

LEARN WEB

ale66

THE LIMITS OF HTML

THE PAGE ORIENTATION

So far, each HTML page is a stand-alone, complete page

Even recurrent elements, e.g., links to reference pages, are repeated

pages might share a common CSS file, but not the HTML structure itself

CASE STUDY

- [unimi.it](#) offers more than 150 study programs
- each programme page has a similar structure and most information is actually replicated
- yet, in principle we should come up with 150+ HTML pages

IDEA: SEMANTIC TAGGING

recurrent parts of the page are annotated by *invisible*, semantic tags

although invisible, the tags guide presentation, and can be replicated across the whole platform, e.g., **unimi.it**

STATIC LAYOUT

SO FAR

pages are intended as rectangular: relatively narrow and tall

- like the book page
- like the computer screen

SOLUTIONS, A

The CSS may define *boxes* with padding around them

The boxes may *float* so the browser will assign it dynamically to the available *viewport*

However, browsers present elements one after the other, until space on the right ends and they continue on a new line below: **undesirable effects and even collapse**

EXPERIENCE: FLOATING ELEMENTS

- Ex. 8.1: Floating elements

Elements start **overflowing** to the *line* below

Explicit placing: not recommended

- Ex. 8.2: Positioning elements

EXPERIENCE: BOXES WITH PADDING

Creating boxes of text: Ex. 9.1

Padding around them: Ex. 9.2

For reference: a better finish

- Ex. 9.3: Applying a Simple Border
- Ex. 9.4: Controlling the Margins

EXERCISE

Try *pair coding* with the Live Share extension of VS Code

- in your project, Box up the official text from [unimi.it](#) and make it float
- make a menu of navigation links and float it (details in past presentations)

LANDING PAGES

Hero page:

- seeks an emotional response
- graphics takes over from text

Landing page

- extends hero
- the page that search engines point
- seeks to entice and guide a deeper visit

EXAMPLE LANDING PAGE

The screenshot shows the homepage of the University of Milan's website. At the top, there is a dark blue header bar with the URL <https://www.unimi.it/en> in the address bar. To the right of the address bar are browser controls for zoom (80%), refresh, and a star icon. On the far right are icons for a user profile and a search function.

The main navigation menu is located below the header, featuring links to Unimia, Registrar, Ariel, Libraries, Who and Where, Webmail, laStatale@work, Info for (with a dropdown arrow), news, and language options (ITA | ENG). A large banner image in the background shows students in a classroom setting, viewed from behind, looking at laptops.

On the left side of the banner, there is a logo for the University's 100th anniversary, featuring the number '100' in white and blue, with the text 'UNIVERSITÀ DEGLI STUDI DI MILANO' underneath. In the center of the banner, there are six categories: Education, Study, Research, Third mission, International, and The University, each with a small icon above it. To the right of these categories are a magnifying glass icon for search and a three-line menu icon.

The main title 'Studying at the University of Milan' is displayed prominently in large white text across the center of the banner. Below this title, a subtitle 'Courses, class timetables and other learning activities' is shown in a smaller white font. In the bottom left corner of the banner, there is a 'LA STATALE news' logo. In the bottom right corner, there is a 'LEARN MORE' button enclosed in a white rectangular box.

MAKE A LANDING PAGE

- Create a *hero image* that is fixed wrt. scrolling but also contains the claim
- go picture-in-picture, e.g.



Fixed landing:

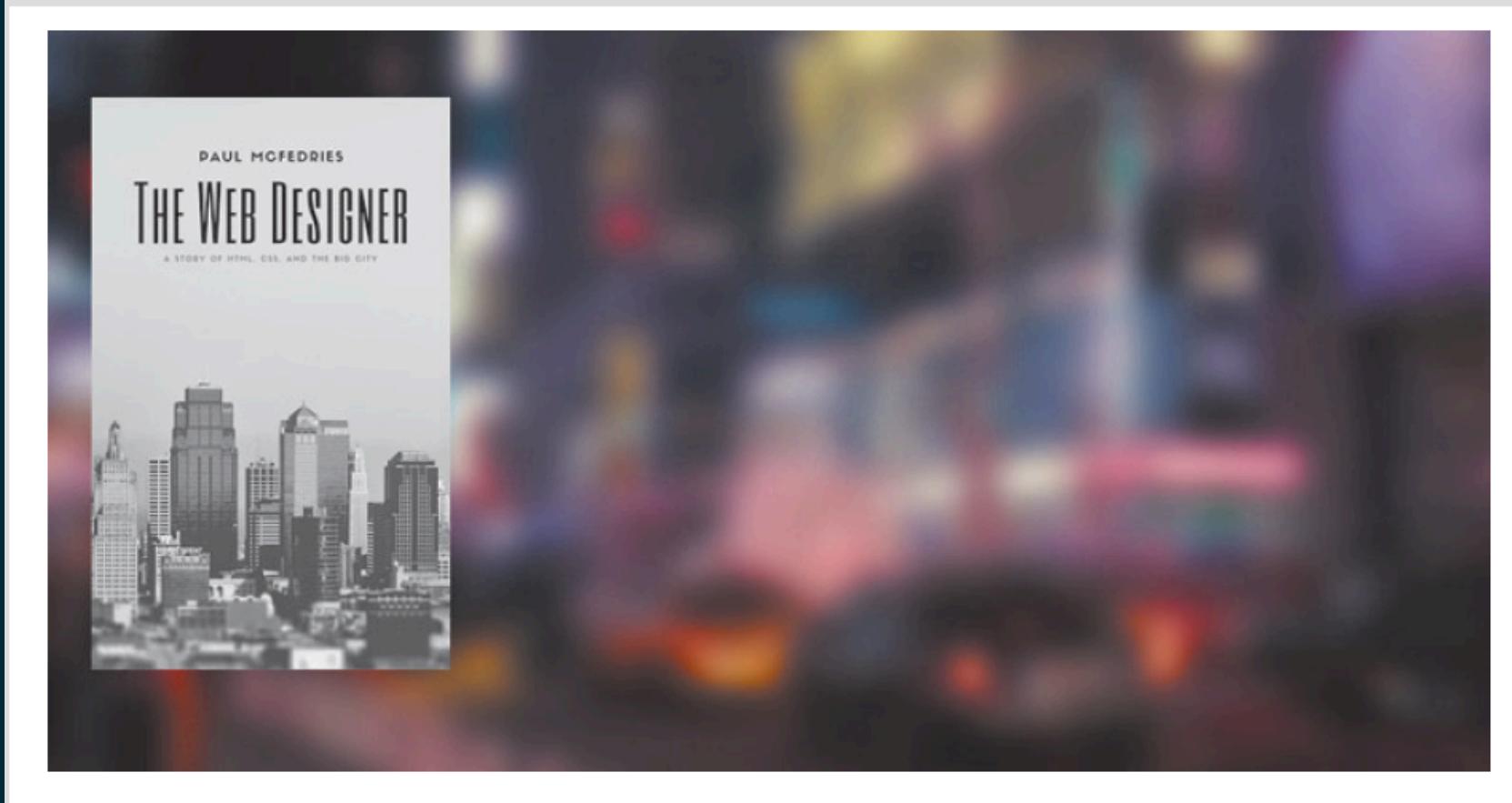
CSS

```
.header {  
    background: url(/images/landing-page-header-bg.jpg);  
    background-attachment: fixed; ← Prevent the hero  
    background-position: right center; image from scrolling  
    background-size: cover; with the content.  
    padding-bottom: 1em;  
    width: 100vw;  
    height: 100vh;  
}
```

HTML

```
<div class="header">  
</div>
```

Left picture is the *what*, background picture is the *where* or *how*



lab on Velasca? the other way around? Keep *lab* and choose an inspirational picture that works for you and your friends

THE IN-PICTURE

CSS

```
.header-image {  
    float: left;  
    width: 33%;  
    margin-top: 3em;  
    padding-right: 3em;  
    text-align: right;  
}
```

The CSS code
for the image

HTML

```
<div class="header">  
    <div class="header-image">  
          
    </div>  
</div>
```

The image
is added as
a standard
HTML img
element.

THE IN-TEXT

CSS	<pre>.header-info { float: right; width: 67%; margin-top: 4em; color: white; }</pre> <p>The CSS code for the product info div</p>
HTML	<pre><div class="header"> <div class="header-image"> </div> <div class="header-info"> <div class="surtitle">Coming Soonish!</div> <h1 class="title">The Web Designer</h1> <h3 class="subtitle"> A story of HTML, CSS, and the big city</h3> <p class="intro"> She knew HTML. She knew CSS. But did she know love? Read this destined-to-be-remaindered novel that The New York Times Book Review described as "reasonably grammatical"; and the Times Literary Supplement called "bathroom-worthy." Pre-order your paperback copy now for just \$14.99. </p> </div> </div></pre> <p>The info is added within a div element.</p>

Check the final effect with the [Live server VS Code extension](#)

SOLUTIONS, B

Complex CSS files can define three (computer, tablet and smartphone) or more *locales*

sense the device and apply the appropriate styles

Example: section with links collapses into the *hamburger* menu

FOR REFRENCE: before :: AND
after ::

::AFTER

We can avoid mis-rendering by forcing browsers to create the visual effects in a certain order

The :: notation is for *pseudo-elements*: we don't see them in the HTML but they go into the *final* page.

The CSS is now changing the HTML!

::BEFORE

apply a certain style before the browser starts rendering the element

See the [example from w3schools](#)

```
1 p::before {  
2   content: "Read this -";  
3   background-color: yellow;  
4   color: red;  
5   font-weight: bold;  
6 }
```

`::before`, `::after` and `::first-letter` are advanced CSS
language

we only cover them by example

See [reference](#)

STYLING TABLES AND OTHER STRUCTURED TEXTS

See the relevant self-learning material at

w3schools and

bootstrap