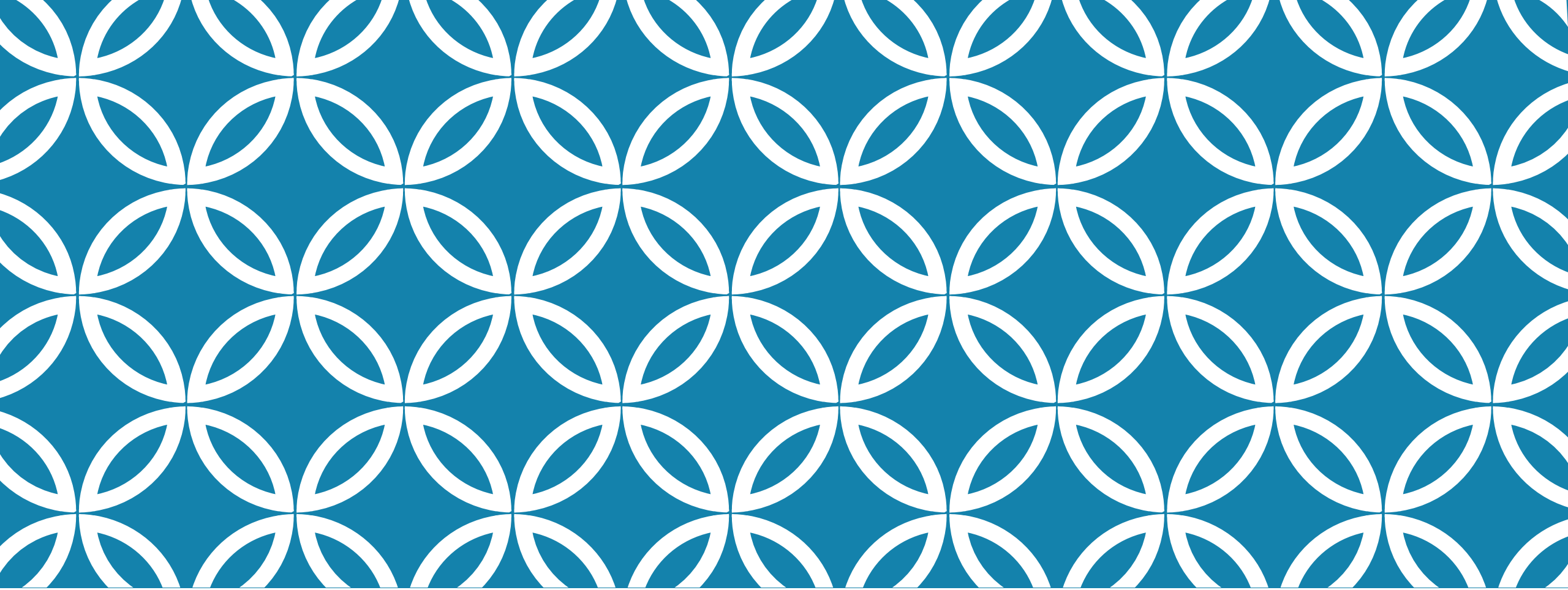


ensia
The National School of
Artificial Intelligence
المدرسة الوطنية للذكاء الاصطناعي

CSS

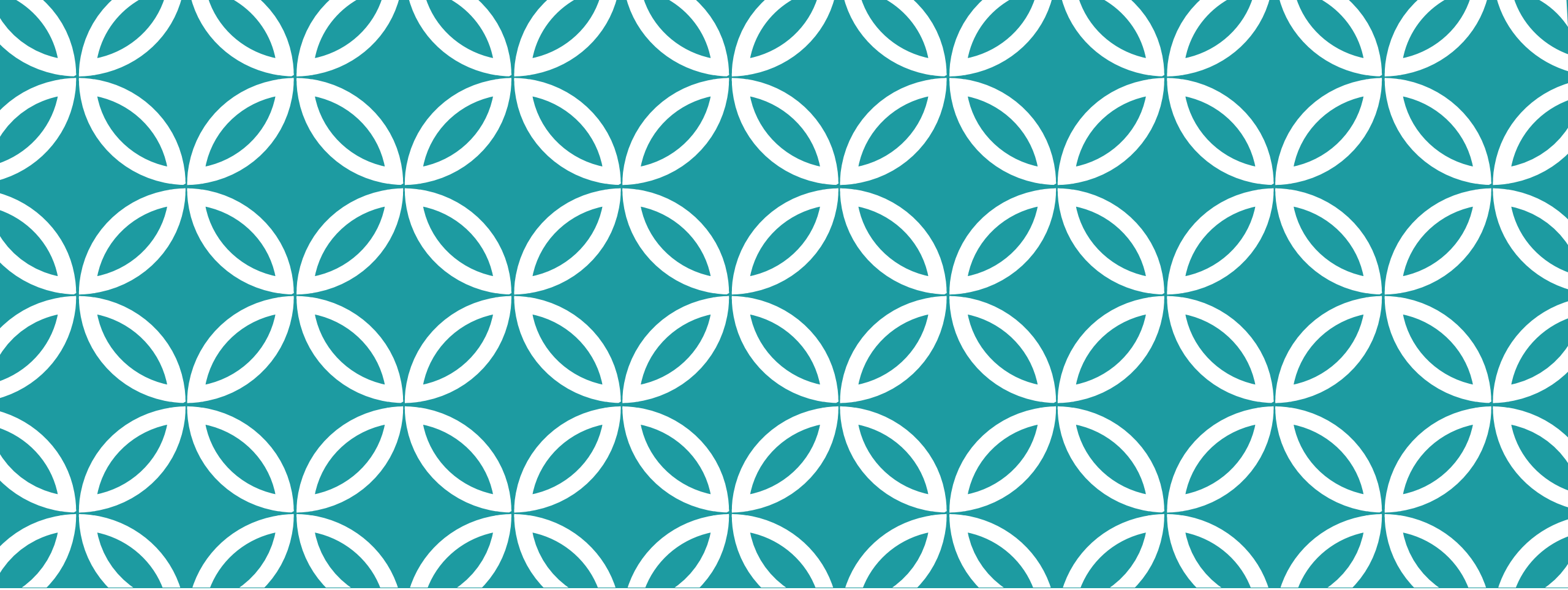
Dr. Amir DJOUAMA
amir.djouama@ensia.edu.dz

PLAN

General information on CSS

Selectors

Properties



GENERAL INFORMATION ON CSS

CASCADING STYLE SHEET

Formatting

Separation of form and content

HTML describes the content

CSS describes the form

Centralization of the visual aspect

CSS is inserted between :

```
<style type="text/css">
<!-- CSS here -->
</style>
```

Or we link a CSS file with :

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

Comments : between /* and */

CSS CLASS FORMAT

CSS is a set of classes. A class is written as follows:

```
selector {
    property1 : value1 ;
    property2 : value2 ;
    .
    .
    .
}
```

CSS CLASS FORMAT

`selector` can be :

- a tag name: the properties apply to all these tags
- a generic name (starting with a dot): the attributes apply to tags using `class= »selector"` (without the point)

The properties designate the modified elements (colour, border, background, margins, etc.)

