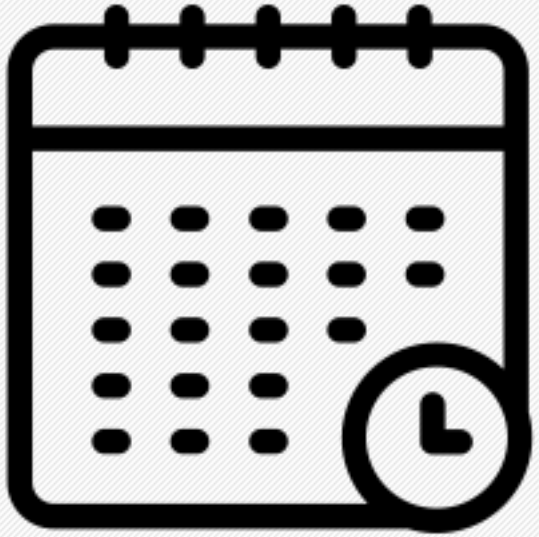

# Conception de site web avec HTML et CSS

## Session 5: bases du CSS

Rkia Fajr

# Informations sur la session





- Bases du CSS
- 15 Novembre 2022
- De 09h00 à 11h00

# Objectifs de la session

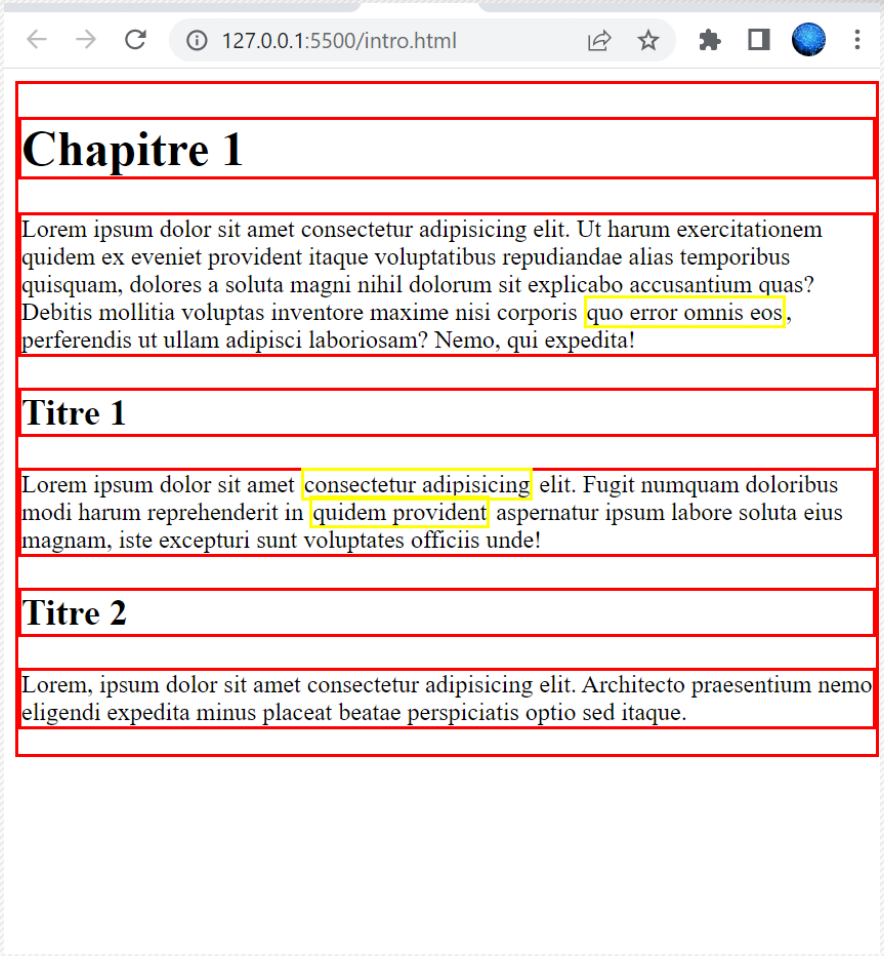
- Présenter le fonctionnement du CSS
- Apprendre à écrire des règles CSS
- Montrer comment les règles CSS s'appliquent aux pages HTML
- Comment spécifier les couleurs en CSS

# Introduction

# Comprendre CSS



- CSS vous permet de créer des règles qui contrôlent la manière dont chaque boîte individuelle (et le contenu de cette boîte) est présenté.



