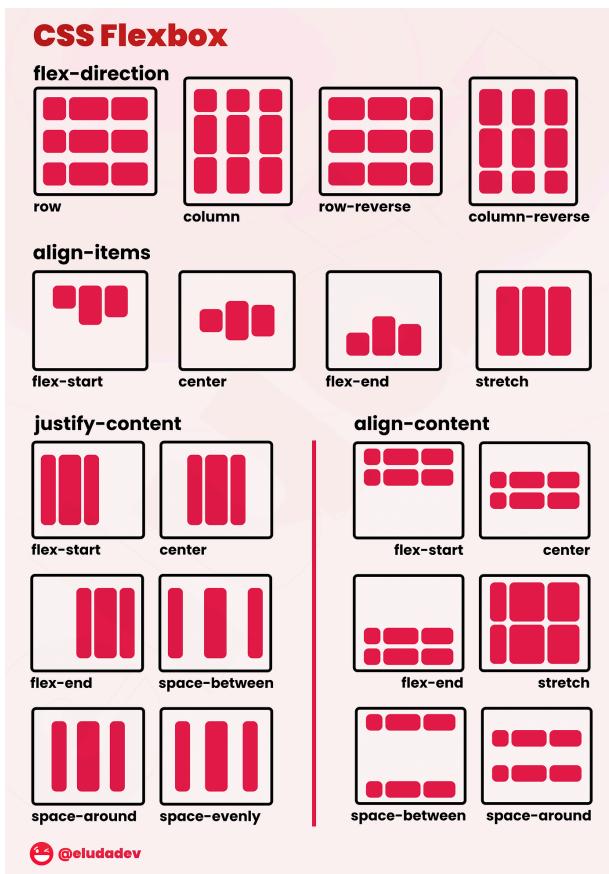
**flex-flow** : raccourci pour flex-direction et flex-wrap.

**justify-content** : définit l'alignement des éléments le long de l'axe principal.

- flex-start (par défaut) : les éléments sont groupés au début de la ligne.
- flex-end : les éléments sont groupés à la fin de la ligne.
- center : les éléments sont centrés.
- space-between : les éléments sont répartis avec un espace égal entre eux.
- space-around : espace égal autour de chaque élément, mais avec un espace visuellement inégal aux extrémités.
- space-evenly : les espaces entre chaque élément et les bords du conteneur sont égaux.

**align-items** : contrôle l'alignement des éléments le long de l'axe secondaire.

- stretch (par défaut) : les éléments s'étirent pour remplir le conteneur.
- flex-start : les éléments sont alignés en haut de l'axe croisé.
- flex-end : les éléments sont alignés en bas de l'axe croisé.
- center : les éléments sont centrés sur l'axe croisé.
- baseline : les éléments sont alignés sur leur ligne de base de texte.



## Propriétés pour les éléments enfants (éléments flexibles)

**order** : contrôle l'ordre d'affichage des éléments dans le conteneur flex. Par défaut, ils suivent l'ordre dans le code source.

**flex-grow** : indique la quantité d'espace qu'un élément doit occuper par rapport aux autres. Si chaque élément a le même facteur d'expansion, l'espace libre sera distribué également entre eux.

**flex-shrink** : permet à un élément de se contracter si nécessaire.

**flex-basis** : fixe la taille initiale de l'élément avant que l'espace restant ne soit distribué. Elle peut être une longueur spécifique (20%, 5rem, etc.) ou auto, ce qui indique de se référer aux propriétés width ou height de l'élément.

**flex** : raccourci combinant flex-grow, flex-shrink et flex-basis. Utiliser cette propriété simplifiée garantit que tous les paramètres flex sont définis.

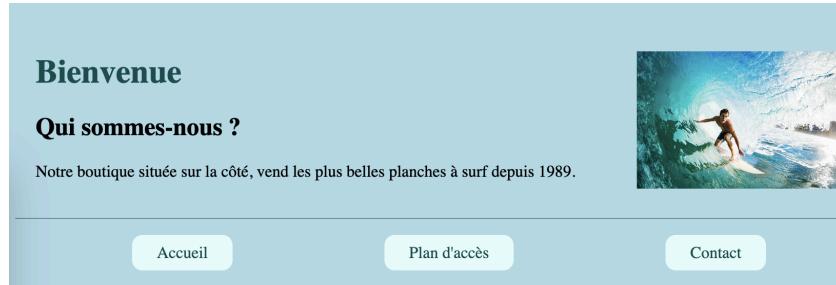
**align-self** : permet d'outrepasser l'alignement défini par align-items pour des éléments individuels, offrant un contrôle granulaire sur chaque élément flexible.

Pour mieux comprendre, voici un générateur : <https://angrytools.com/css-flex/>

Pour vous exercer, il existe ce jeu : <https://flexboxfroggy.com/#fr>

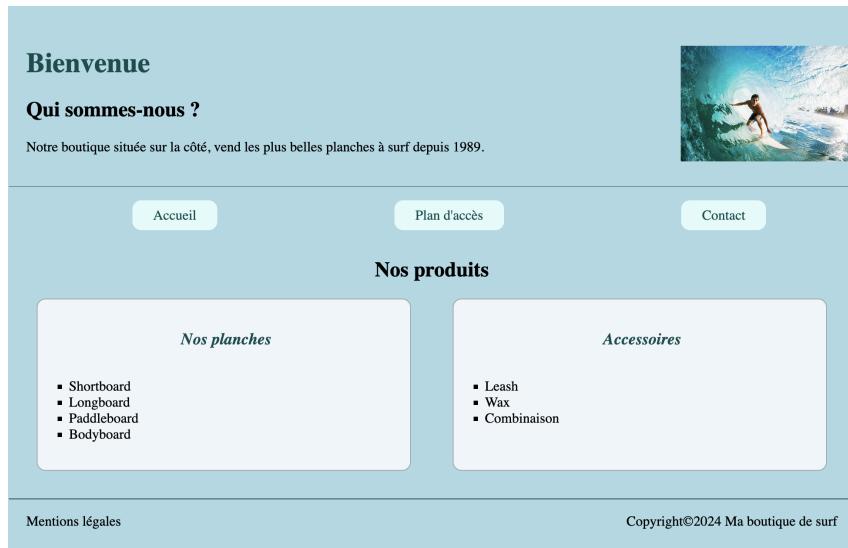
## Exercice 8 :

1. Réalisez l'affichage suivant à l'aide des flexbox:



- Déclarez la zone du menu en « flex »
- Chaque item doit occuper un tiers de l'espace disponible
- Chaque item est sans décoration, avec un fond de couleur (au choix), les coins sont arrondis, une marge intérieure gauche et droite plus grande que la marge intérieure haute et basse

2. Positionnez les blocs de produits et les éléments de bas de page comme ceci :



3. Passez vos fichiers au validateur W3C :

- HTML : <https://validator.w3.org>
- CSS : <https://jigsaw.w3.org/css-validator/>

Ce TP doit être rendu **à la fin du cours ou la veille de la prochaine séance**, au format **.zip**, en incluant tous les fichiers que vous avez créés, par e-mail à l'adresse suivante : [adelaide.louis@campuslandes.com](mailto:adelaide.louis@campuslandes.com).

### Optionnel :

Pour les plus rapides, remplacez les flexbox avec des grid :

Un guide : <https://css-tricks.com/snippets/css/complete-guide-grid/>

Un générateur : <https://angrytools.com/css-grid/>