The values indicate which values are to replace the designated properties.

CSS EXAMPLE

style.css

```
body { font-family : Arial ; }
p { background-color : #F0C0C0 ; border : thin solid black ; }
.titre { color : yellow ; }
```

CSS EXAMPLE

Index.html

```
<html>
<head>
	<link rel="stylesheet" type="text/css" href="style.css"/>
	<title>CSS Example</title>
</head>
<body>
	<h1 class="titre">Title in yellow</h1>
	<p>A paragraph with a border and backgroung</p>
</body>
</html>
```

UNITS OF LENGTH

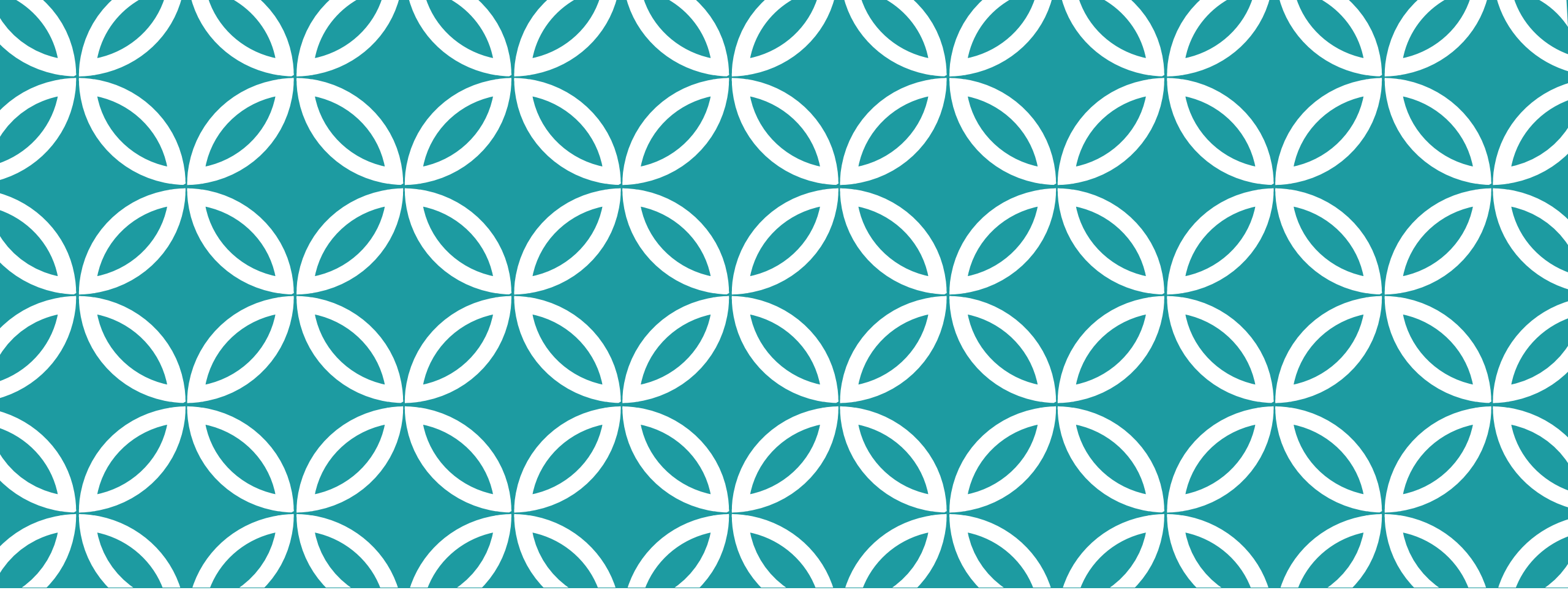
`px`	:	pixel
`em`	:	m-length
`ex`	:	x-height
`in`	:	inch either 25,4 mm
`cm`	:	centimeter
`mm`	:	millimeter
`pt`	:	point either 1/72 in
`pc`	:	pica either 12 points , 1/6 in

COLORS

keywords : `black, blue, brown, cyan, gray, green, pink, purple, red, …`

Hexa code : `#999999`, RGB component

fonction rgb() : `rgb(r,g,b)` with $0 \leq r,g,b \leq 255$



SELECTORS

SELECTORS

One element

- `p { color : yellow ; background-color : blue ; }`

Several elements

- `h1, div, p { color : yellow ; background-color : blue ; }`

The universal selector

- `* { background-color : blue ; }`

CLASSES

```
.red { color : red ; }
```

```
.red { color : red ; }
div.red { color : yellow ; }
```

```
.red { color : red ; }
div.red { background-color : green ; }
```

SEVERAL CLASSES TO THE SAME ELEMENT

```
*.yellow { color : yellow ; }
div.yellow { color : green ; }
.class1 { color : red ; }
.class2 { font-style : italic ; }
.class3 { background-color : blue ; }
```

SEVERAL CLASSES TO THE SAME ELEMENT

Example

```
<h1 class = "yellow"> Title in yellow </h1>

<div class = "class1"> Texte en red </div>

<div class = "class1 class2"> Text in red and in italic
</div>

<div class = "class1 class3"> Text in red and blue
background </div>

<div class = "yellow class2 classe3"> Text in green and
in italic with blue background </div>
```

ID SELECTOR

```
div { color : black ; }

#blue { color : white ; background-color : blue; }


<div> Text in black </div>

<div id = "blue"> Text in white with blue background
</div>
```

ATTRIBUTE SELECTORS

```
* [title] { background-color : yellow ; }
h2 [title] [id] { background-color : yellow ; }
a[href$=".dz"] {color: red;}
```

EXAMPLE

The following example will apply a red background to all links that have a `title` attribute.

```
a[title] { background-color: red; }
```

The following example will apply a red background to all links that have a `rel` attribute with the value `external`.

```
a[rel=external] { background-color: red; }
```

EXAMPLE

/* All spans with the "lang" attribute are in bold */

- span[lang] {font-weight:bold;}

/* All spans in ar are in green */

- span[lang="ar"] {color:green;}

/* All spans in American English are in blue */

- span[lang="en-US"] {color: blue;}

EXAMPLE

/* All internal links have an orange background colour */

- a[href^="#"] {background-color:orange}

/* All links whose urls end in ".dz" are in red */

- a[href$=".dz"] {color: red;}

/* All links containing "example" in the url have a grey background */

- a[href*="example"] {background-color: #CCCCCC;}

CONTEXTUAL PARENT-DESCENDANT SELECTORS

```
Parent-element descendant-element { definition of the
style ; }

ul li { background-color : red ; color : blue ; }

Parent-element > child-element { définition du style ;
}
```

EXAMPLE

p h2 {color: green}

" h2s that are in a p "

p > h2 {font-size: 30pt}

- " h2 that are directly in a p "

ADJACENT ELEMENT SELECTORS

```
element1 + element2 { style def ; }
h1 + p { … }
```

