Web Responsive Design

websites on all devices

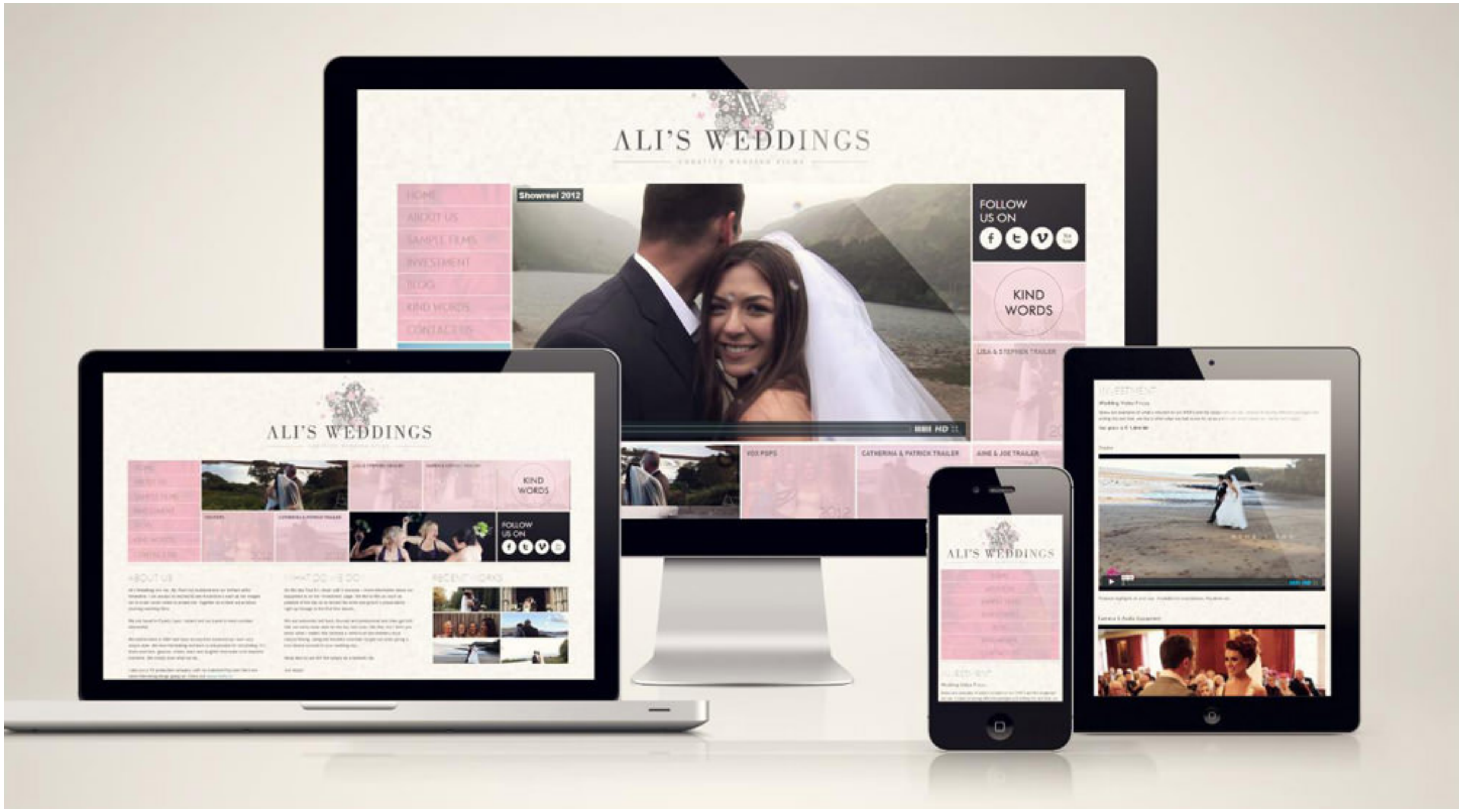
Responsive design

Responsive web design is an approach aimed at sites to provide a **optimal vision** and an **easy-to-read experience** on any device.

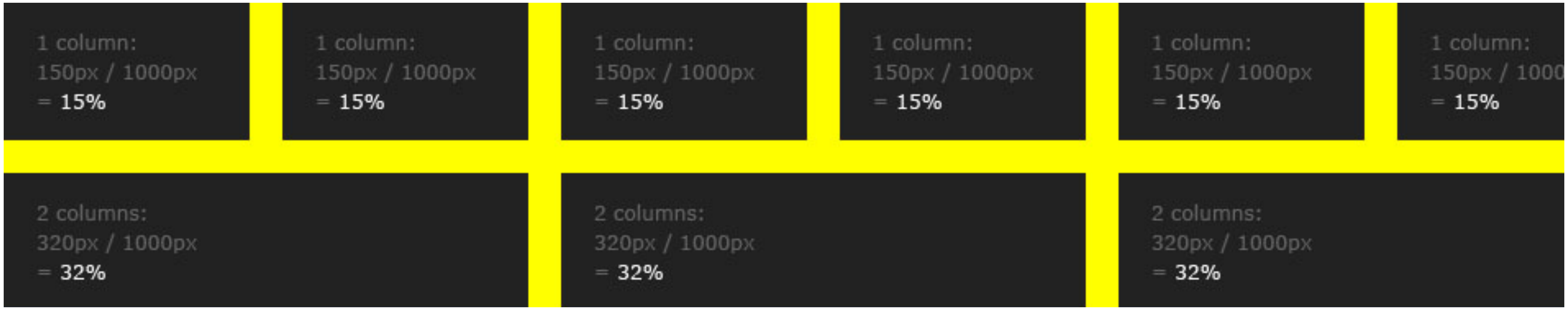
Responsive design is based on **fluid layouts** with grids proportional to the content, **flexible images** and various **media query** commands.

Responsive design

This is also one of the reasons why the role of the Digital Designer is in great demand right now.
Responsive design is a technique that refers to the basics of layout, using structural logic schemes and graphic cages.



grids



Examples

https://radicalicons.com/

https://ilo.so/

http://zomigi.com/blog/hiding-and-revealing-portions-of-images/

Media Query

Thanks to media queries we can define different types of layouts of our page. Normally the structure and graphics of a website refers exclusively to its CSS. Media queries do not actually change the content but only the CSS defining them according to a decisive measure.

stili_1024.css

color: red
font-size: 14

stili_960.css

color: red
font-size: 18

stili_420.css

color: yellow
font-size: 24

Media Query

```
@media screen and (max-width:
1024px) {

.header {
color: red;
font-size: 14pt;
}

}
```

```
@media screen and (max-width:
960px) {

.header {
color: red;
font-size: 18pt;
}

}
```

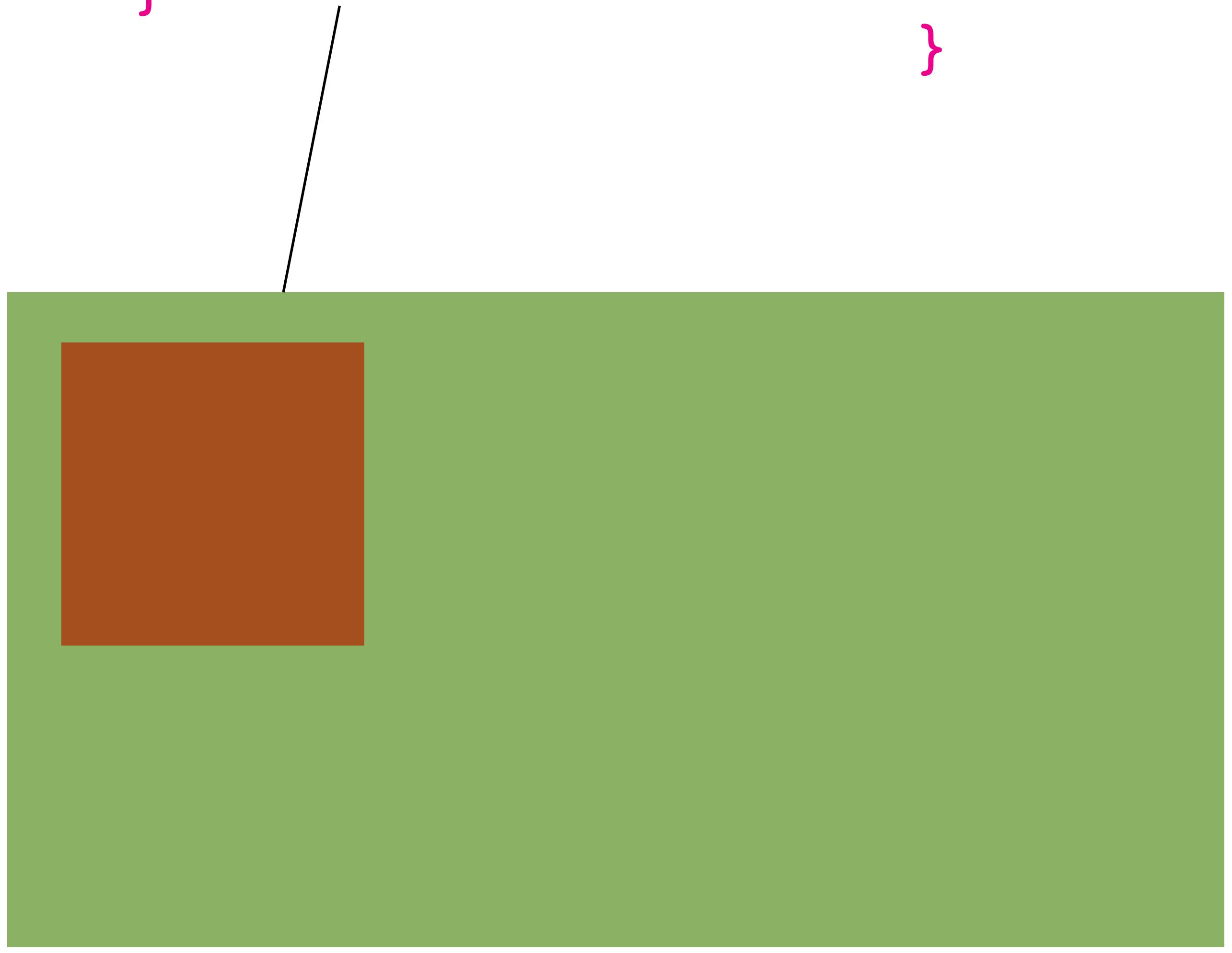
```
@media screen and (max-width:
420px) {

.header {
color: yellow;
font-size: 24pt;
}
}
```