# Définition de style CSS



# Définition de style CSS



```
h1, h2, h3 {
        font-family: Arial;
        color: yellow;}
```

PROPERTY VALUE

# Exemple de style



## Chapitre 1

Lorem ipsum *dolor* sit amet, consectetur adipisicing elit. Ratione consequuntur totam atque. Totam hic ab numquam ipsam sint vel ratione consectetur aliquam, fugit facere cupiditate quo recusandae tenetur ipsa beatae?

### Titre 1

Lorem ipsum dolor, sit amet consectetur adipisicing elit. Nisi quaerat beatae repellat. Esse, voluptate! Consectetur delectus dolores ullam dolore odit totam eligendi eveniet adipisci unde, labore praesentium aliquam iure nostrum?

# Exemple de style



exemple.html > html > head > meta

```html
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <link rel="stylesheet" href="exemple.css" />
    <title>Exemple css</title>
  </head>
  <body>
    <h1>Chapitre 1</h1>
    <p>
      Lorem ipsum <i>dolor</i> sit amet, consectetur adipisicing elit. Ratione
      consequuntur totam atque. Totam hic ab numquam ipsam sint vel ratione
      consectetur aliquam, fugit facere cupiditate quo recusandae tenetur ipsa
      beatae?
    </p>
    <h2>Titre 1</h2>
    <p>
      Lorem ipsum dolor, sit amet consectetur adipisicing elit. Nisi quaerat
      beatae repellat. Esse, voluptate! Consectetur delectus dolores ullam
      dolore odit totam eligendi eveniet adipisci unde, labore praesentium
      aliquam iure nostrum?
    </p>
  </body>
</html>
```

exemple.css

```css
body {
  font-family: Arial, Verdana, sans-serif;
}
h1,
h2 {
  color: #ee3e80;
}
p {
  color: #665544;
}
```

# Utiliser le CSS externe



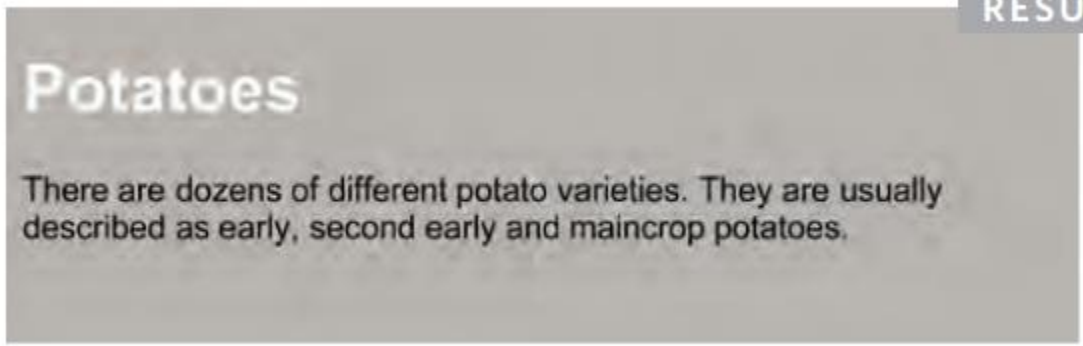
HTML

```
<!DOCTYPE html>
<html>
  <head>
    <title>Using External CSS</title>
    <link href="css/styles.css" type="text/css"
      rel="stylesheet" />
  </head>
  <body>
    <h1>Potatoes</h1>
    <p>There are dozens of different potato
       varieties. They are usually described as
       early, second early and maincrop.</p>
  </body>
</html>
```

CSS

```
body {
    font-family: arial;
    background-color: rgb(185,179,175);}
h1 {
    color: rgb(255,255,255);}
```