As an example, this rule specifies that a P element that follows a MATH element should not have an indent:

```
MATH + P { text-indent: 0 }
```

The following produces a reduction in the vertical space between an `H1` element and the `H2` element immediately following it:

```
H1 + H2 { margin-top: -5mm }
```

Here, the rule is similar to the previous one, with the difference that it includes an additional attribute selector. Thus, a particular formatting applies to `H1` elements with a `class="opener"` attribute:

```
H1.opener + H2 { margin-top: -5mm }
```

PSEUDO-CLASSES APPLICABLE TO LINKS

`a:[pseudo-class]`

- `:visited`
- `:focus`
- `:hover`
- `:active`
- `:link`

PSEUDO-CLASSES APPLICABLE TO LINKS

Example

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

a:focus{
    background-color: green;
}
```

LES PSEUDOCLASSES ET PSEUDO-ÉLÉMENTS

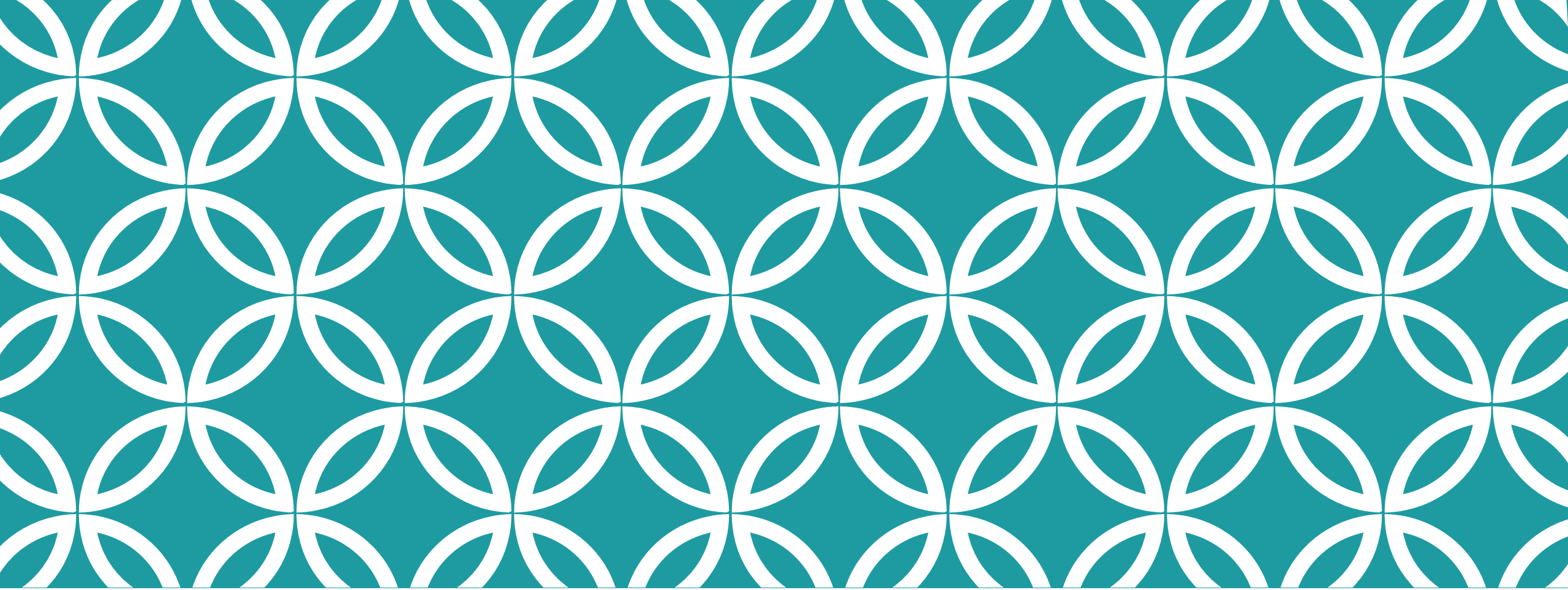
```
::first-child
::last-child
::first-letter
::first-line
::before { content : "before" ; style def ; }
::after { content : "after" ; style def ; }
```

!IMPORTANT DECLARATION

Managing conflict

```
* {color : black !important ; background-color : yellow
; }

div {color : blue ; background-color : white ; }
```



PROPERTIES

PROPERTIES (1)

```
color : valeur ; : foreground colour
background-color : valeur ; : background colour
background-image : url(URL) ; : background image
```

PROPERTIES (2)

border-style : style{1,4} ;

style can take the following values:

- none : no border
- hidden : idem except table cell
- dotted : short tips
- dashed : long dashes
- solid : continue
- double : 2 continuous parallel lines
- groove : recessed border
- ridge : embossed border
- inset : recessed border where each rib has only one colour
- outset : raised border with only one colour on each side

PROPERTIES (3)

1, 2, 3 or 4 values can be specified

- 1 : 4 sides
- 2: the first applies to the top and bottom sides, the second applies to the right and left sides
- 3: the first applies to the high sides, the second applies to the right and left sides, the third applies to the low sides
- 4: top, right, bottom, left

PROPERTIES (4)

border-width : width{1,4} ;

width can take the following values :

- thin | medium |thick
- Numeric value

1, 2 or 4 values can be specified

- 1 : 4 sides
- 2: the first applies to the top and bottom sides, the second applies to the right and left sides
- 3: the first applies to the high sides, the second applies to the right and left sides, the third applies to the low sides
- 4: top, right, bottom, left

PROPERTIES (5)

```
border : width style color ;
```

```
h1 { border : 5px double blue ; }
```

is equivalent to

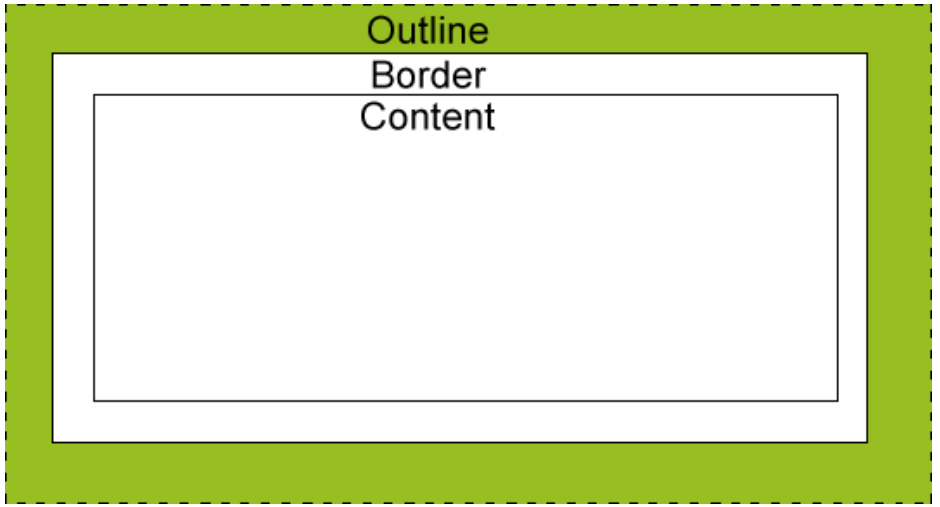
```
h1 { border-width : 5px ; border-style : double ;
border-color: blue; }
```

PROPERTIES (6)