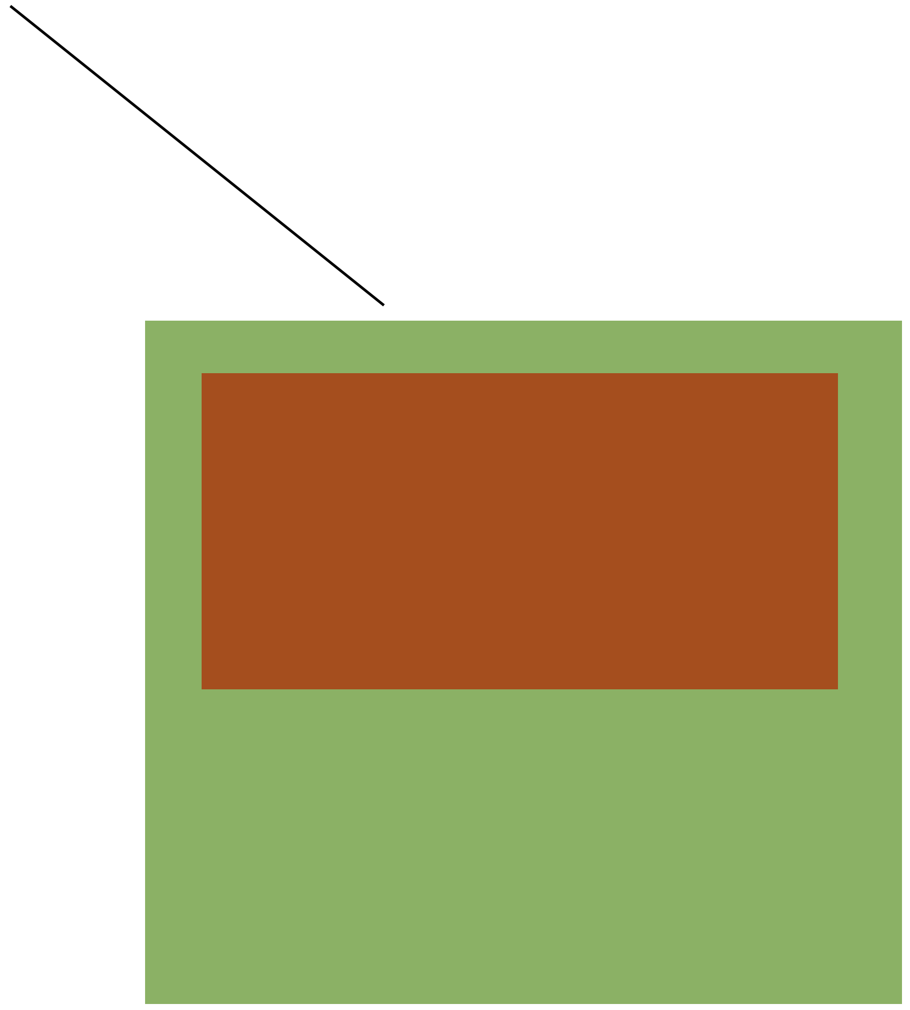
Media Query

Media queries work automatically once the reference code is entered, they are additional rules that are “replaced” on the previous ones.

```
.box_1{
   width: 30%;
}
```

```
 @media screen and (max-width: 960px){
.box_1{
   width: 80%;
}
}
```



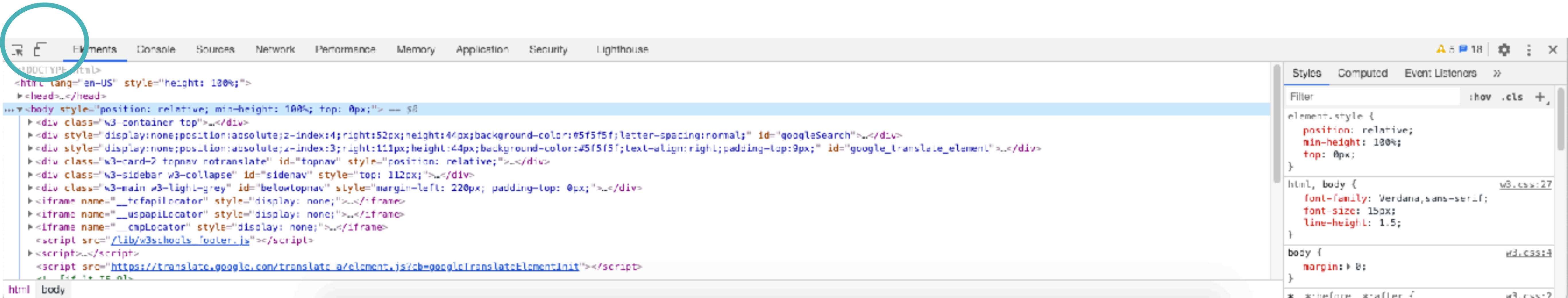


To be inserted in HTML

For a correct functioning it is necessary to insert (in the head) this tag which defines the adaptation to the various devices

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

Ora è possibile visualizzare il nostro lavoro in modalità smartphone.



```
<!DOCTYPE html>
<html lang="en-US" style="height: 100%;">
 ▶<head>…</head>
…▼<body style="position: relative; min-height: 100%; top: 0px;"> == $0
   ▶<div class="w3-container top">…</div>
   ▶<div style="display:none;position:absolute;z-index:4;right:52px;height:44px;background-color:#5f5f5f;letter-spacing:normal;" id="googleSearch">…</div>
   ▶<div style="display:none;position:absolute;z-index:3;right:111px;height:44px;background-color:#5f5f5f;text-align:right;padding-top:9px;" id="google_translate_element">…</div>
   ▶<div class="w3-card-2 topnav notranslate" id="topnav" style="position: relative;">…</div>
   ▶<div class="w3-sidebar w3-collapse" id="sidenav" style="top: 112px;">…</div>
   ▶<div class="w3-main w3-light-grey" id="belowtopnav" style="margin-left: 220px; padding-top: 0px;">…</div>
   ▶<iframe name="__tcfapiLocator" style="display: none;">…</iframe>
   ▶<iframe name="__uspapiLocator" style="display: none;">…</iframe>
   ▶<iframe name="__cmpLocator" style="display: none;">…</iframe>
    <script src="/lib/w3schools_footer.js"></script>
   ▶<script>…</script>
    <script src="https://translate.google.com/translate_a/element.js?cb=googleTranslateElementInit"></script>
```

```
element.style {
   position: relative;
   min-height: 100%;
   top: 0px;
}
html, body {                                w3.css:27
   font-family: Verdana,sans-serif;
   font-size: 15px;
   line-height: 1.5;
}
body {                                      w3.css:4
   margin: ▶ 0;
}
```

Contents

With responsive design it is possible to **hide certain content**. We need to think in what area our user is viewing the website, under what circumstances and with what device.

Example
http://colly.com/archives/

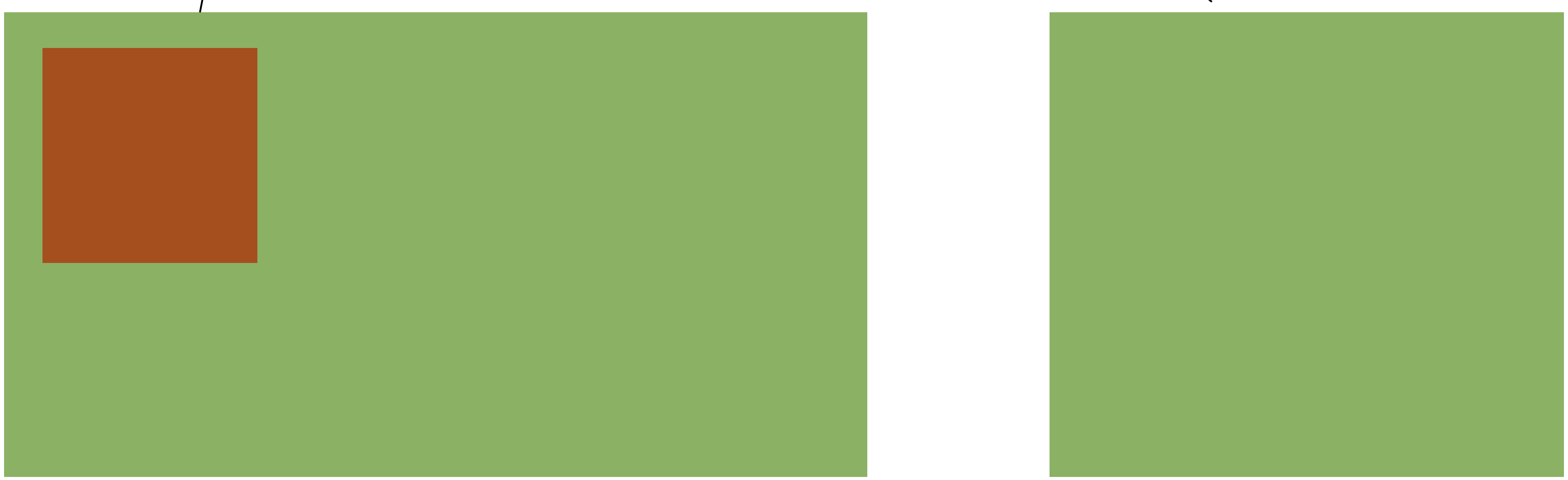
The omission of some content is essential to communicate the main information, so the user will be satisfied to get what he wants in a short time.

Media Query - hide

To delete a content you need the display property with the respective values: block and none.

```
.box_1{
   display: block;
}
```

```
 @media screen and (max-width: 960px){
.box_1{
   display: none;
}
}
```

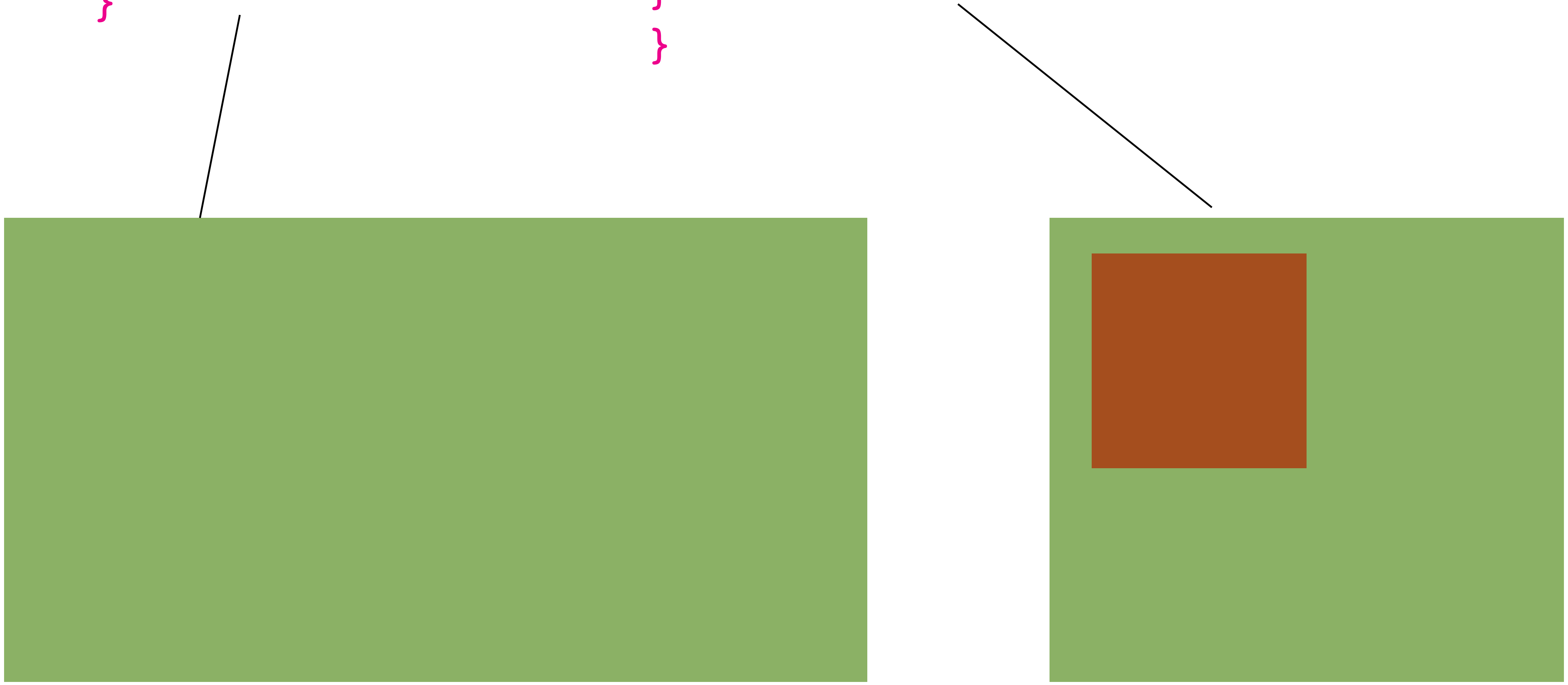


Media Query - show

In the opposite case just reverse the properties.

```
.box_1{
   display: none;
}
```