RESULT



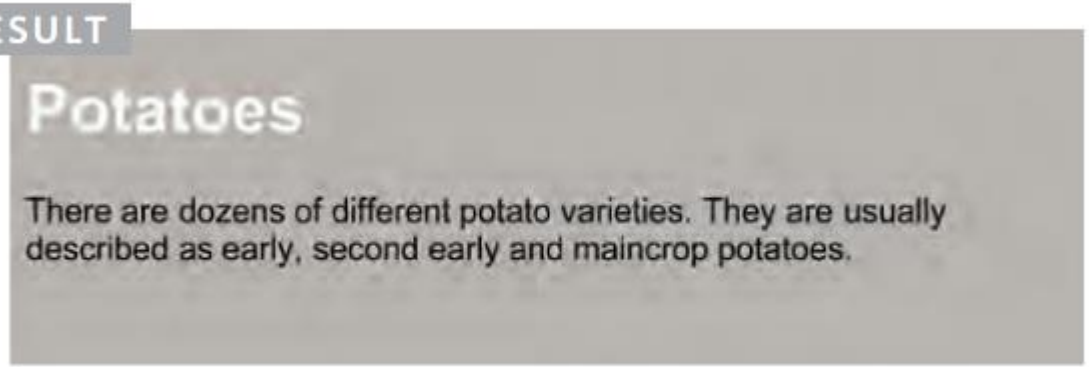
# Utiliser CSS interne



HTML + CSS

```
<!DOCTYPE html>
<html>
  <head>
    <title>Using Internal CSS</title>
    <style type="text/css">
      body {
          font-family: arial;
          background-color: rgb(185,179,175);}
      h1 {
          color: rgb(255,255,255);}
    </style>
  </head>
  <body>
    <h1>Potatoes</h1>
    <p>There are dozens of different potato
       varieties. They are usually described as
       early, second early and maincrop.</p>
  </body>
</html>
```

RESULT



# CSS selectors



| SELECTOR | MEANING | EXAMPLE |
|---|---|---|
| UNIVERSAL SELECTOR | Applies to all elements in the document | `* {}`<br>Targets all elements on the page |
| TYPE SELECTOR | Matches element names | `h1, h2, h3 {}`<br>Targets the <h1>, <h2> and <h3> elements |
| CLASS SELECTOR | Matches an element whose `class` attribute has a value that matches the one specified after the period (or full stop) symbol | `.note {}`<br>Targets any element whose `class` attribute has a value of `note`<br>`p.note {}`<br>Targets only <p> elements whose `class` attribute has a value of `note` |
| ID SELECTOR | Matches an element whose `id` attribute has a value that matches the one specified after the pound or hash symbol | `#introduction {}`<br>Targets the element whose `id` attribute has a value of `introduction` |

# CSS selectors



| | | |
|---|---|---|
| CHILD SELECTOR | Matches an element that is a direct child of another | `li>a {}`<br>Targets any <a> elements that are children of an <li> element (but not other <a> elements in the page) |
| DESCENDANT SELECTOR | Matches an element that is a descendent of another specified element (not just a direct child of that element) | `p a {}`<br>Targets any <a> elements that sit inside a <p> element, even if there are other elements nested between them |
| ADJACENT SIBLING SELECTOR | Matches an element that is the next sibling of another | `h1+p {}`<br>Targets the first <p> element after any <h1> element (but not other <p> elements) |
| GENERAL SIBLING SELECTOR | Matches an element that is a sibling of another, although it does not have to be the directly preceding element | `h1~p {}`<br>If you had two <p> elements that are siblings of an <h1> element, this rule would apply to both |

# Comment les règles css s'appliquent en cascade



HTML

```
<h1>Potatoes</h1>
<p id="intro">There are <i>dozens</i> of different
   <b>potato</b> varieties.</p>
<p>They are usually described as early, second early
   and maincrop potatoes.</p>
```

RESULT

**Potatoes**

There are *dozens* of different potato varieties.

They are usually described as early, second early and maincrop potatoes.

CSS

```
* {
  font-family: Arial, Verdana, sans-serif;}
h1 {
  font-family: "Courier New", monospace;}
i {
  color: green;}
i {
  color: red;}
b {
  color: pink;}
p b {
  color: blue !important;}
p b {
  color: violet;}
p#intro {
  font-size: 100%;}
p {
  font-size: 75%;}
```

1. Last rule

2. More specific

# Héritage de style



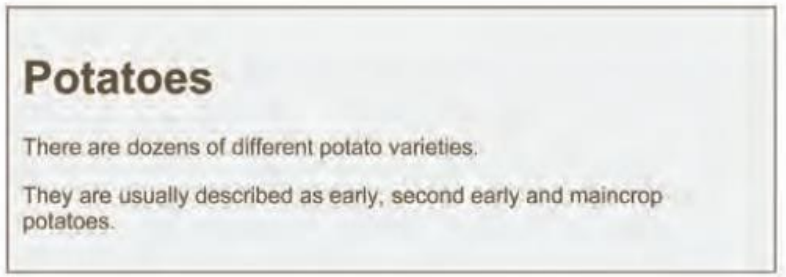
HTML

```
<div class="page">
  <h1>Potatoes</h1>
  <p>There are dozens of different potato
     varieties.</p>
  <p>They are usually described as early, second
     early and maincrop potatoes.</p>
</div>
```