```
margin : value {1,4} ;
padding : value {1,4} ;
```

PROPERTIES (7)

```
outline-style : valeur {1,4} ;
outline-width : valeur {1,4} ;
outline-color : valeur {1,4} ;
outline : width color style ;
```



PROPERTIES (8)

```
font-family : "value" ;

font-size : absolute value | relative value | value |
pourcent value

    absolute value : xx-small, x-small, small, medium,
large, x-large, xx-large

    relative value : smaller, larger

font-weight : normal | bold | bolder | lighter | x00 ; :
```
bold with x integer : $1 \leq x \leq 9$

```
font-style : normal | italic ;

font-variant : normal | small-caps ;
```

PROPERTIES (9)

```
text-transform : none | uppercase | lowercase | capitalize;

text-decoration : none | underline | overline | line-through;

line-height : normal | valeur | pourcent ;

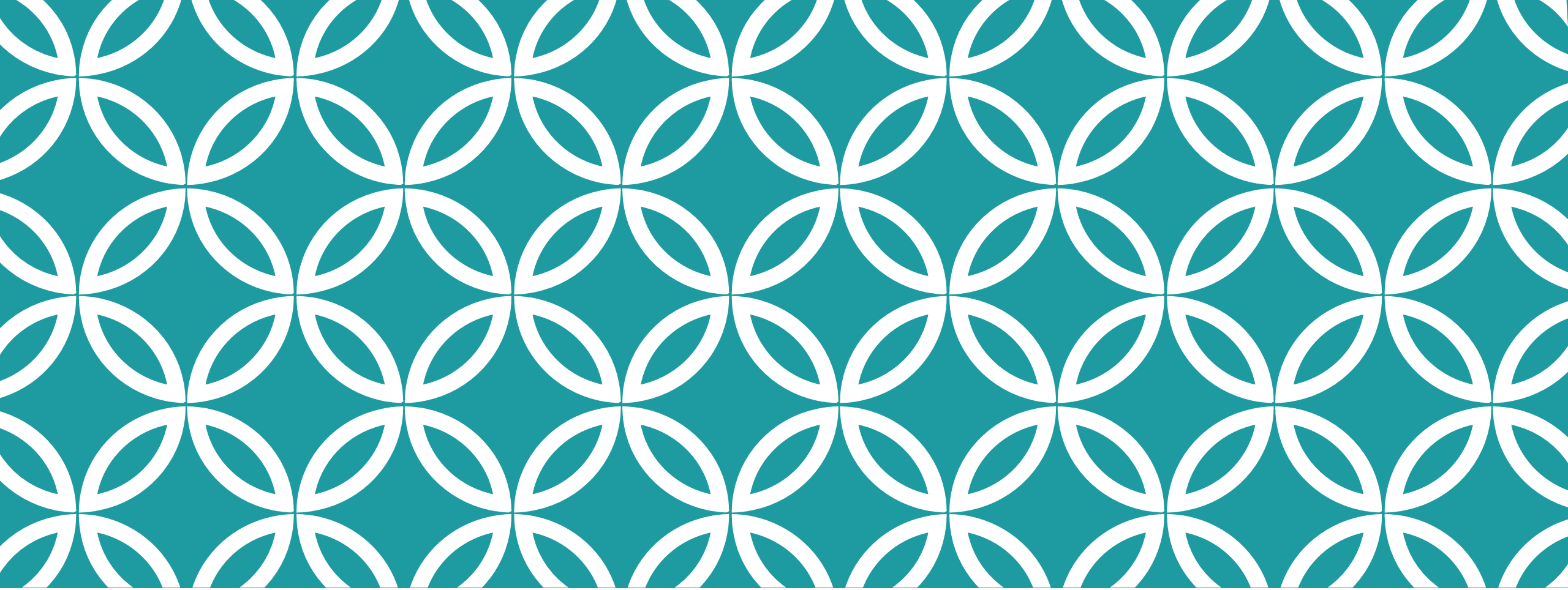
font : style variant weight [size/line-height family] ;
```

LES PROPRIÉTÉS (10)

```
text-align : left | center | right | justify ;

letter-spacing : normal | valeur ;

word-spacing : normal | valeur ;
```



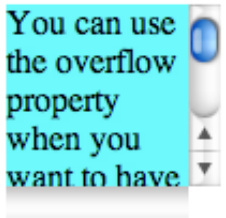
DIMENSIONING AND POSITIONING

DIMENSIONING AND POSITIONING

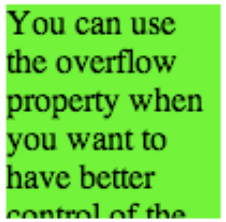
The dimensioning of the elements

- `width : <value> | pourcent | auto | inherit ;`
- `height : <value> | pourcent | auto | inherit ;`
- `overflow : visible | hidden | scroll | auto | inherit ;`
  - `visible` : the overflowing content is displayed
  - `hidden` : the overflowing content is hidden
  - `scroll` : systematic lifts even without overflow
  - `auto` : lifts in case of overflow
- `min-heigth : <value> | <pourcent> | inherit ;`
- `max-heigth : <value> | <pourcent> | none | inherit ;`
- `min-width : <value> | <pourcent> | inherit ;`
- `max-width : <value> |< pourcent> | none | inherit ;`

overflow:scroll



overflow:hidden



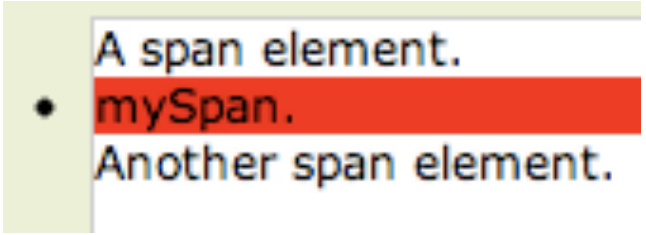
THE RENDERING OF THE ELEMENTS

`display : none | inline | block | list-item | table | inline-table I none`

- `inline` : on a line
- `block` : bloc (like <h1>, <p>, , ...)
- `list-item` : list (like <li>)

Example

```
span#mySpan { background-color:red; display:list-item; }
```



THE POSITIONING OF THE ELEMENTS

Floating

- `float : left | right | none | inherit`

Prevent floating for block elements

- `clear : none | left | right | both | inherit`
- `none` : floating allowed
- `left` : left float prohibited
- `right` : right float prohibited
- `both` : left right float prohibited

EXEMPLE

testCSSClearfix.html

RELATIVE POSITIONING