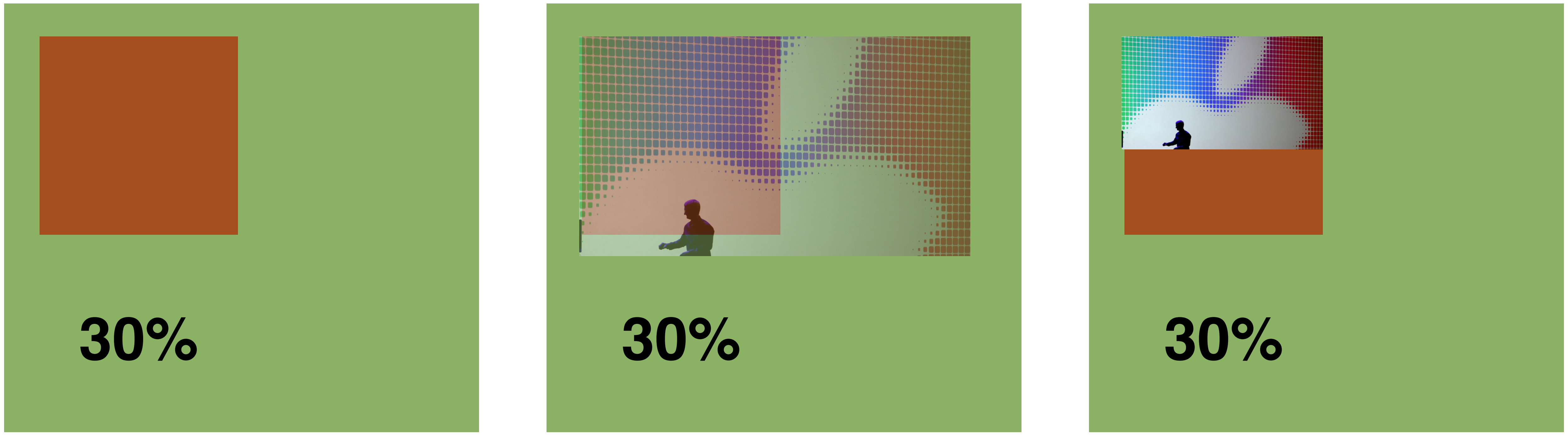
```
 @media screen and (max-width: 960px){
.box_1{
   display: block;
}
}
```



Flexible images

As the images are also content, they will have to adapt to the layout, so flexible images are introduced to do this.

To create a flexible image you need to set up a fluid grid, the latter will change its width.
The image will follow the bending of the grid and then “adapt” to the layout.

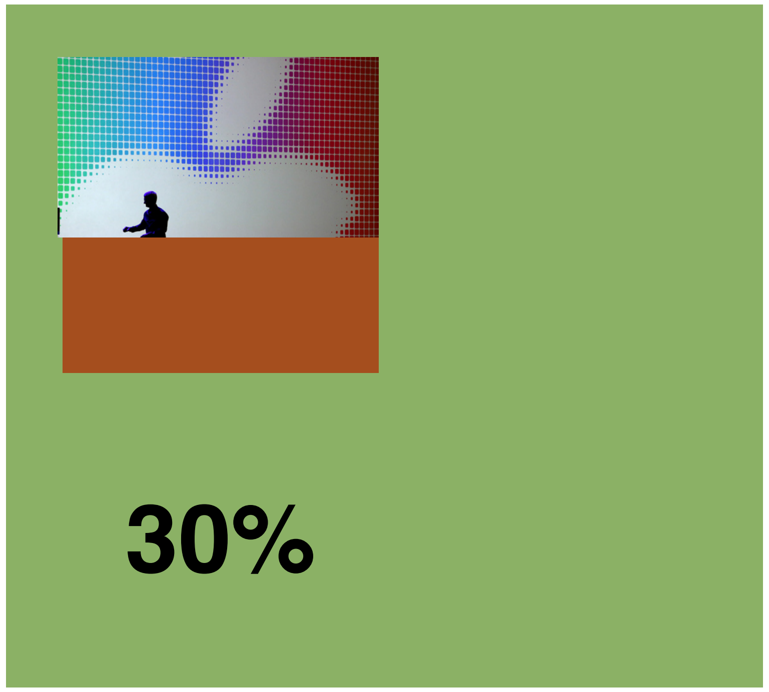


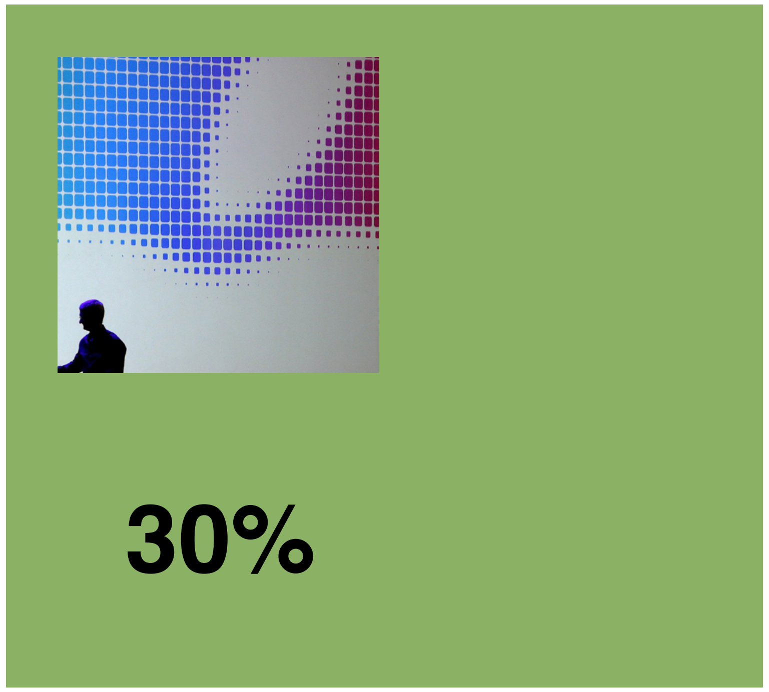
Flexible images

Two methods for two different needs:
width:100% to inserti images via tag img in html.
background-size: cover for background images

```
.image{
   width: 100%;
   height:auto;
}
```

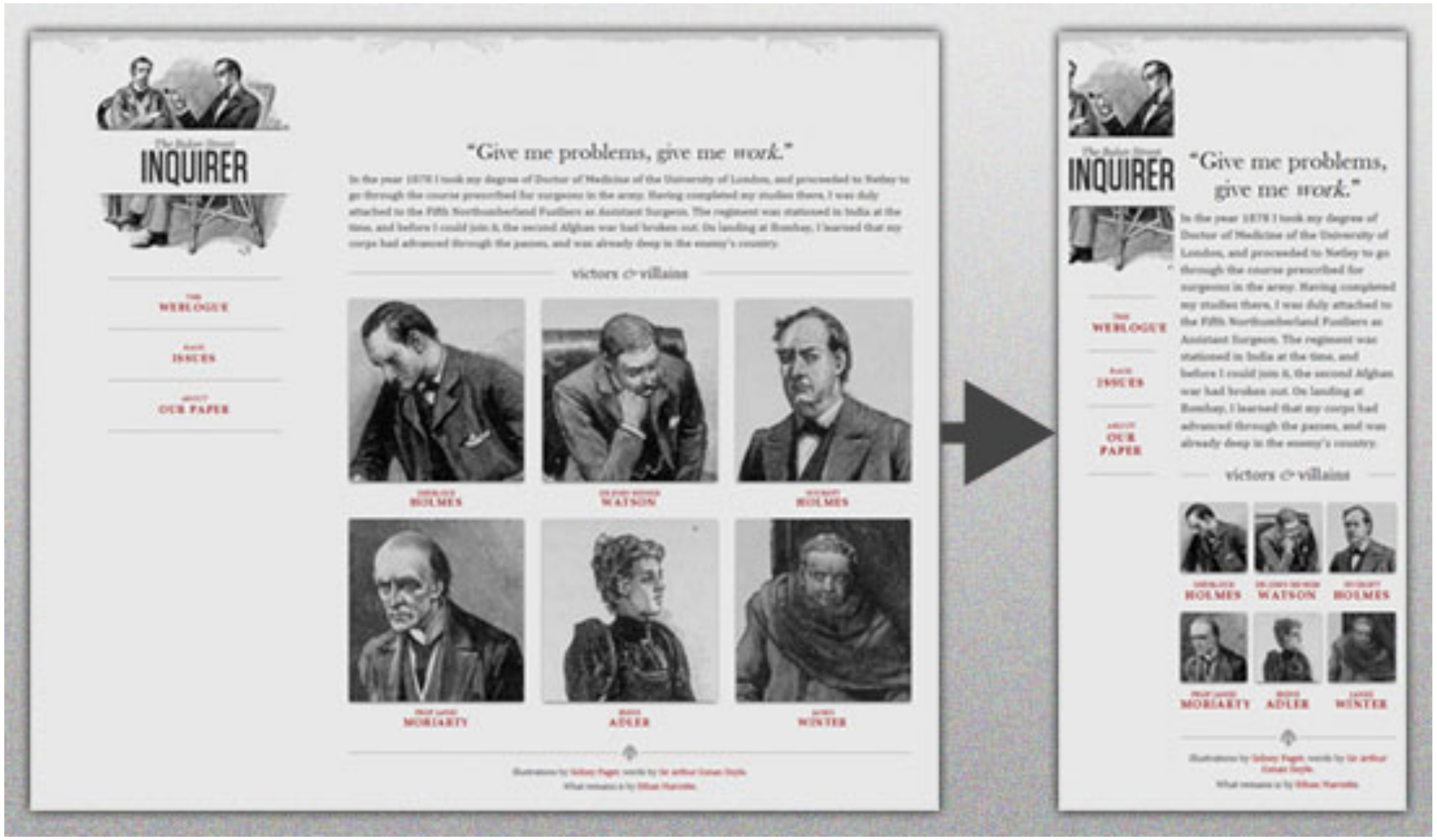
```
.background{
   background-size: cover;
   background-image: sfondo.jpg;
}
```





images

In order to make images “flexible” we can attribute them different techniques: among them the reduction in %, the reduction through media query or the simple “cropping”.