CSS

```
body {
  font-family: Arial, Verdana, sans-serif;
  color: #665544;
  padding: 10px;}
.page {
  border: 1px solid #665544;
  background-color: #efefef;
  padding: inherit;}
```

RESULT



# Couleur

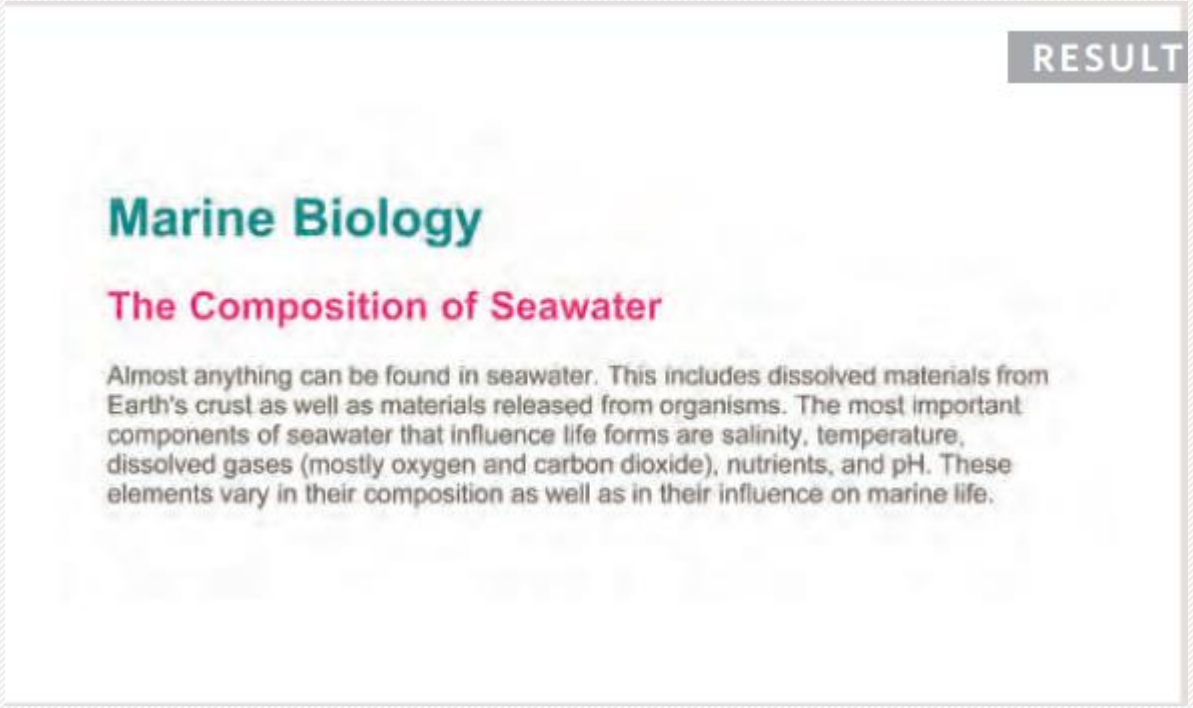
# La propriété color



CSS

```css
/* color name */
h1 {
  color: DarkCyan;}
/* hex code */
h2 {
  color: #ee3e80;}
/* rgb value */
p {
  color: rgb(100,100,90);}
```