```
position : relative
```

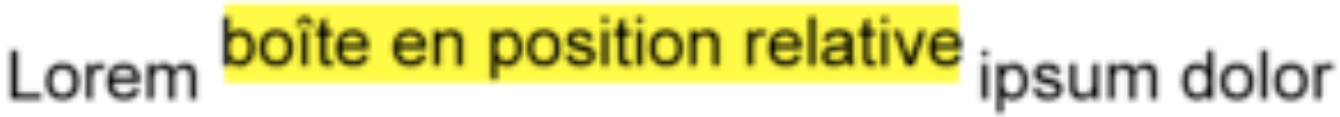
with

- `left : <value> | <pourcent> | auto | inherit`
- `top : <value> | <pourcent> | auto | inherit`
- `right : <value> | <pourcent> | auto | inherit`
- `bottom : <value> | <pourcent> | auto | inherit`
  - where `<value>` is positive or negative

EXAMPLES

```
<p>
  Lorem
  <span class="yellow">
    boîte en position relative
  </span>
  ipsum dolor.
</p>
```

```
.yellow {
  position: relative;
  bottom: 5px;
  background-color: #ffff00;
}
```



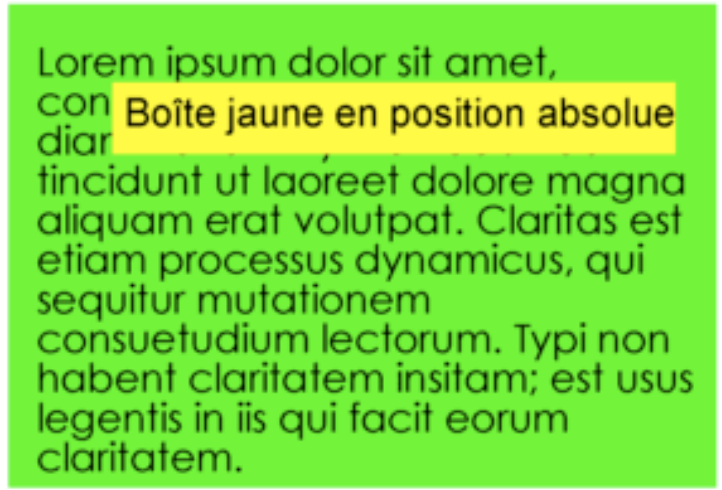
ABSOLUTE POSITIONING

```
position : absolute
```

EXAMPLES

```
.green {
  position: relative;
  background-color: #00ff00;
  width: 15em;
}
.yellow {
  position: absolute;
  top: 1em;
  right: 1em;
  background-color: #ffff00;
}
```

```
<div class="green">
  <p>
    …
  </p>
  <p class="yellow">
    Boîte jaune en position absolue
  </p>
</div>
```



EXAMPLE

```
div.frame1 { position : absolute ;
    border : thin solid black ;
    left : 0px ;
    top : 0px ;
    width : 100% ;
    height : 20% ; }

div.frame2 { position : absolute ;
    border : thin solid black ;
    left : 0px ;
    top : 20% ;
    width : 20% ;
    height : 100% ; }
```

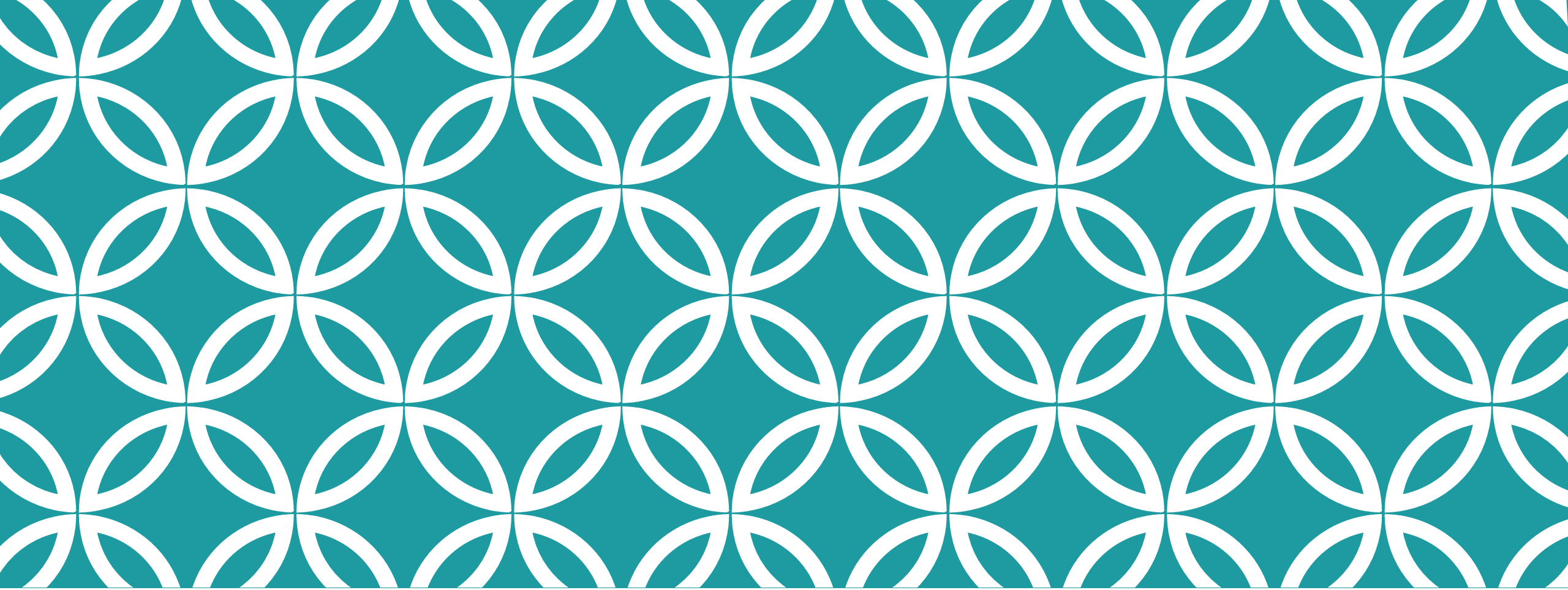
```
div.frame3 { position : absolute ;
    border : thin solid black ;
    left : 20% ;
    top : 20% ;
    width : 80% ;
    height : 80% ; }
```

LE POSITIONNEMENT FIXE