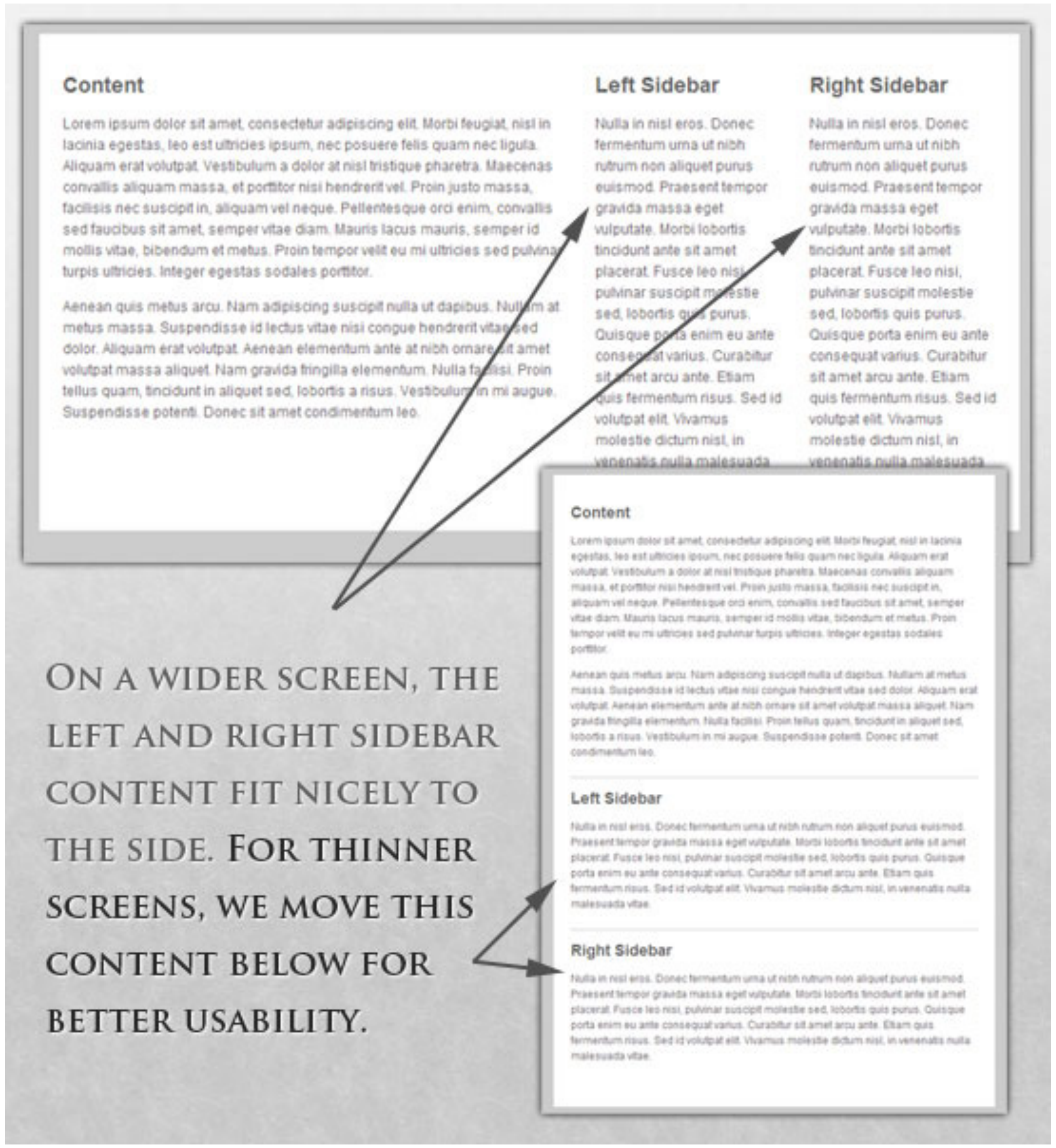


“Give me problems, give me work.”

In the year 1878 I took my degree of Doctor of Medicine of the University of London, and proceeded to Netley to go through the course prescribed for surgeons in the army. Having completed my studies there, I was duly attached to the Fifth Northumberland Fusiliers as Assistant Surgeon. The regiment was stationed in India at the time, and before I could join it, the second Afghan war had broken out. On landing at Bombay, I learned that my corps had advanced through the passes, and was already deep in the enemy's country.

Illustrations by Sidney Paget, words by Sir Arthur Conan Doyle.
What remains is by Ethan Marcotte.

texts


Content
Left Sidebar
Right Sidebar
On a wider screen, the left and right sidebar content fit nicely to the side. For thinner screens, we move this content below for better usability.
Content
Left Sidebar
Right Sidebar

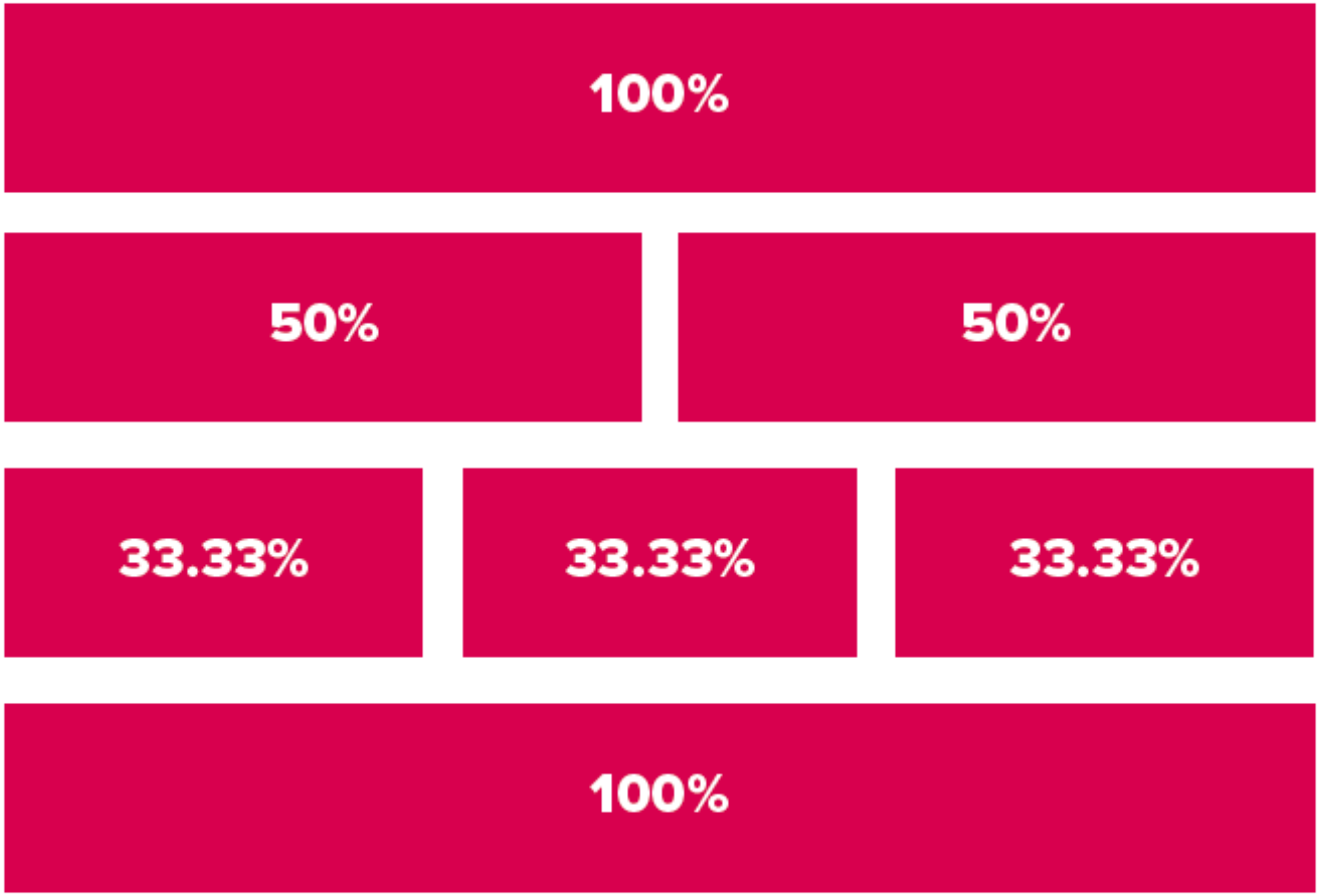
Texts such as images when inserted in efficient grids will follow the course of the columns until the arrival of a point of no visibility.

Also here the media queries that define when to change the layout of the grid itself come into play.