RESULT



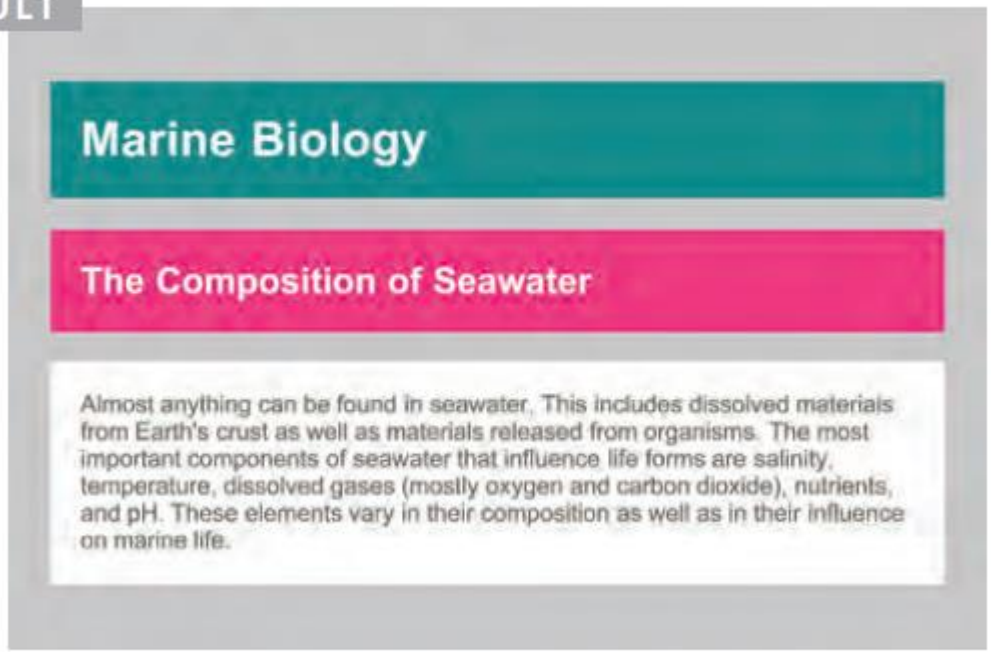
# Background-color



CSS

```
body {
  background-color: rgb(200,200,200);}
h1 {
  background-color: DarkCyan;}
h2 {
  background-color: #ee3e80;}
p {
  background-color: white;}
```

RESULT

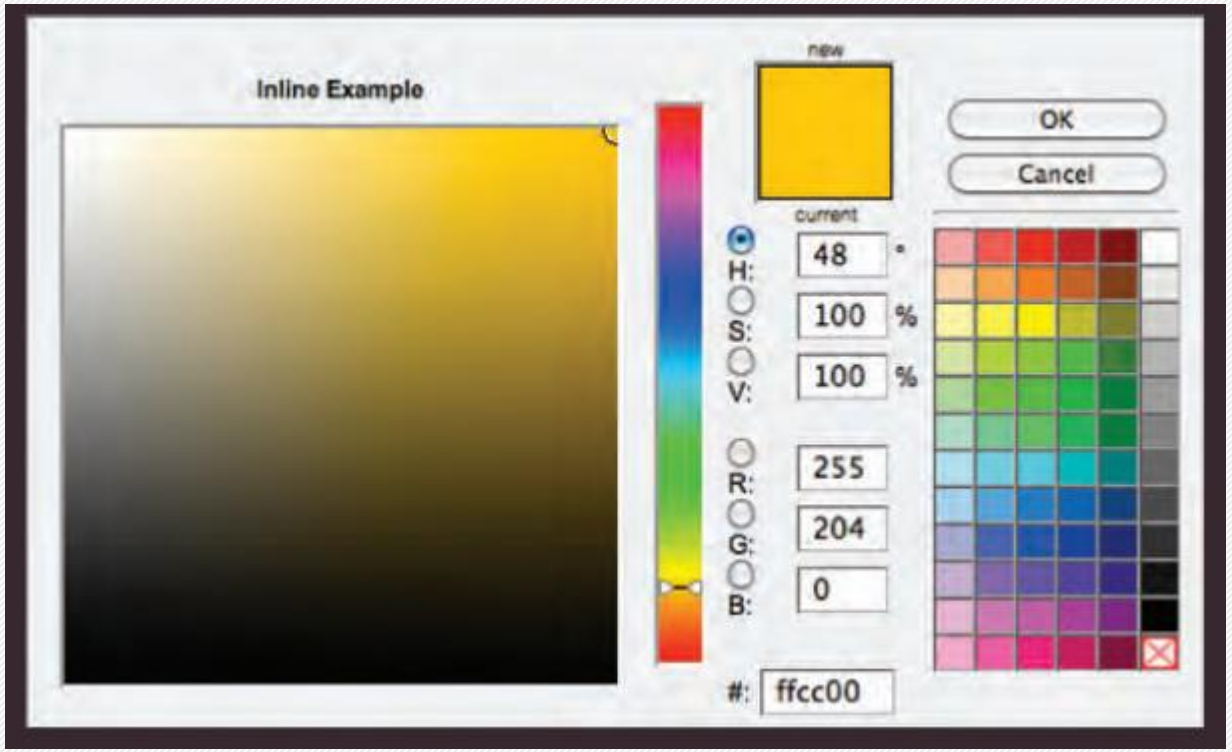


# Comprendre la couleur



# Comprendre la couleur





# Contrast






Text is harder to read when there is low contrast between background and foreground colors.


Text is easier to read when there is higher contrast between background and foreground colors.


For long spans of text, reducing the contrast a little bit improves readability.