```
position : fixed
```

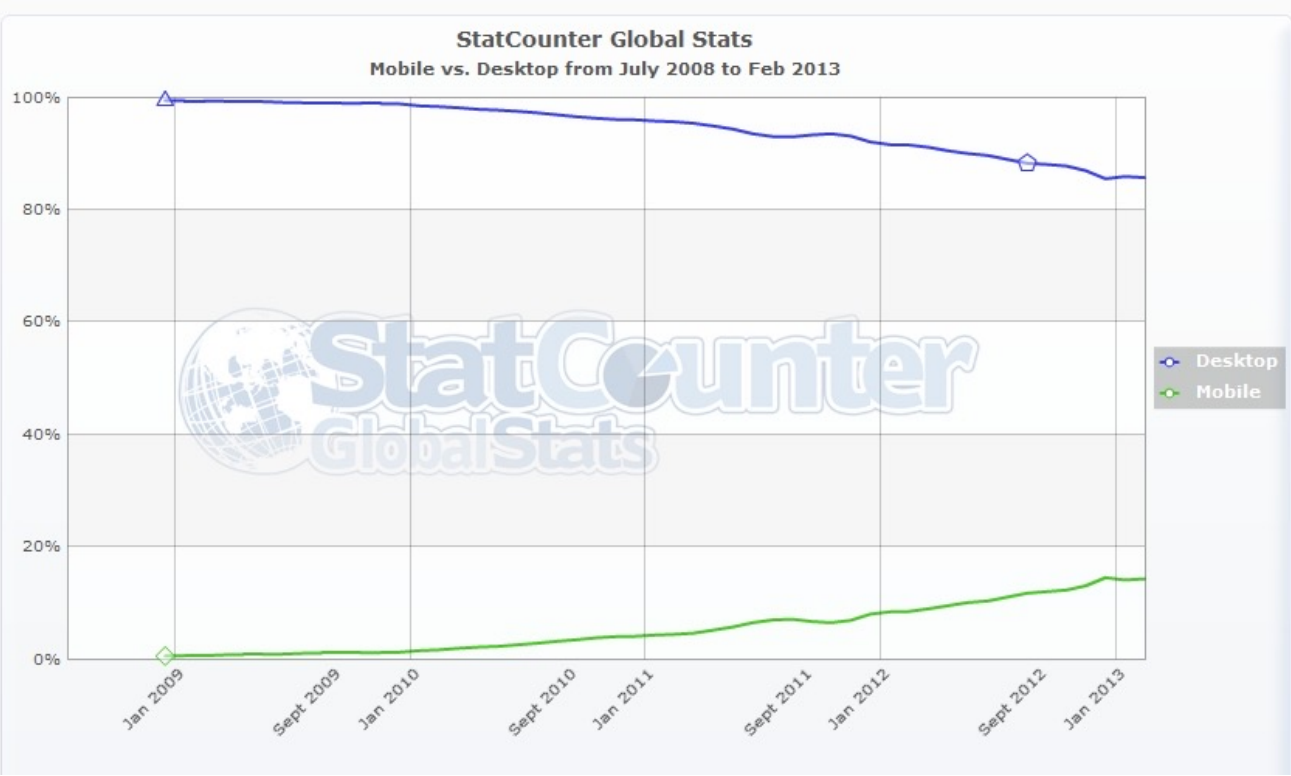
Special case of absolute positioning

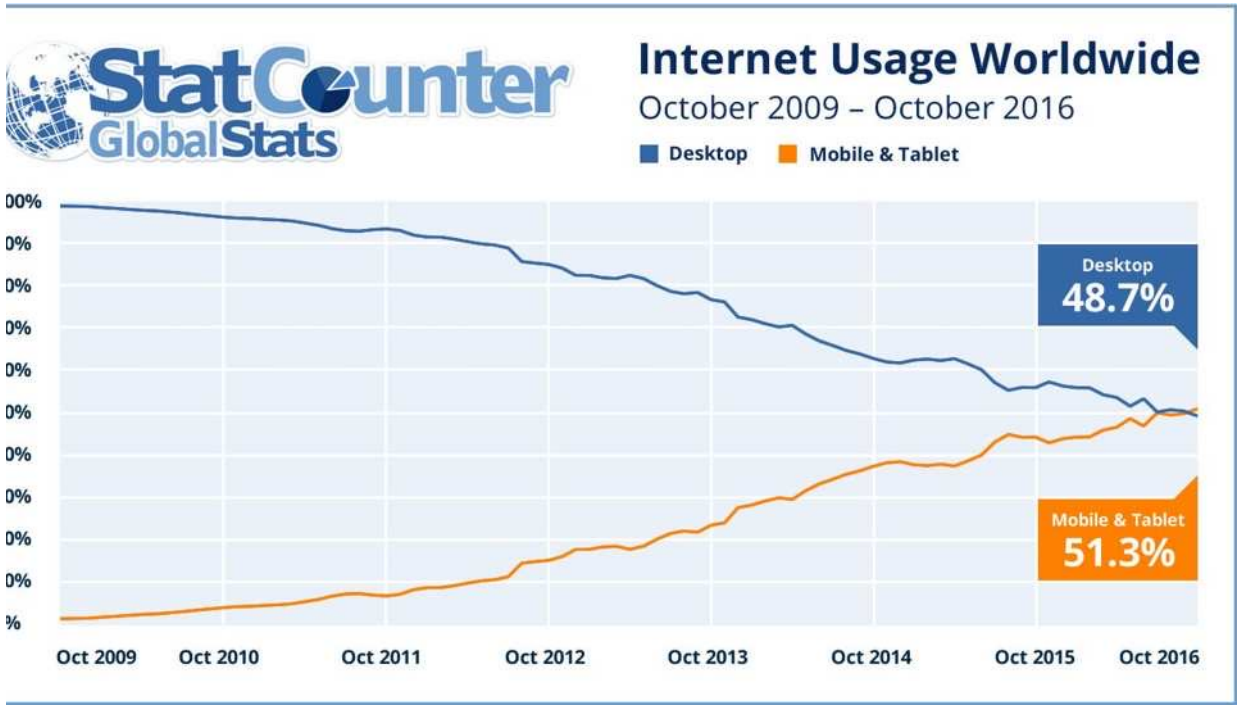
The container is not the parent element but the browser window



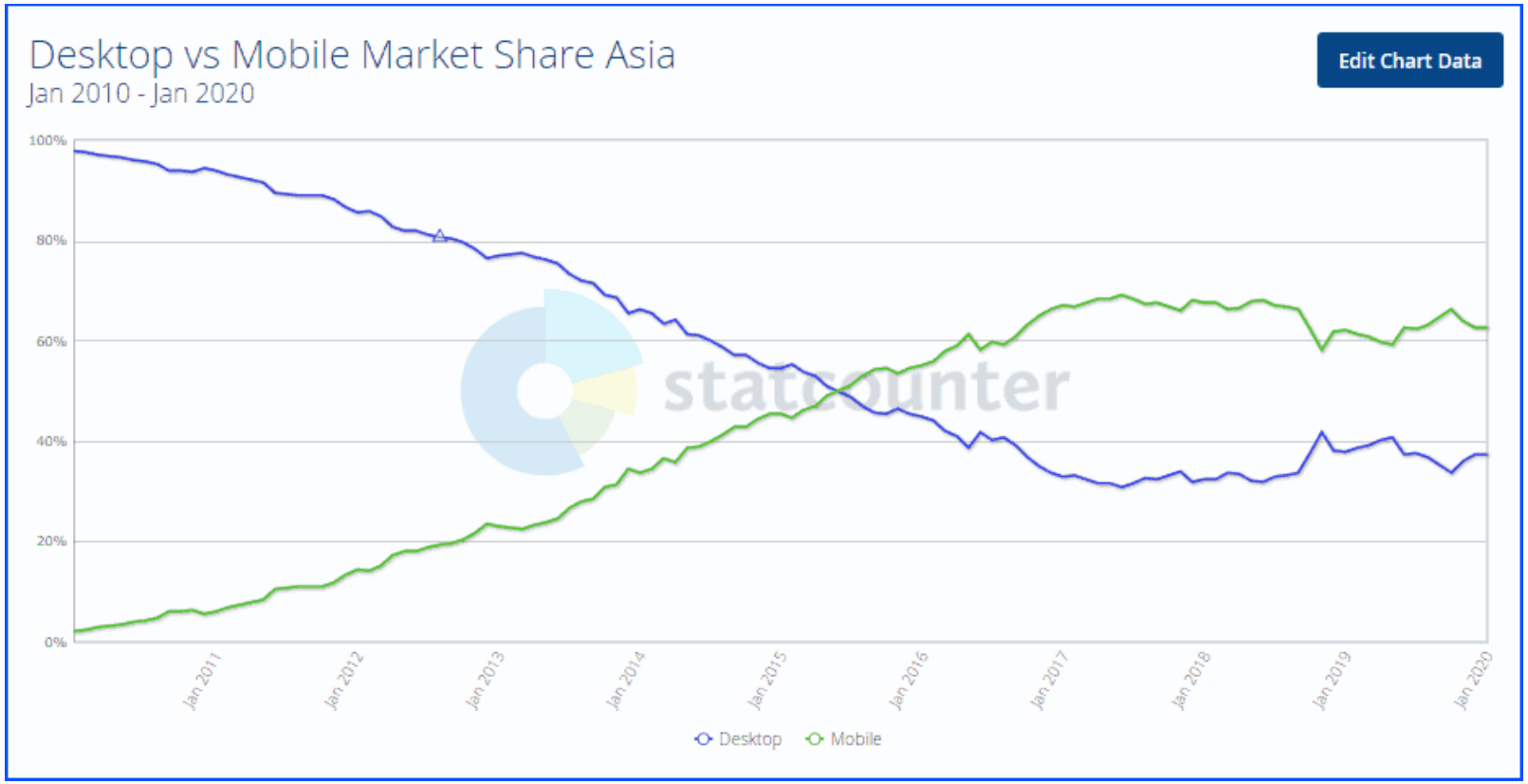
RESPONSIVE DESIGN

WHY?


StatCounter Global Stats
Mobile vs. Desktop from July 2008 to Feb 2013
100%
80%
60%
40%
20%
0%
Jan 2009
Sept 2009
Jan 2010
Sept 2010
Jan 2011
Sept 2011
Jan 2012
Sept 2012
Jan 2013
Desktop
Mobile
StatCounter
GlobalStats


StatCounter
GlobalStats
Internet Usage Worldwide
October 2009 – October 2016
Desktop
Mobile & Tablet
Desktop
48.7%
Mobile & Tablet
51.3%
Oct 2009
Oct 2010
Oct 2011
Oct 2012
Oct 2013
Oct 2014
Oct 2015
Oct 2016

POURQUOI?


Desktop vs Mobile Market Share Asia
Jan 2010 - Jan 2020
Edit Chart Data
100%
80%
60%
40%
20%
0%
Jan 2011
Jan 2012
Jan 2013
Jan 2014
Jan 2015
Jan 2016
Jan 2017
Jan 2018
Jan 2019
Jan 2020
Desktop
Mobile
statcounter

WHAT IS IT?


FIONA HUTCHINSON
Home
Biography
Stockists
Exhibitions
My Collections
Shop Now
Contact
My Blog
Beautiful Handmade Sterling Silver & Gemstones Jewellery
MY COLLECTIONS

PRINCIPLE

Make a single website that adapts to all screen formats

The width of the screen is used to determine the display of the elements

HOW ?

Using CSS media queries

Allows you to adapt the CSS to the width of the screen

Exp :