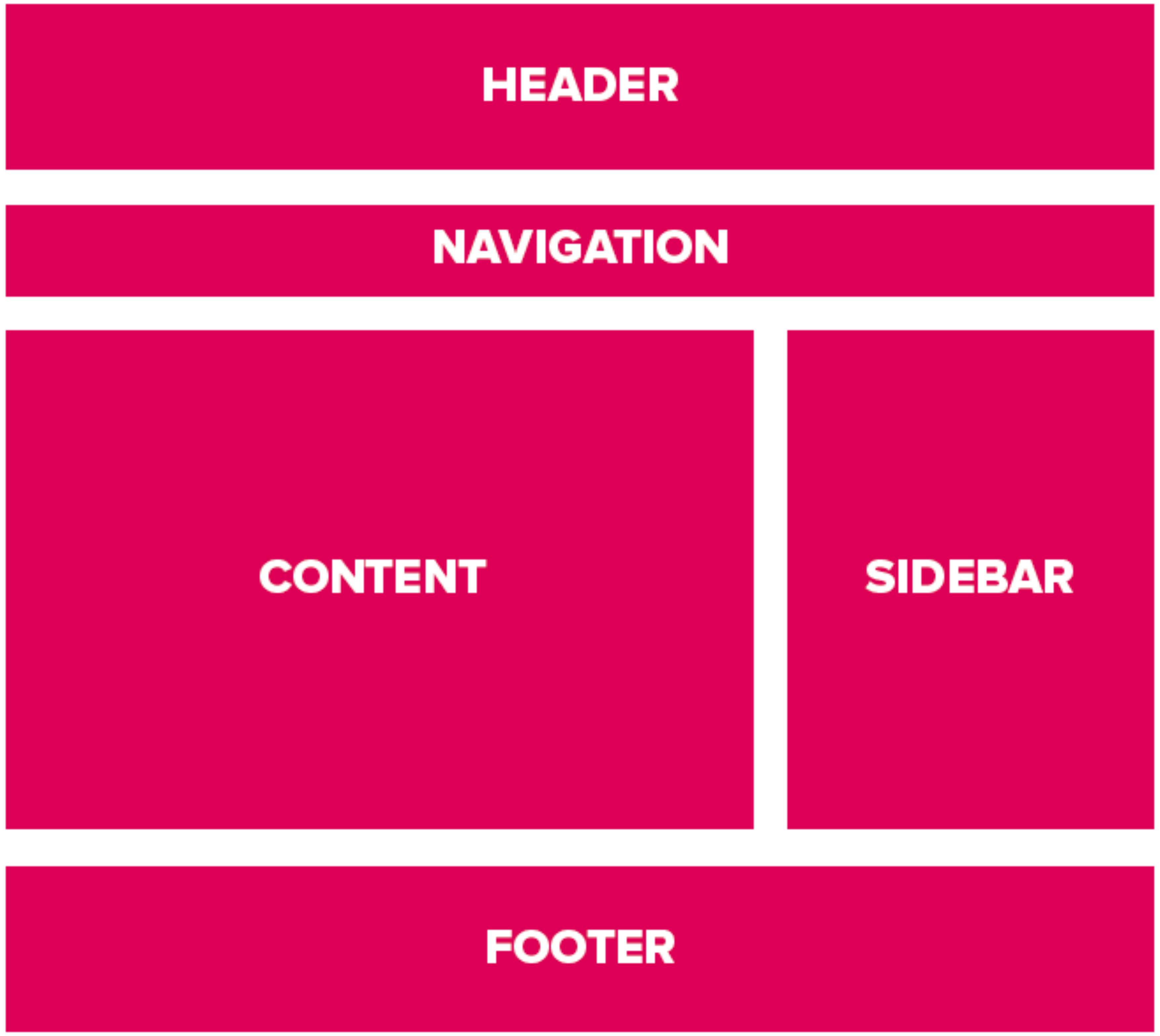
Testing

I heard you want to be a web developer
Here are a few devices to test your site

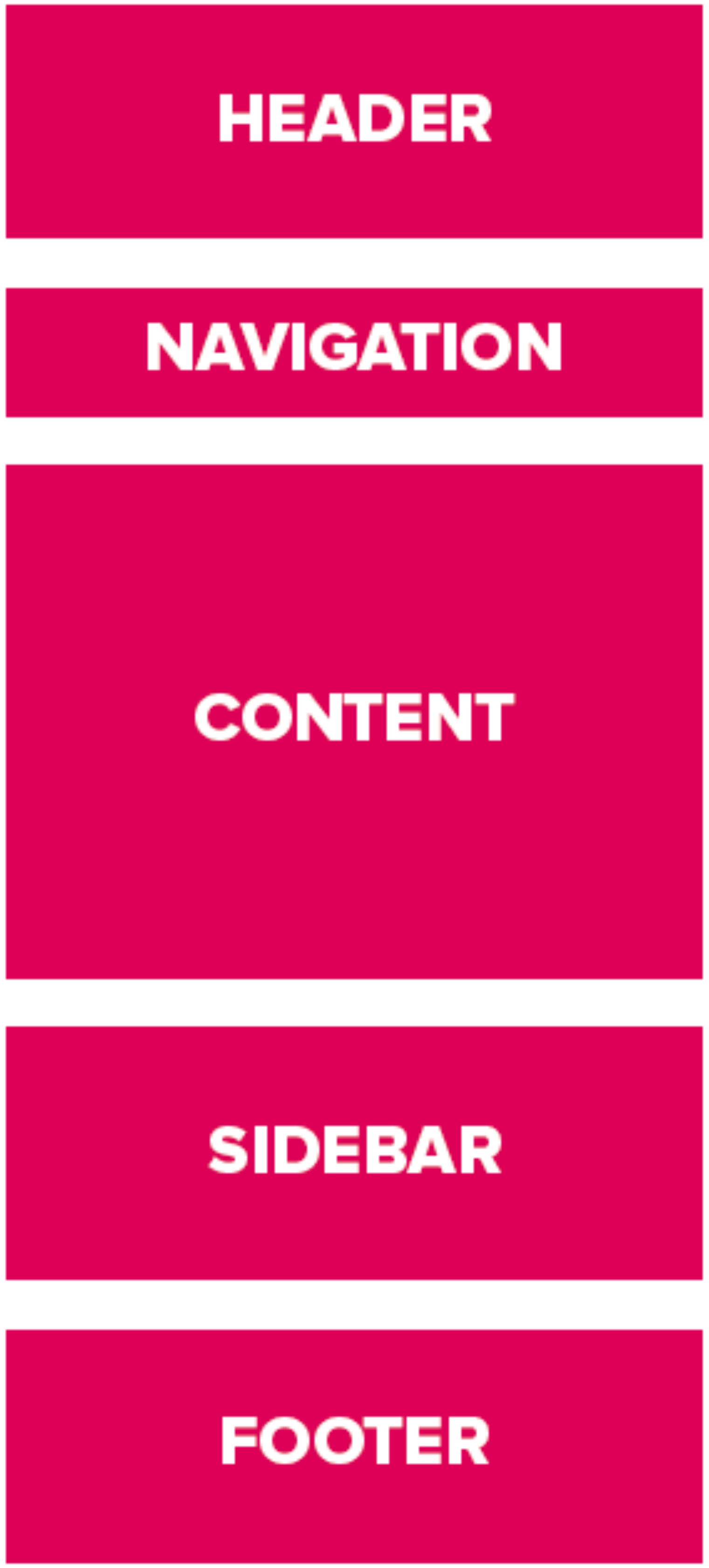
Settings



Settings

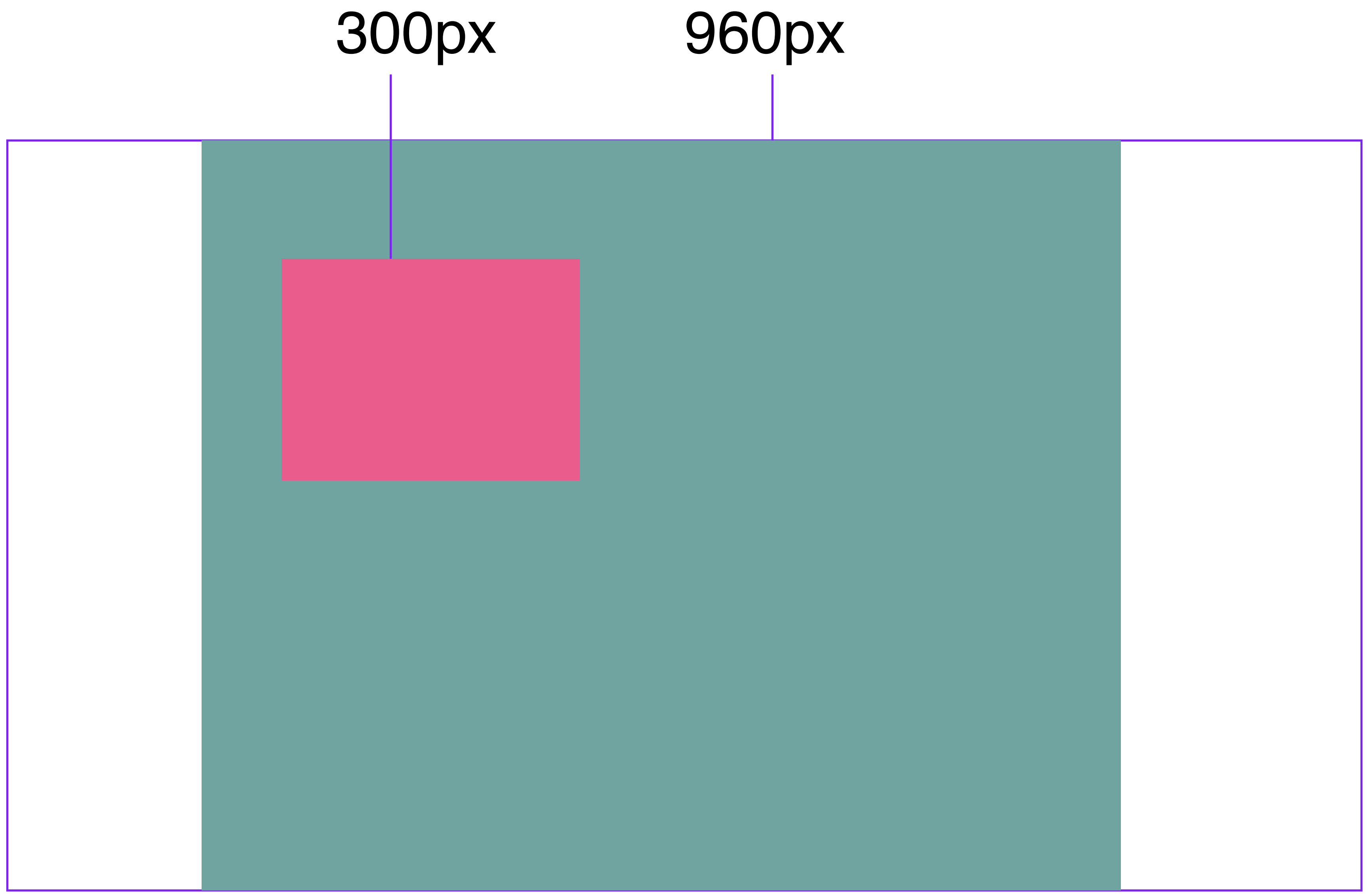


Settings



from px to %

The conversion from pixel to percentage is done thanks to the proportions, to do this you need to know 2 fundamental values:
The size of the graphic cage (eg. 960px/1024px)
The size of the single block (eg. 300px)



from px to %

To get the value in % we must refer to some mathematical rules:

(initial size / container size)* 100 = final value

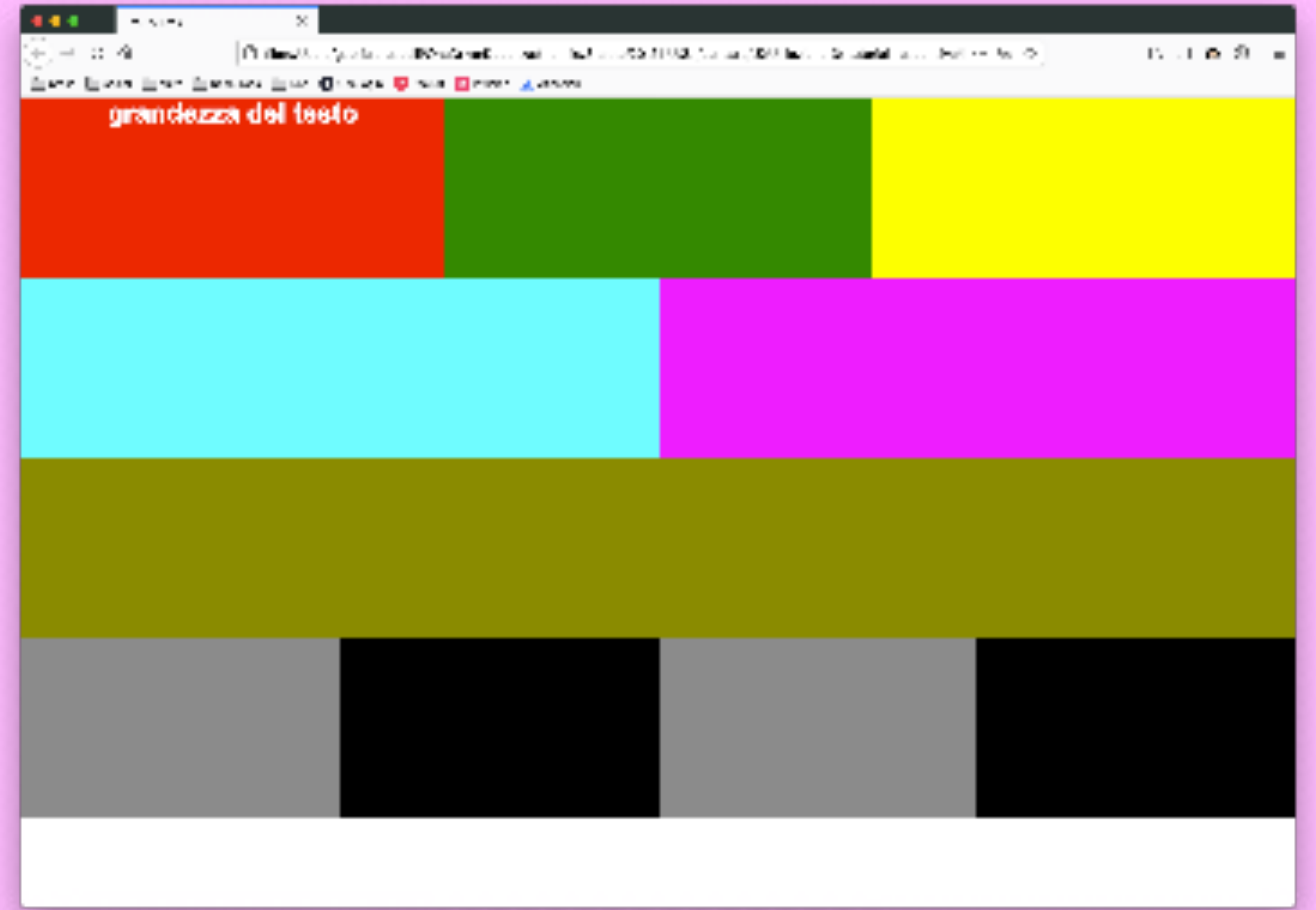
in our case:

(300/960)*100 = 31.25%

Now this new value will be the actual width of our box:

