Build your landing page in 2 hours.

25th July, Le Wagon at Shoreditch House

arthur@lewagon.com

Today's objective

Code and design this page

Getting started

Download the sprint material on Github

Getting started #2

Download Sublime Text

Outline

1. Front-end 101
2. HTML basics (page structure, syntax & common tags)
3. CSS basics (fonts & colors)
4. Structuring a page with `div`
5. Custom design with `id` and `class`

1. Front-end 101

General knowledge about front-end languages

Frontend Languages

The languages your browser speaks

content
+ structure

style

dynamism
+ usability

Content & Structure

HTML

This is what Google cares about. Think of SEO.

Design

How it looks like.

Dynamic behavior

Animate stuff on screen (e.g. client-side form validation)

2. HTML Basics

All you need to know about HTML

HTML tags help you identify content

Hence, browser default styles apply

You can also apply your own style rules

HTML Skeleton

```
<!DOCTYPE html>









<!-- end of file -->
```

HTML Skeleton

```html
<!DOCTYPE html>
<html>
</html>
<!-- end of file -->
```

HTML Skeleton

```html
<!DOCTYPE html>
<html>
  <head>



  </head>
  <body>

  </body>
</html>
<!-- end of file -->
```

HTML Skeleton

```html
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Title of your page</title>
  </head>
  <body>

  </body>
</html>
<!-- end of file -->
```

HTML Skeleton - head & Google

```
<head>
  <!-- Google result's text-->
  <title>Coding Bootcamp Le Wagon | Europe's Best Coding B
  <!-- Google result's description-->
  <meta name="description" content="Le Wagon is Europe's b
</head>
```

Coding Bootcamp Le Wagon | Europe's Best Coding Bootcamp

https://www.lewagon.com/ ▾

Le Wagon is Europe's best coding bootcamp for creative people & entrepreneurs. Learn to code in 9 weeks with our fullstack coding bootcamp.

Paris

My name is Anne-Sophie, and I was part of Le Wagon Batch ...

FAQ

We know you have questions about our program. That's why ...

Program

Apply Now - FAQ - Demoday - Alumni - Le Wagon Berlin - Bali

Apply Now

9 weeks to learn ruby on Rails, HTML/CSS, Javascript, and ...

HTML Skeleton - head & Facebook

```
<head>
  <meta property="og:title" content="Coding Bootcamp Le Wa
  <meta property="og:image" content="facebook-card.jpg">
  <meta property="og:description" content="Le Wagon is the
  <meta property="og:site_name" content="Le Wagon"/>
</head>
```

HTML Skeleton - body

`body` is where you write content displayed on the page.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Hello world</title>
  </head>
  <body>
    <h1>Hello buddies!</h1>
  </body>
</html>
<!-- end of file -->
```

HTML Syntax

name
value
<element attr="sthg"> [...] </element>
Start tag
Attribute
Content
End tag

Example

```html
<a href="https://www.lewagon.com" target="_blank">
  Le Wagon
</a>
```

Result: Le Wagon

Questions:

- What is the element's name?
- What is the element's content?
- What are the 2 attributes (name and value)?

Titles

```
<h1>[...]</h1>  <!-- Only one per page! SEO important -->
<h2>[...]</h2>
<h3>[...]</h3>
<h4>[...]</h4>
<h5>[...]</h5>
<h6>[...]</h6>
```

Paragraphs

```
<p>
  Lorem ipsum dolor sit amet, consectetur adipisicing eli
  Veritatis laboriosam mollitia autem at ab omnis iure qu
  asperiores inventore eos nam aut iusto officiis deserun
  nihil, sequi tempore impedit quae?
</p>
```

Emphasize

```
<p>
  You can emphasize <em>some words</em>,
  and even <strong>more if needed</strong>
</p>
```

Lists

```html
<!-- Bullet points -->
<h2>Shopping List</h2>
<ul>
  <li>Milk</li>
  <li>Butter</li>
</ul>

<!-- Numbers -->
<h2>World Cup 2018</h2>
<ol>
  <li>France</li>
  <li>Uruguay</li>
  <li>Belgium</li>
  <li>England</li>
</ol>
```

Images