# Opacity



CSS

```
p.one {
  background-color: rgb(0,0,0);
  opacity: 0.5;}
p.two {
  background-color: rgb(0,0,0);
  background-color: rgba(0,0,0,0.5);}
```

RESULT

# HSL color

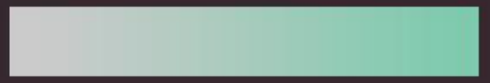


## HUE

Hue is the colloquial idea of color. In HSL colors, hue is often represented as a color circle where the angle represents the color, although it may also be shown as a slider with values from 0 to 360.

## SATURATION

Saturation is the amount of gray in a color. Saturation is represented as a percentage. 100% is full saturation and 0% is a shade of gray.



## LIGHTNESS

Lightness is the amount of white (lightness) or black (darkness) in a color. Lightness is represented as a percentage. 0% lightness is black, 100% lightness is white, and 50% lightness is normal. Lightness is sometimes referred to as *luminosity*.

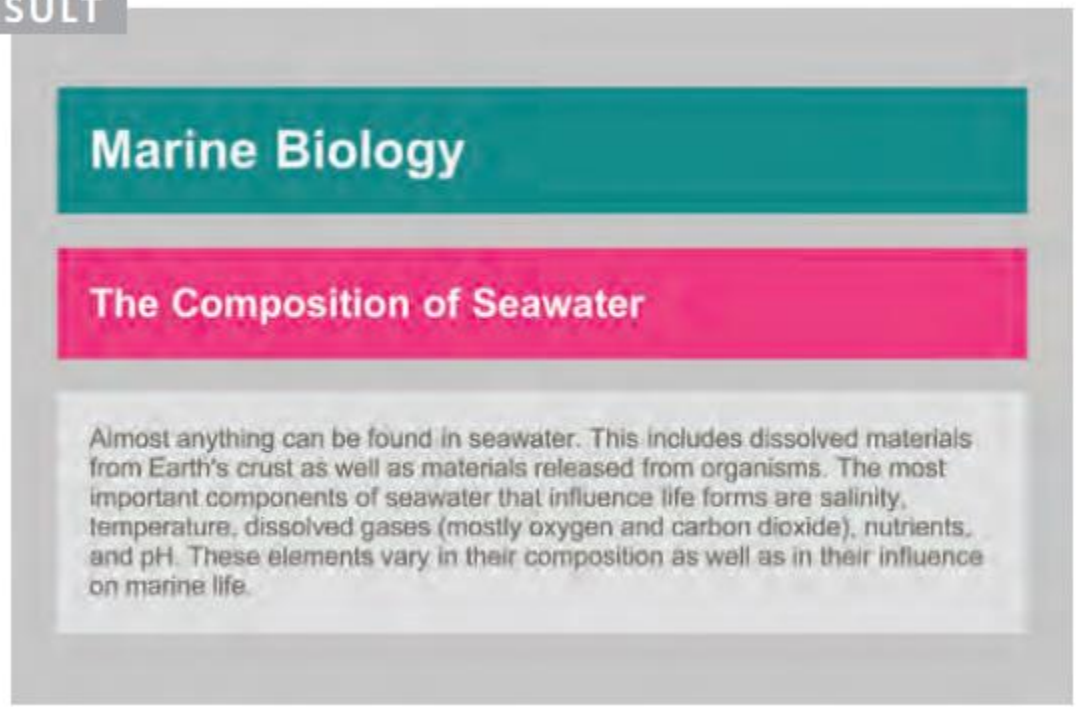
# HSL et HSLA



CSS

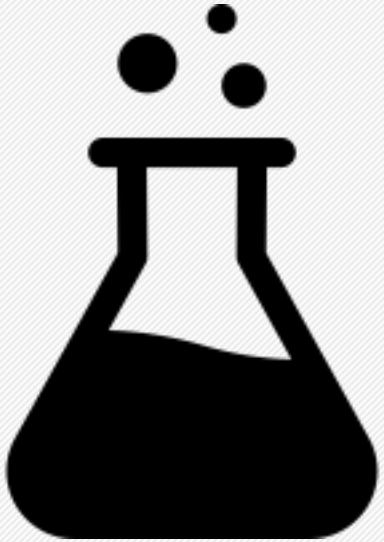
```
body {
  background-color: #C8C8C8;
  background-color: hsl(0,0%,78%);}
p {
  background-color: #ffffff;
  background-color: hsla(0,100%,100%,0.5);}
```

RESULT



# Exercice d’application





TP4.pdf sur BlendLearn