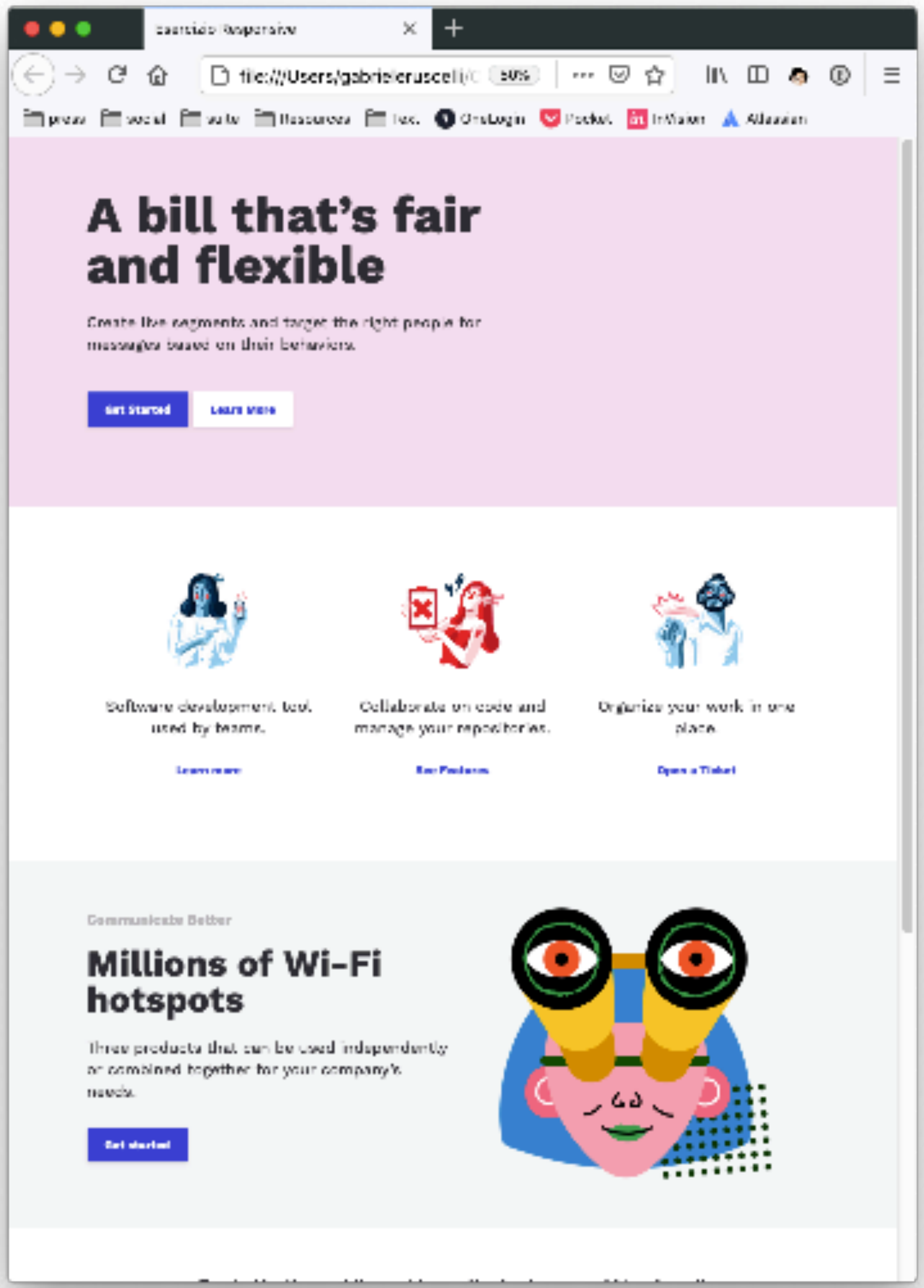
```
.box{
   width: 31.25%;
}
```

Tutorial



Exercise

Given the layout already created in HTML, you will choose the new layouts (tablets and smartphones).



A bill that's fair and flexible

Create live segments and target the right people for messages based on their behaviors.

Software development tool used by teams.

Collaborate on code and manage your repositories.

Organize your work in one place.

Communicate Better

Millions of Wi-Fi hotspots

Three products that can be used independently or combined together for your company's needs.

Viewport

Screen-related measures

units of measurement

vh	vw	vmin	vmax
Viewport Height: height of your screen or browser window	Viewport Width: width of your screen or browser window	Viewport Minimum: shortest side of your screen or browser window	Viewport Massima: longest side of your screen or browser window

units of measurement

1vh 1vw 1vmin 1vmax

=

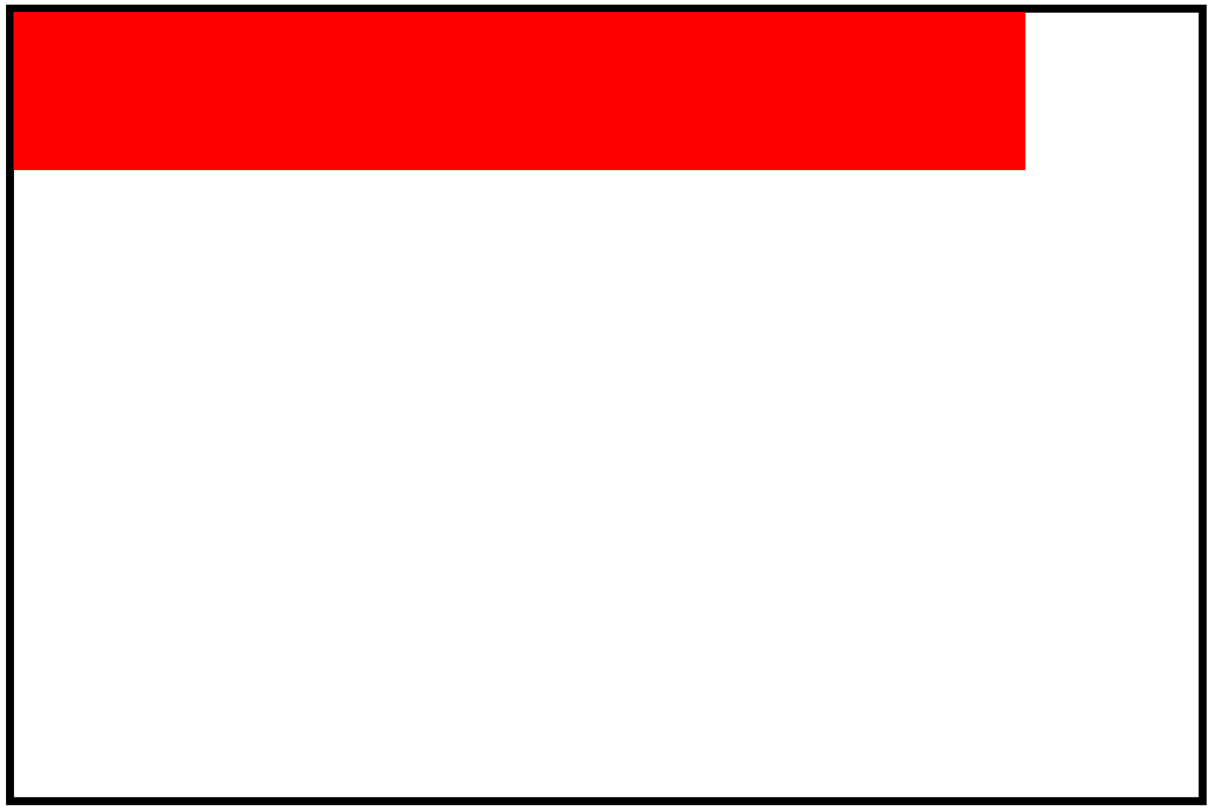
1%

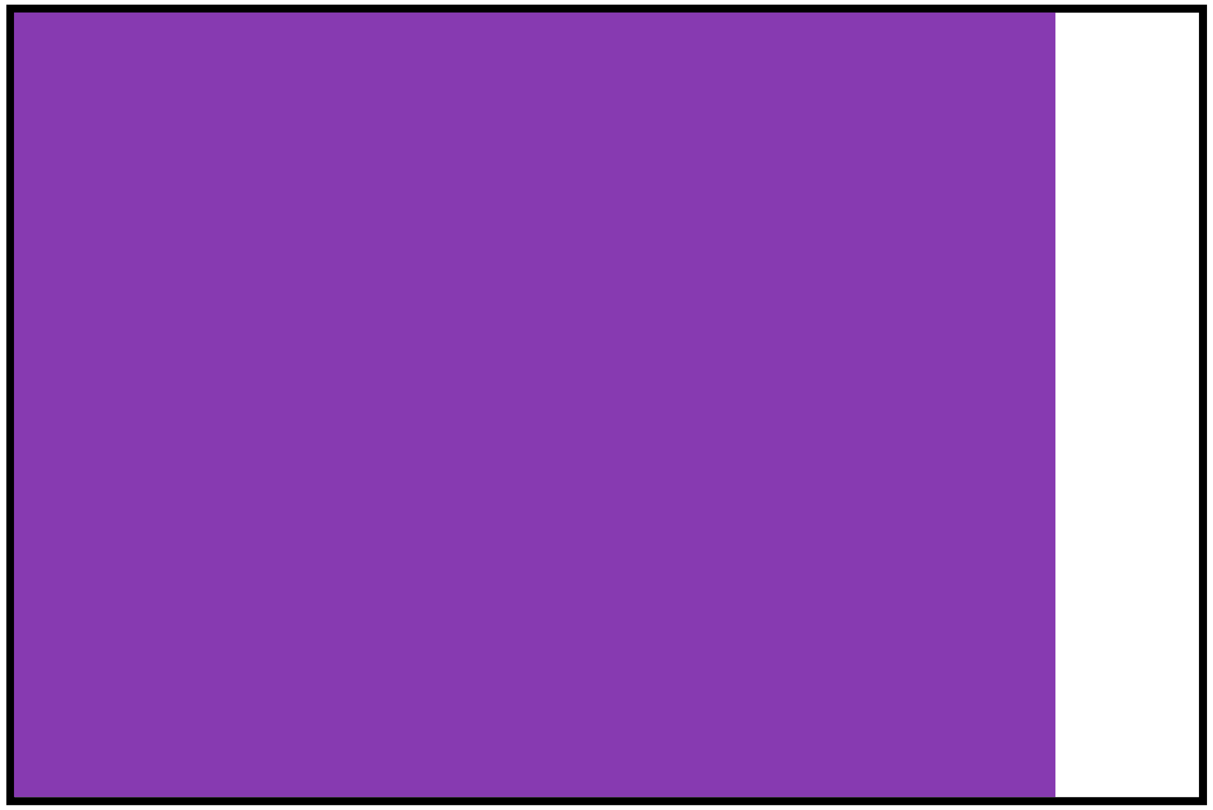
of the screen or browser window

vh

```
.red_box{
   width: 80%;
   height: 10vh;
}
```

```
.purple_box{
   width: 80%;
   height: 100vh;
}
```