```
<img src="logo.png" alt="Le Wagon logo">
```

Forms

```
<form>
  <input type="email">
  <input type="password">
  <input type="submit" value="Sign up">
</form>
```

Let's add some HTML to our page.

Sprint #1 (20 minutes)

- Start with the `boilerplate`
- Take your time for the setup (Sublime Text / Google Chrome)
- Replace current `images` with personal ones
- **Focus on the content (texts and images) of the result page we want**. Don't worry about the design righ now.
- Your time to code the HTML content of your page.

3. CSS Basics

Style rules for fonts, colors, images & links.

Web without CSS?

Let's cut the `<head>` on facebook.com with Chrome dev tool.

This is how a website looks like without CSS .

Linking a stylesheet

index.html

```
1 <head>
2   <link rel="stylesheet" href="style.css">
3 </head>
4 <body>
5   <h1>Hello</h1>
6 </body>
```

style.css

```
1 h1 {
2   color: red;
3 }
```

CSS Syntax

selection of one or more elements of the page..

```
1 selector {
2   property: value;
3   property: value;
4   property: value;
5 }
```

..on which we define style rules

CSS Vocabulary

CSS selector

```
1 selector {
2   property: value;
3   property: value;
4   property: value;
5 }
```

style rules

style properties

CSS Example

we select
all <h2> of the page

```
1 h2 {
2   color: red;
3   font-size: 20px;
4   font-family: arial;
5 }
```

CSS Colors

```
color: #FF530D;
color: rgb(255, 83, 13);
color: rgba(255, 83, 13, 1.0);
```

Tips for grey scale

```
body {
  color: rgb(10, 10, 10);
}
```

Text vs. Background

```
body {
  color: rgb(10, 10, 10);
}
```

text color

```
1 body {
2  color: orange;
3  background-color: yellow;
4 }
```

background color

Fonts - family

```
1 body {
2  font-family: Times, serif;
3 }
```

Abc

```
1 body {
2  font-family: Arial, sans-serif;
3 }
```

Abc

```
1 body {
2  font-family: Courier, monospace;
3 }
```

Abc

Fonts - size & spacing

```
1 body {
2  letter-spacing: 10px;
3 }
```

```
1 body {
2  font-size: 30px;
3 }
```

```
1 body {
2  line-height: 1.6;
3 }
```

Fonts - decoration

```
1 p {
2  text-decoration: underline;
3 }
```

Abc

```
1 a {
2  text-decoration: none;
3 }
```

sign up

↓

sign up

Fonts - alignment

```
1 p {
2  text-align: center;
3 }
```

This is a centered paragraph aligned with text-align

```
1 p {
2  text-align: justify;
3 }
```

This is a justified paragraph aligned with text-align

Fonts - weight

```
1 body {
2  font-weight: 100;
3  // from 100 to 900.
4 }
```

Light text

```
1 body {
2  font-weight: bold;
3 }
```

Bold text

Let's add some simple CSS (fonts, colors).

Sprint #2 (10 minutes)

- Pick two fonts on Google fonts (1 for `body` + 1 for `h1, h2, h3`) and embed them into the `head` section.
- Don't forget to `link` your `style.css` in the `head` as well.
- Pick a color scheme on Color Hunt or Coolors.
- Define CSS rules for colors & fonts: `body`, `h1`, `h2`, `h3`, `a`.
- Also design the `a:hover` state.

4. Structuring a page with `div`

You need to split you page into sections.

Real life...

Paris, France – Airbnb
Guest
https://www.airbnb.co.uk/s/Paris--France/
Paris
Anytime
2 guests
Become a Host
Help
Sign Up
Log In
FOR YOU
HOMES
EXPERIENCES
PLACES
Room type
Price range
Instant Book
More filters
Enter dates to see full pricing. Additional fees apply. Taxes may be added after login.
NEW
From £28 · Studio sur cour, calme, proche T...
Entire condominium · 1 bed
From £28 · Studio rdc à Paris centre Proche...
Entire flat · 1 bed
1 review
From £37 · ❤ Cosy studio for couples ❤ Vie...
Entire flat · 1 bed
From £50 · Eiffel Tower-tiny studio 14m2
Entire flat · 1 bed
1 review
From £28 · Studio 21m2 calme entre la ville ...
Entire flat · 1 bed
From £46 · Cosy 25m² in Paris 11th
Entire flat · 2 beds
Search as I move the map
Language and currency

...is made of `<div>`

Box model - content

Box model - `padding` & `margin`

Box model - border

Border's syntax