```
@media screen and (max-width: 480px)
{
    nav
    {     width: 100%; }
}
```

Don't forget the viewports

- To avoid browsers distorting their screen widths
- `<meta name="viewport" content="width=device-width" />`

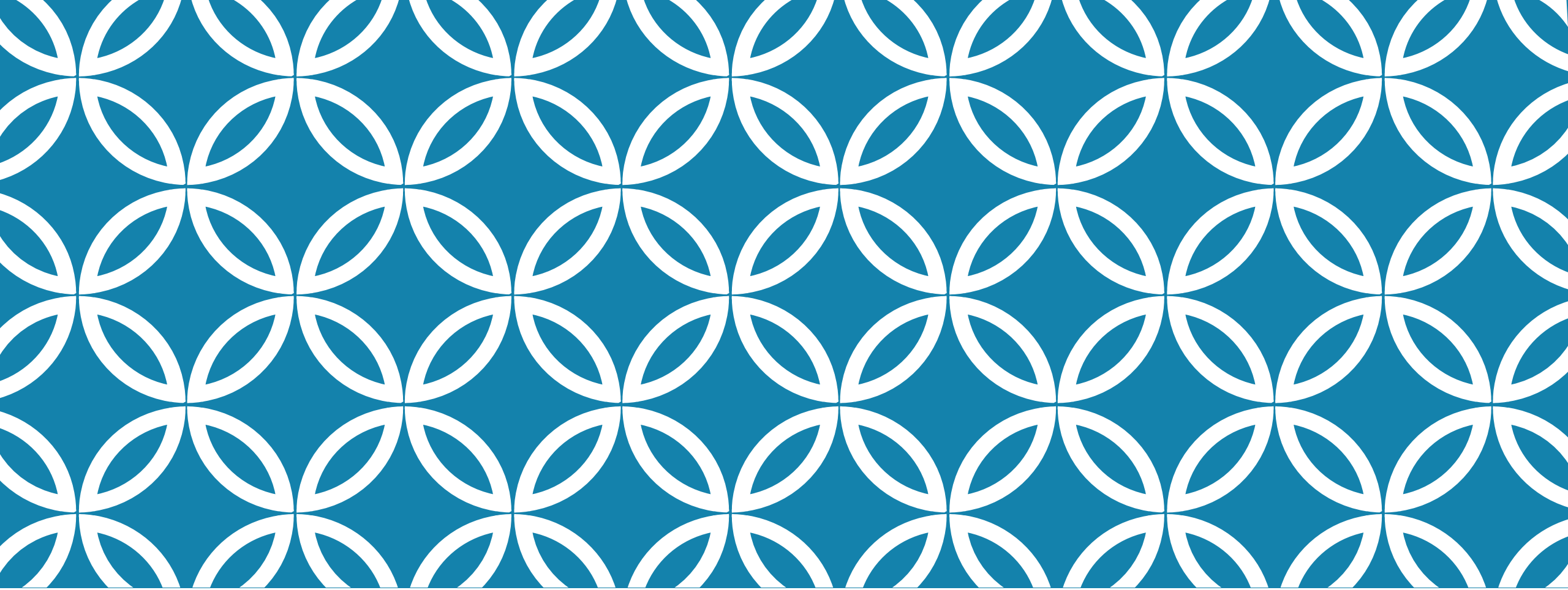
HOW ?

Example:

- In the tag
  - `<link rel="stylesheet" media="screen and (max-width: 1280px)" href= "fichier1.css" />`
- In the CSS file
  - `@media all and (min-width: 1024px) and (max-width: 1280px)`
  - `@media all and (orientation: portrait)`

In media we find:

- `Screen`
- `Handheld`
- `Print`
- `Tv`
- `Projection`
- `All`



ensia
The National School of
Artificial Intelligence
المدرسة الوطنية للذكاء الاصطناعي

CSS3

PRÉFIXE NAVIGATEUR (PRÉFIXE VENDEURS)

New CSS3 features defined by W3C

Not yet standardized by browsers

Browsers have developed their own CSS3 rules

- They were used in the W3C recommendations

- Android | Chrome | iOS | Safari : -webkit-
- Firefox: -moz-
- Opera: -o-

BORDERS

CSS 2.1 defines border styles

- Does not meet designers' expectations
- The solution was to place images in the background

`Border-radius`

- For rounding off the edges
- Used in most new browsers
- Values in pixels
- Can define the four separate corners
- `Border-radius: tl tr br bl`
- The separate radii can be defined as
- `Border-radius : x / y`

BORDERS

`Border-image`

- Intended to put an image on the border
- `Border-image: url(chemin_img) x repeat`
  - `X` : value in % to define the position where the image should be cut
  - `Repeat`: to repeat image

SHADING

`box-shadow | text-shadow`

- Allows you to add a shadow to a box or text
- Two mandatory parameters
- They indicate the right and bottom shifts
- Third option to change the blur of the shading
- Optional fourth to manage colour
- `box-shadow: 10px 15px 25px #4CD;`
- `text-shadow: 3px 5px 3px #4FF;`

TRANSPARENCE ET OPACITÉ

`rgba(...)`

- Used to define an opacity
- `rgba(r,g,b,O)`

`opacity`

- Reel value <= 1
- `opacity: 0.5;`

We can also manage the transparency of the shading

- `Box-shadow: 10px 10px 10px rgba(222,126,0,0.6)`

MULTIPLE BACKGROUNDS

`Background-image`: an essential tool in CSS2.1

- Limited to one image

`Background-image: url(...) url(...) url(...) ...`

- The funds will overlap (with their own sizes ...)

GRADIENTS

Only a solid background colour could be used,

Simple gradient