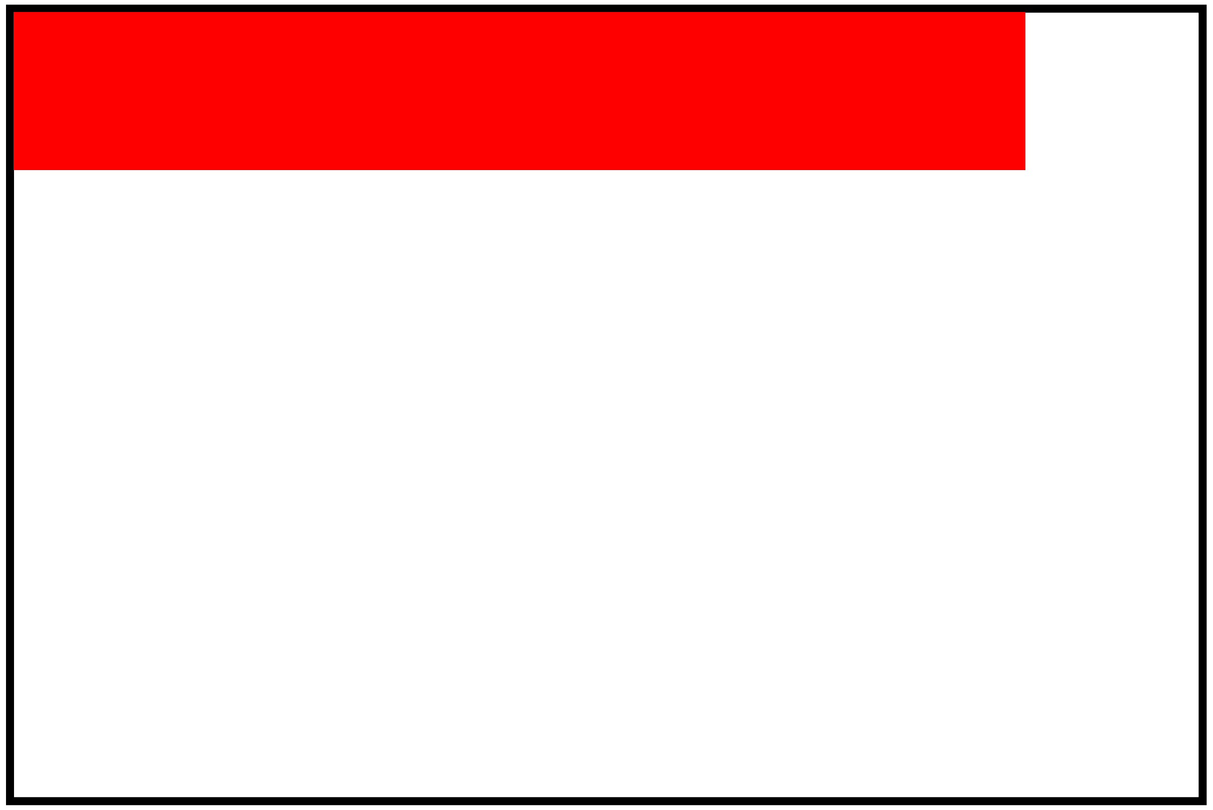


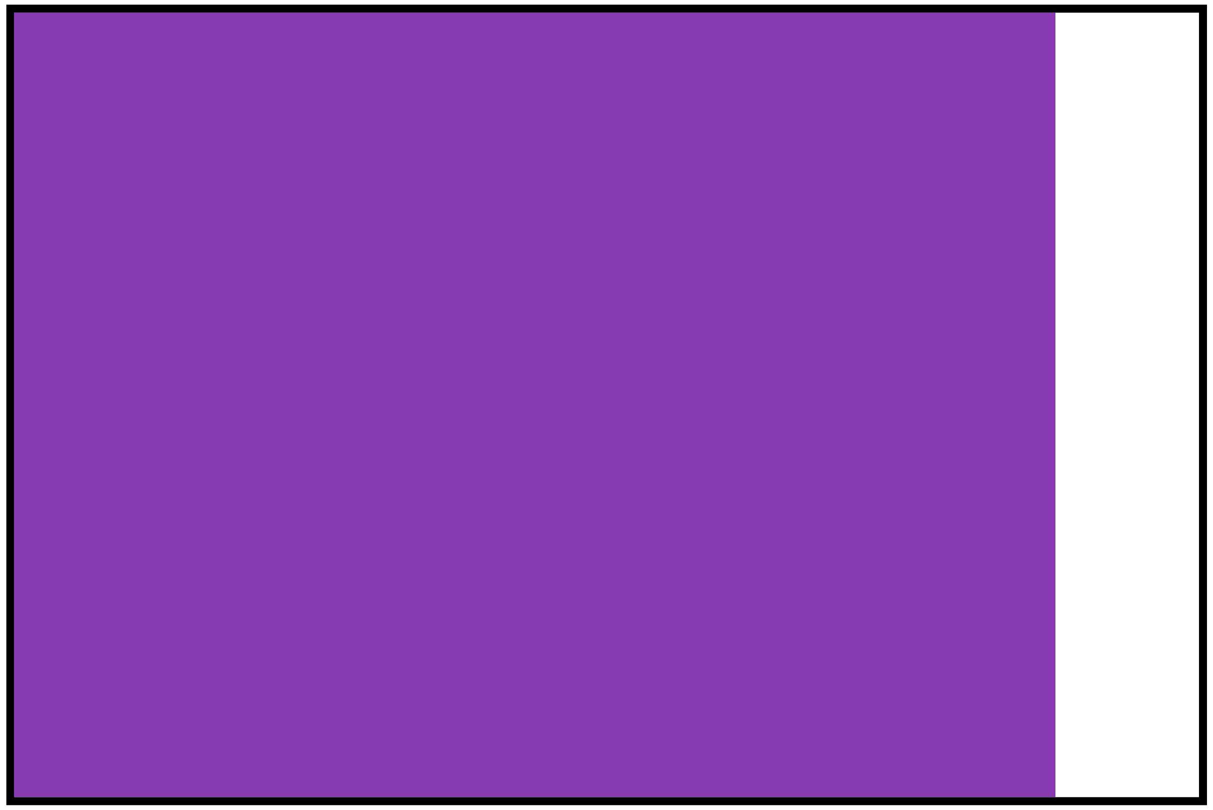


vw

```
.red_box{
   width: 80vw;
   height: 10vh;
}
```

```
.purple_box{
   width: 80vw;
   height: 100vh;
}
```

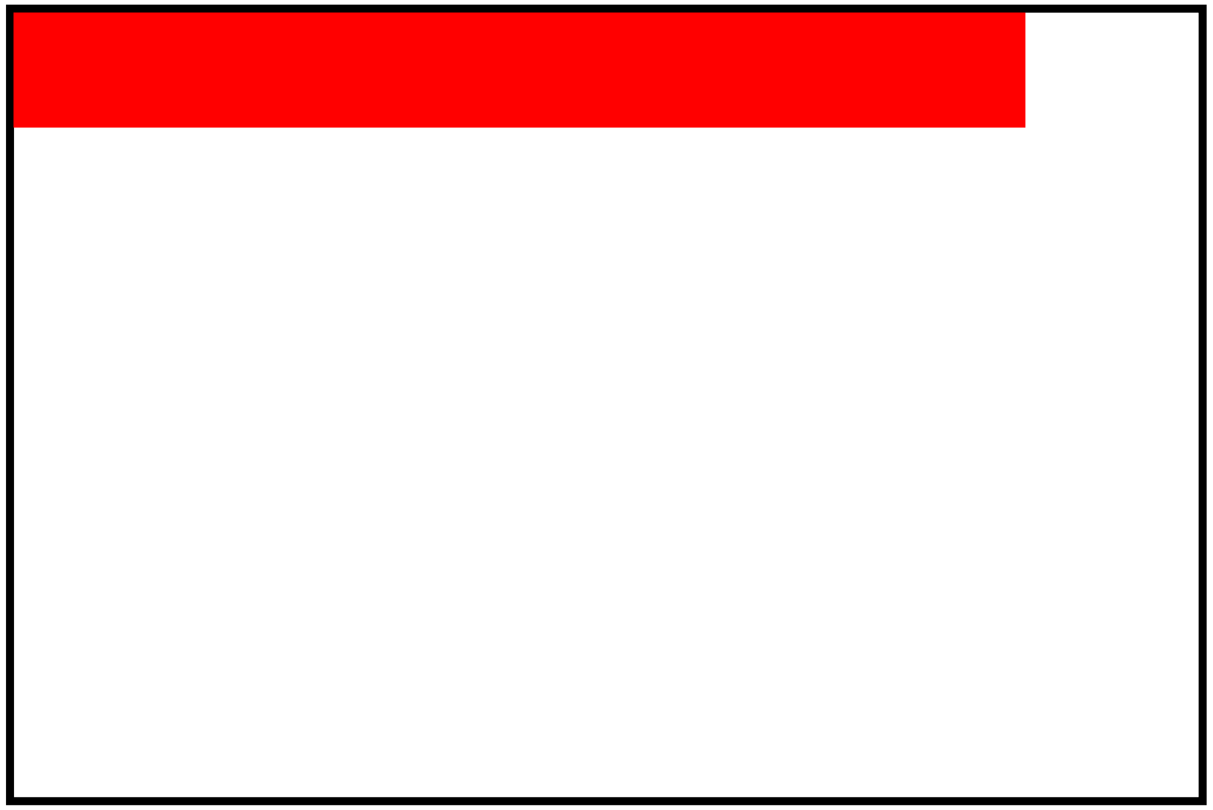
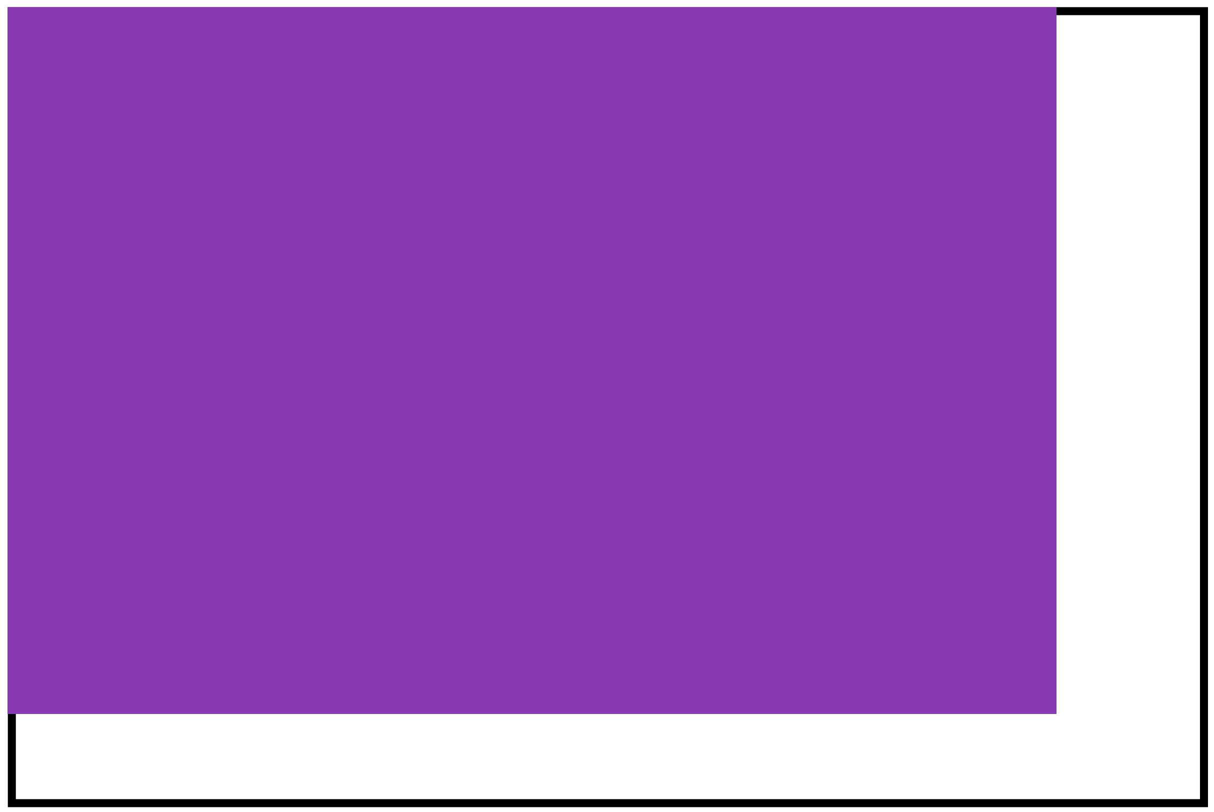