```
div {
  border-top: 1px solid red;
  border-right: 2px dotted black;
  border-bottom: 1px dashed green;
  border-left: 2px dotted black;
}
```

Border radius

```
border-radius: 5px;
```

Box shadow

box-shadow: 2px 2px 3px blue;

Units

```
/* Absolute */
div {
  width: 50px;
}

/* Relative to parent */
div {
  width: 50%;
}

/* Relative to font size */
div {
  width: 2em;
}
```

Classic `<div>` design

```
background: white;                          /* White backg
padding: 30px;                              /* Internal sp
border-radius: 4px;                         /* Small radiu
box-shadow: 0 15px 35px rgba(50,50,93,.1); /* Stripe shad
```

An example of this design here.

`<div>` centering technique

```
width: 300px;   /* Fix the width */
margin: 0 auto; /* Set automatic margins for left & right
```

Classic centering technique.

Let's add some structure to our HTML with `div` tags.

Sprint #3 (10 minutes)

- Look carefully at our result to understand the `<div>` structure of the page.
- Once you have it, quickly add the `<div>` in your HTML to wrap the relevant content.

Correction for `<div>` structure

```
<body>
  <div> <!-- Container (to center content) -->
    <div> <!-- Header -->
      ...
    </div>
    <div> <!-- Section -->
      ...
    </div>
    <div> <!-- Section -->
      ...
    </div>
    <div> <!-- Section -->
      ...
    </div>
    <div> <!-- Footer -->
      ...
    </div>
  </div>
</body>
```

5. Custom design with `id` and `class`

Let's do some real design.

What's the issue?

How do you style only the logo?

HTML

```html
<img src="logo.png">

<ul>
  <li><img src="paul.png"></li>
  <li><img src="ringo.png"></li>
  <li><img src="george.png"></li>
  <li><img src="john.png"></li>
</ul>
```

CSS

```css
img {
  width: 40px;
}
```

Name your element with an `id`

HTML

```html
<img src="logo.png" id="logo">

<ul>
  <li><img src="paul.png"></li>
  <li><img src="ringo.png"></li>
  <li><img src="george.png"></li>
  <li><img src="john.png"></li>
</ul>
```

CSS

```css
#logo {
  width: 40px;
}
```

How do you style only team pictures?

HTML

```html
<img src="logo.png" id="logo">

<ul>
   <li><img src="paul.png"></li>
   <li><img src="ringo.png"></li>
   <li><img src="john.png"></li>
   <li><img src="george.png"></li>
</ul>
```

Name your elements with a `class`

HTML

```html
<img src="logo.png" id="logo">

<ul>
   <li><img src="paul.png" class="img-circle"></li>
   <li><img src="ringo.png" class="img-circle"></li>
   <li><img src="john.png" class="img-circle"></li>
   <li><img src="george.png" class="img-circle"></li>
</ul>
```

CSS

```css
.img-circle {
  border-radius: 50%;
}
```

`id` or `class`?

unique

HTML

```html
<img src="logo.png" id="logo">

<ul>
    <li><img src="paul.png" class="img-circle"></li>
    <li><img src="ringo.png" class="img-circle"></li>
    <li><img src="john.png" class="img-circle"></li>
    <li><img src="george.png" class="img-circle"></li>
</ul>
```

re-usable

You can combine them

HTML
```html
<img src="paul.png" class="img-circle">
```

CSS
```css
.img-circle {
  border-radius: 50%;
}
```

You can combine them

HTML
```html
<img src="paul.png" class="img-circle shadowed">
```

CSS
```css
.img-circle {
  border-radius: 50%;
}

.shadowed {
  box-shadow: 5px 5px 3px black;
}
```

You can combine them

HTML

```html
<img src="paul.png" class="img-circle shadowed" id="leader">
```

CSS

```css
.img-circle {
  border-radius: 10px;
}

.shadowed {
  box-shadow: 5px 5px 3px black;
}

#leader {
  border: 5px solid red;
}
```

What's your intuition?

What are the best `class` names?

- `.btn-red` or `.btn-signup`?
- `.background-blue` or `.background-home`?
- `.img-user` or `.img-circle`?

Tip: think graphical

A classic naming convention