```
background : linear-gradient(top|left, color1 X%, color2 Y%, … colorN Z%)
```

- Exp:
- `background-image : linear-gradient(left, white, black)`

GRADIENTS

Angular gradient

Not limited to vertical or horizontal (top | left)

```
background : linear-gradient(val deg, color1 X%, color2 Y%, … colorN Z%)
```

- Exp:
- `background-image : linear-gradient(30 deg, white, black)`

GRADIENTS

Radial gradient

```
background : linear-gradient(start, form, color1 X%, color2 Y%, … colorN Z%)
```

Start: `top, middle, bottom et left, center, right`

Forme:

- Composed of the keywords `circle` and `ellipse`
- `Cover:` occupe toute la surface du conteneur
- `Closest-side:` si `circle`, le dégradé est circulaire et s'arrête sur le côté le plus proche, si `ellipse`, il s'étend sur toute la surface,
- `Closest-corder:` le dégradé s'étend jusqu'au coin le plus proche de son conteneur

LES DÉGRADÉS

- `Farthest-side` : le dégradé s'étend jusqu'au coté le plus éloigné de son conteneur
- `Farthest-corner` : le dégradé s'étend jusqu'au coin le plus éloigné de son conteneur
- `Contain` : la forme circulaire ou elliptique est entièrement contenue dans le conteneur
- Exp:
- ```
  background-image: radial-gradient(center top, circle cover, white, #111);
  ```
  ```

LES DÉGRADÉS

Autres

- Dégradé répétitifs
- `Background : repeating-linear-gradient (top|left, couleurs …)`
- `Background : repeating-linear-gradient (départ, forme, couleurs …)`

- Dégradé RGBA
- RGBA à la place des couleurs

LES TRANSFORMATIONS

Permettent de faire tourner, changer l'échelle ou tordre un élément

Transformation 2D

- `transform : rotate(x deg);`
- `transform : scale(x); //x représente %`

Transformation 3D

- `rotate3d(x,y,z,angle)`
- `rotatex(deg)`
- `rotatey(deg)`
- `rotatez(deg)`
- `perspective: nombre`

LES TRANSITIONS

Effectue une transition dans les transformations

`transition-property: prop1, prop2 … propn`

- Liste des propriétés sur lesquels on applique la transition

`transition-duration : N sec`

`transition-timing-function: valeur`

- Type de fonction de transition
- `linear` : vitesse constante
- `ease-in` : la vitesse augmente
- `ease-out` : la vitesse diminue
- `ease-in-out` : la vitesse est lente au début et à la fin

LES TRANSITIONS

- `transition-delay` : **temps d'attente avant de commencer**

Possible de tout mettre sur une seule ligne

- `Transition: prop durée fonction délai`

Exp

```
nav li a {
        transition-property: background;
        transition-duration: 1s;
        transition-timing-function: ease-out;
        transition-delay: 0s;
}
```

LES ANIMATIONS

`animation-name`: nom de l'animation

`animation-duration`: durée de l'animation

`animation-iteration-count`: nombre de fois où l'animation est réalisée

`animation-timing-function`: type de fonction d'animation

- `linear | ease-in | ease-out | ease-in-out`

`animation-delay`: délai d'attente

LES ANIMATIONS

Une fois la règle terminée, on défini l'animation

```
@keyframes -nom_animation {
        from {
                définition de la position de départ
        }
        X% {
                définition de la position à X% de l'animation
        }
        to {
                définition de la position d'arrivée
        }
}
```

EXEMPLES

```
 /* The animation code */
@keyframes example {
    from {background-color: red;}
    to {background-color: yellow;}
}

/* The element to apply the animation to */
div {
    width: 100px;
    height: 100px;
    background-color: red;
    animation-name: example;
    animation-duration: 4s;
}
```

EXEMPLES

```
 /* The animation code */
@keyframes example {
    0%   {background-color: red;}
    25%  {background-color: yellow;}
    50%  {background-color: blue;}
    100% {background-color: green;}
}

/* The element to apply the animation to */
div {
    width: 100px;
    height: 100px;
    background-color: red;
    animation-name: example;
    animation-duration: 4s;
}
```

EXEMPLES

```
 /* The animation code */
@keyframes example {
    0%   {background-color: red; left:0px; top:0px;}
    25%  {background-color: yellow; left:200px; top:0px;}
    50%  {background-color: blue; left:200px; top:200px;}
    75%  {background-color: green; left:0px; top:200px;}
    100% {background-color: red; left:0px; top:0px;}
}

/* The element to apply the animation to */
div {
    width: 100px;
    height: 100px;
    position: relative;
    background-color: red;
    animation-name: example;
    animation-duration: 4s;
}
```