vw (2)

```
.red_box{
  width: 80vw;
  height: 10vw;
}
```

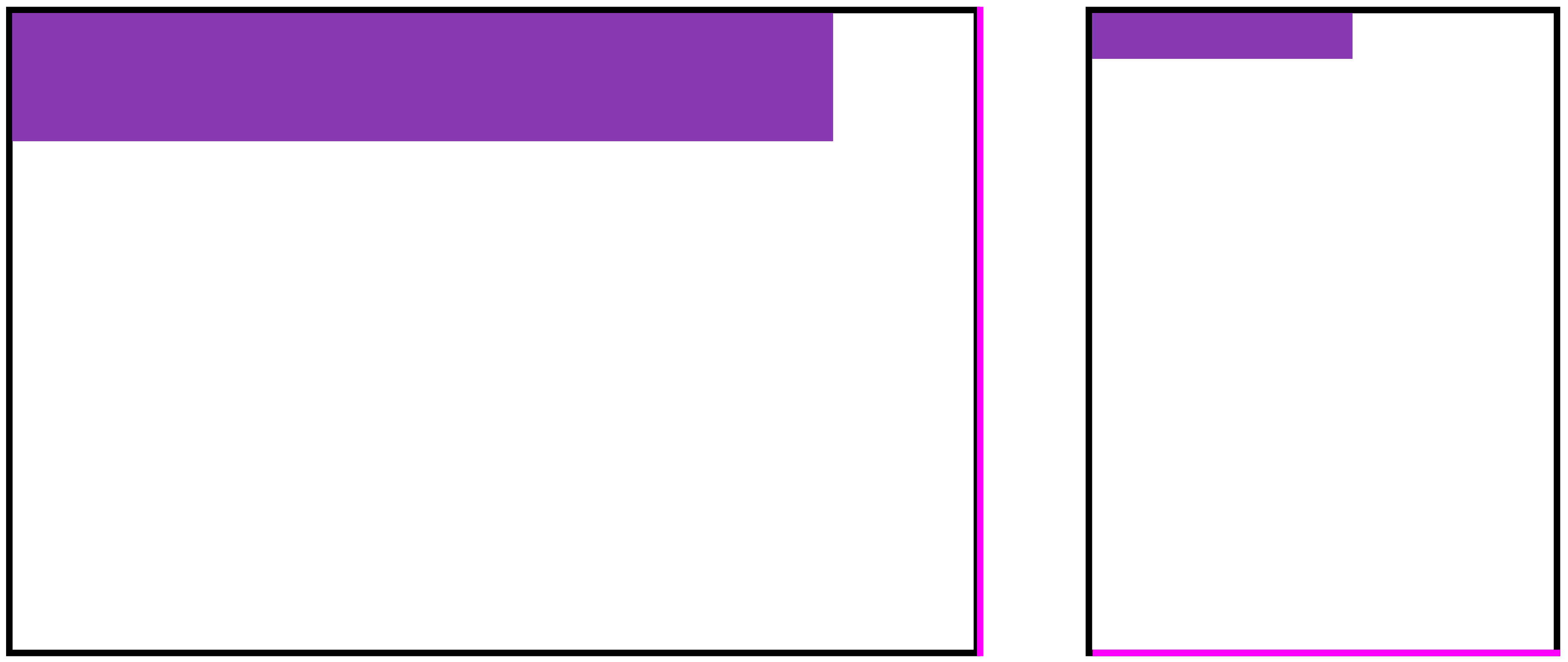
```
.purple_box{
  width: 80vw;
  height: 50vw;
}
```

vmin

```
.purple_box{
   width: 80%;
   height: 10vmin;
}
```

10% of the shortest side of your screen or browser window

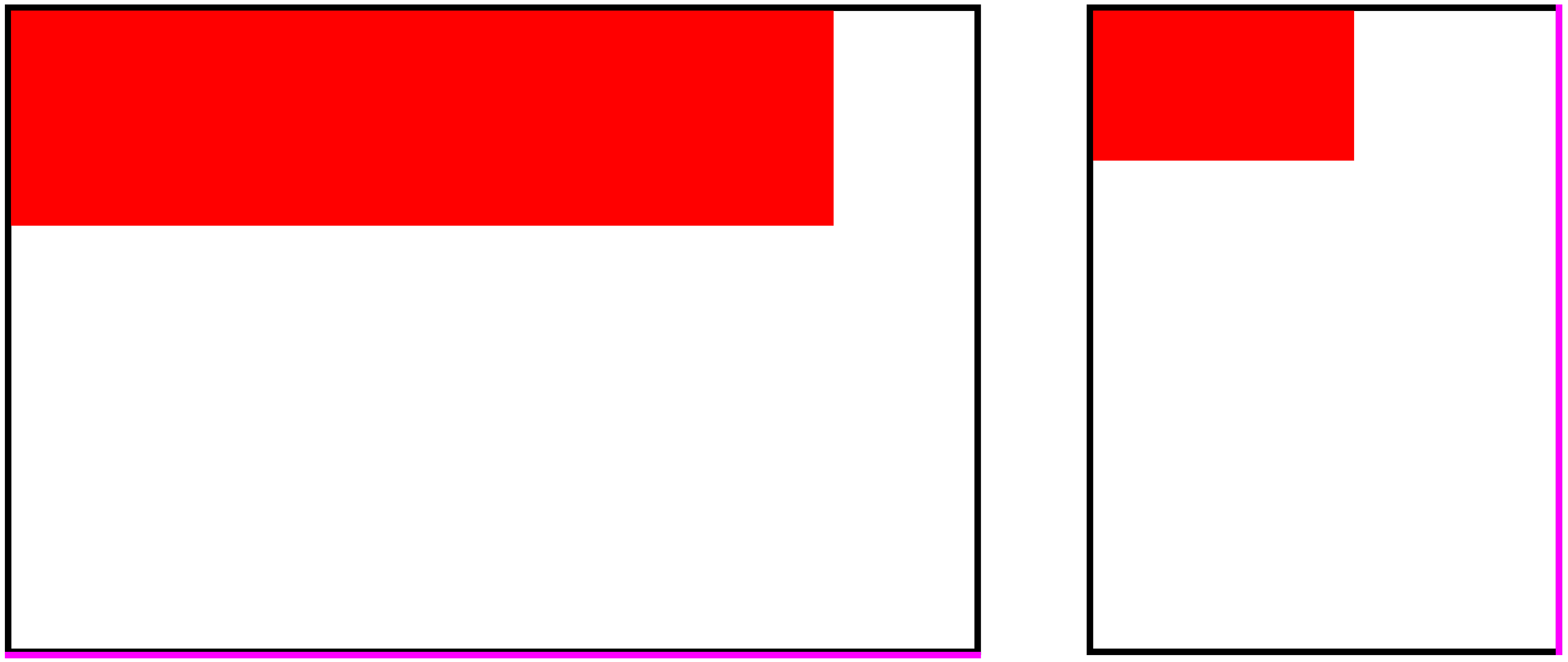


In size management you can block resizing when the screen defines the small side.

vmax

```
.red_box{
   width: 80%;
   height: 10vmax;
}
```

10% of the longest side of your screen or browser window



Nella gestione di grandezze è possibile bloccare il ridimensionamento quando lo schermo definisce il lato grosso.
In questo caso stringo la pagina lateralmente è il box sarà invariato

viewport measures

Useful for:

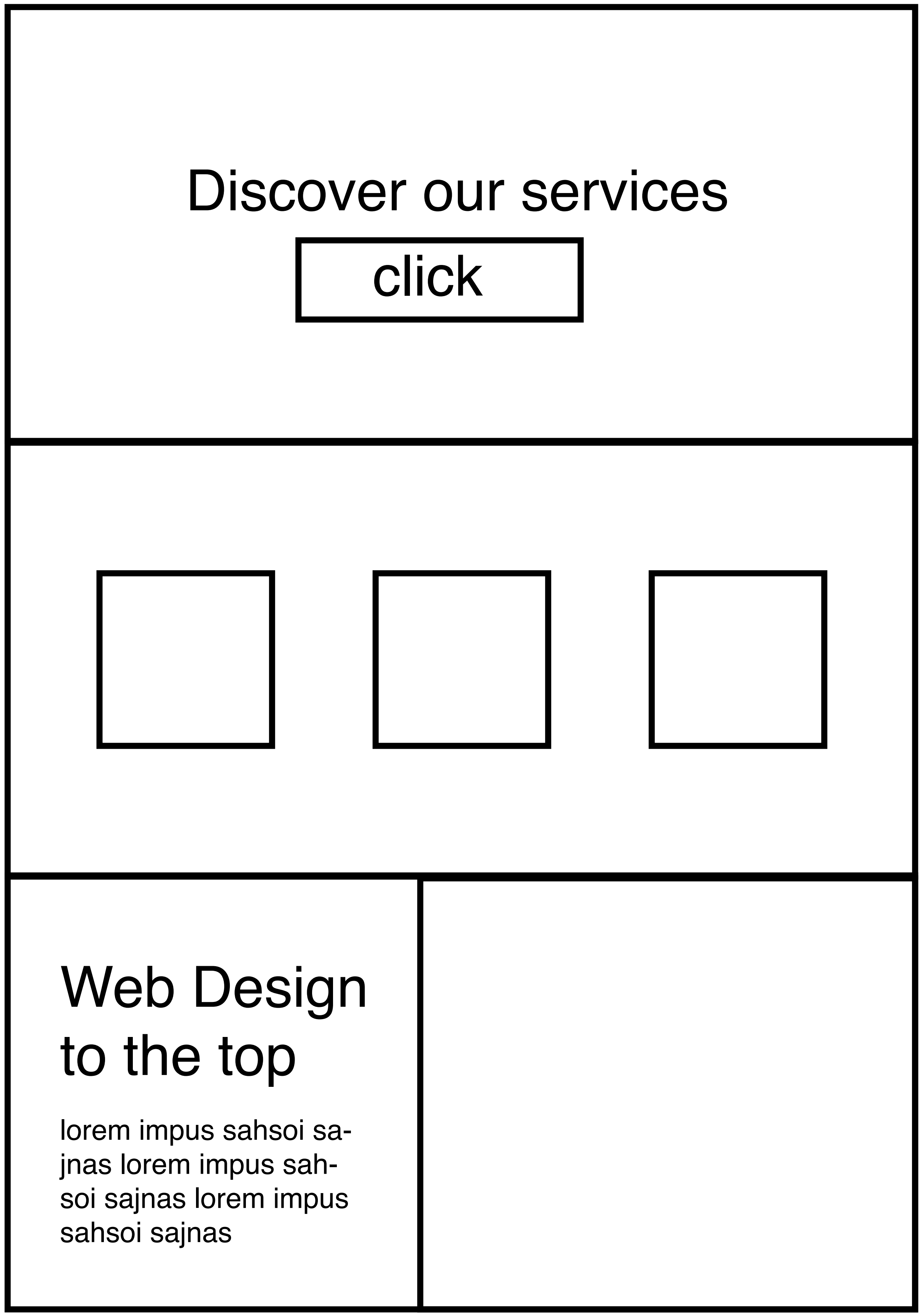
occupy the whole vertical space of the page

make text responsive

make square boxes responsive (before you had to create them only in pixels)

design complex, precise and fluid layouts

Exercise


Discover our services
click
Web Design to the top
lorem impus sahsoi sajnas lorem impus sahsoi sajnas lorem impus sahsoi sajnas

with teacher