```
/* Convention */
.element-design

/* Examples*/
.text-center
.text-justify
.btn-red
.btn-green
.btn-big
.list-inline
.form-horizontal
.img-rounded
.img-circle
```

Let's rehearse our CSS selectors

Element selectors

```
<!-- index.html -->
[...]
<body>
  <h1>Hello World</h1>
</body>
```

combined with

```
/* style.css */
h1 {
  color: red;
  font-weight: bold;
}
```

makes the `h1` elements red and bold.

Class selectors

```
<!-- index.html -->
[...]
<body>
  <p>This paragraph is not justified</p>
  <p class="text-justify">This one is</p>
  <p class="text-justify">This one also</p>
</body>
```

combined with

```
/* style.css */
.text-justify {
  text-align: justify;
}
```

makes only the second and third paragraphs justified.

Id selectors

```html
<!-- index.html -->
<body>
  <div id="banner">
    <h1>Le Wagon</h1>
    <p>We bring tech skills to creative people</p>
  </div>
</body>
```

combined with

```css
/* style.css */
#banner {
  background-image: url("example.jpg");
  background-size: cover;
}
```

puts an image background on the unique div with `id="banner"`.

Descendant selectors

```html
<!-- index.html -->
<body>
  <div id="banner">
    <h1>Le Wagon</h1>
    <p>We bring tech skills to creative people</p>
  </div>
</body>
```

combined with

```css
/* style.css */
#banner h1 {
  color: white;
}
```

`h1` children of the element `id="banner"` will be white.

Grouping

```
/* style.css */
h1, h2, h3 {
  font-weight: bold;
}
```

is a shortcut syntax for

```
/* style.css */
h1 {
  font-weight: bold;
}
h2 {
  font-weight: bold;
}
[...]
```

Pseudo-class

```
/* style.css */
a {
  color: red;
  text-decoration: none;
}

a:hover {
  text-decoration: underline;
}
```

will make links underlined when the mouse hovers over them.

Quiz

Do you get it? Let's see!

Question #1

What's the text color?

```
<!-- index.html -->
<body>
  <p class="text-red">
    Lorem ipsum dolor sit amet, consectetur adipisicing e
  </p>
</body>
```

combined with

```
/* style.css */
p {
  color: black;
}
.text-red {
  color: red;
}
```

Question #2

What's the text color?

```
<!-- index.html -->
<body>
  <p id="bio" class="text-red">
    Lorem ipsum dolor sit amet, consectetur adipisicing e
  </p>
</body>
```

combined with

```
/* style.css */
.text-red {
  color: red;
}
#bio {
  color: green;
}
```

Specificity of selectors

```
p {                     /* least specific */
  color: black;
}
.text-red {             /*        Ã¢â€ â€œ       */
  color: red;
}
#bio {                  /* most specific  */
  color: green;
}
```

Check out the specificity calculator

Let's do some real design with `class` and `id`.

Sprint #4 (20 minutes)

- Name the elements you want to design with `class` and `id`
- Code like a pro and use the naming convention `.element-design` when choosing your `class` names.
- Once you've added all your `class` and `id` in the HTML, it's time to design the corresponding elements in the CSS to get this result (we'll see how to make the cards horizontal just after).

How can we make our cards horizontal?

We can use `flexbox`!

Flexbox - Basics

```
.flexbox {
  display: flex;
}
```

Flexbox - space around

```
.flexbox {
  display: flex;
  justify-content: space-around;
}
```

Flexbox - space between

```
.flexbox {
  display: flex;
  justify-content: space-between;
}
```

Flexbox - align items

```
.flexbox {
  display: flex;
  justify-content: space-between;
  align-items: center;
}
```

Solution

```html
<!-- index.html -->
<div class="cards-flex">
  <div class="card">
    <!-- content of card... -->
  </div>
  <div class="card">
    <!-- content of card... -->
  </div>
  <div class="card">
    <!-- content of card... -->
  </div>
</div>
```

combined with this CSS

```css
/* style.css */
.cards-flex {
  display: flex;
  justify-content: space-around;
}
```

Thank you!

- Want to go further? have a look at our fullstack program .
- Follow us on meetup to come to our other workshops like **Ruby for beginners**, **Javascript for beginners**, **API for beginners** and many more!
- *Code your MVP Landing Page* - full day of coding on the 1st of September, only 20 seats Register here.