5. Styling with CSS

Table of Content

Display Property

The `display` property specifies the display behavior (the type of rendering box) of an element.

The Display properties can contain the following values:

- inline
- block
- inline-block
- none
- flex
- grid

We will talk about the flex and grid values later in this course. Let’s talk about the first 4 values here.

▼ **inline**

The element generates one or more inline element boxes that do not generate line breaks before or after themselves. In normal flow, the next element will be on the same line if there is space.

Some examples of inline elements are `span, img, a, strong, b, i, etc.`

▼ **block**

The element generates a block element box, generating line breaks both before and after the element when in the normal flow.

Some examples of the block elements are `h1 to h6, p, div, form, header, footer, etc.`

▼ inline-block

- Compared to `display: inline`, the major difference is that `display: inline-block` allows to set a width and height on the element.
- Also, with `display: inline-block`, the top and bottom margins/paddings are respected, but with `display: inline` they are not.
- Compared to `display: block`, the major difference is that `display: inline-block` does not add a line-break after the element, so the element can sit next to other elements.

▼ none

Using a `display` value of `none` on an element will remove it from the accessibility tree

. This will cause the element and all its descendant elements to no longer be announced by screen reading technology.

CSS Float and Clear

▼ CSS Float

The CSS `float` property specifies how an element should float.

The `float` property is used for positioning and formatting content e.g. letting an image float left to the text in a container.

The `float` property can have one of the following values:

- `left` - The element floats to the left of its container
- `right` - The element floats to the right of its container
- `none` - The element does not float (will be displayed just where it occurs in the text). This is default
- `inherit` - The element inherits the float value of its parent

In its simplest use, the `float` property can be used to wrap text around images.

▼ CSS Clear

When we use the `float` property, and we want the next element below (not on right or left), we will have to use the `clear` property.

The `clear` property specifies what should happen with the element that is next to a floating element.

It supports **`left, right, both, none`** and **`inherit`** values. You can clear the float from an element by defining its’ clear as both

▼ CSS Overflow

If a floated element is taller than the containing element, it will "overflow" outside of its container. We can use the **`overflow`** property to define how the overflow content should behave.

This property is a shorthand for the following CSS properties:

- `overflow-x`
- `overflow-y`

This property supports the values:

```
overflow: visible;
overflow: hidden;
overflow: clip;
overflow: scroll;
overflow: auto;
overflow: hidden visible;
```

CSS Position

The position attribute supports the following values:

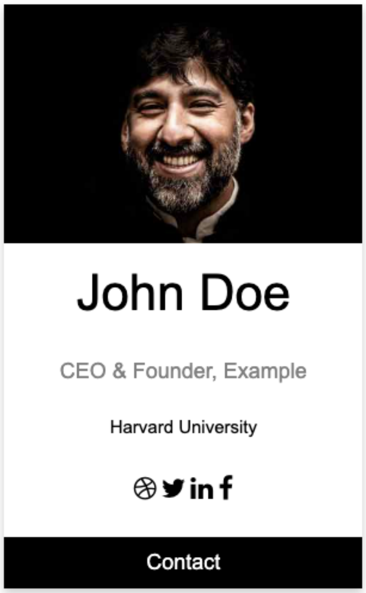
```
position: static;
position: relative;
position: absolute;
```

```
position: fixed;
position: sticky;
```

- `static`: Default
- `relative`: A relatively positioned element is an element whose computed position value is relative. The top and bottom properties specify the vertical offset from its normal position; the left and right properties specify the horizontal offset.
- `absolute`**:** The element is removed from the normal document flow, and no space is created for the element in the page layout. It is positioned relative to its closest positioned ancestor if any; otherwise, it is placed relative to the initial containing block. Its final position is determined by the values of top, right, bottom, and left.
- `sticky`**:** The element is positioned according to the normal flow of the document, and then offset relative to its nearest scrolling ancestor and containing block (nearest block-level ancestor), including table-related elements, based on the values of top, right, bottom, and left. The offset does not affect the position of any other elements.
- `fixed`**:** The element is removed from the normal document flow, and no space is created for the element in the page layout. It is positioned relative to the initial containing block established by the viewport.

Assignments

1. Make a profile card like this

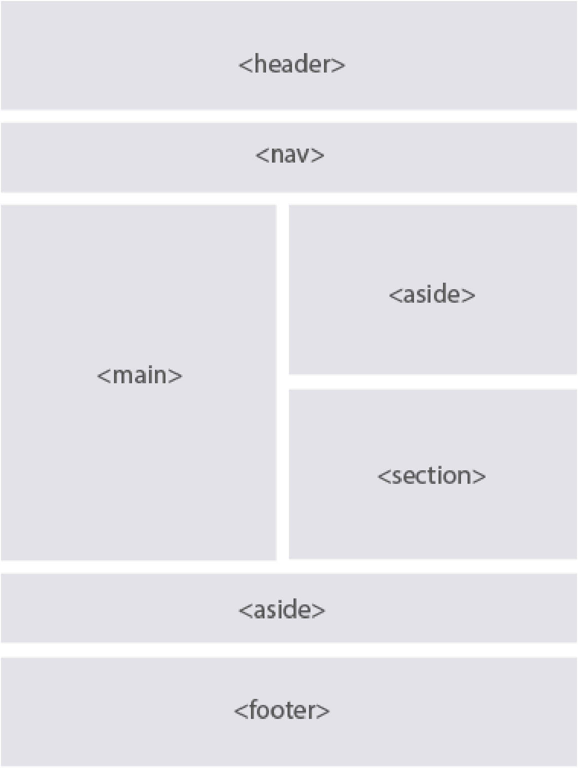


John Doe

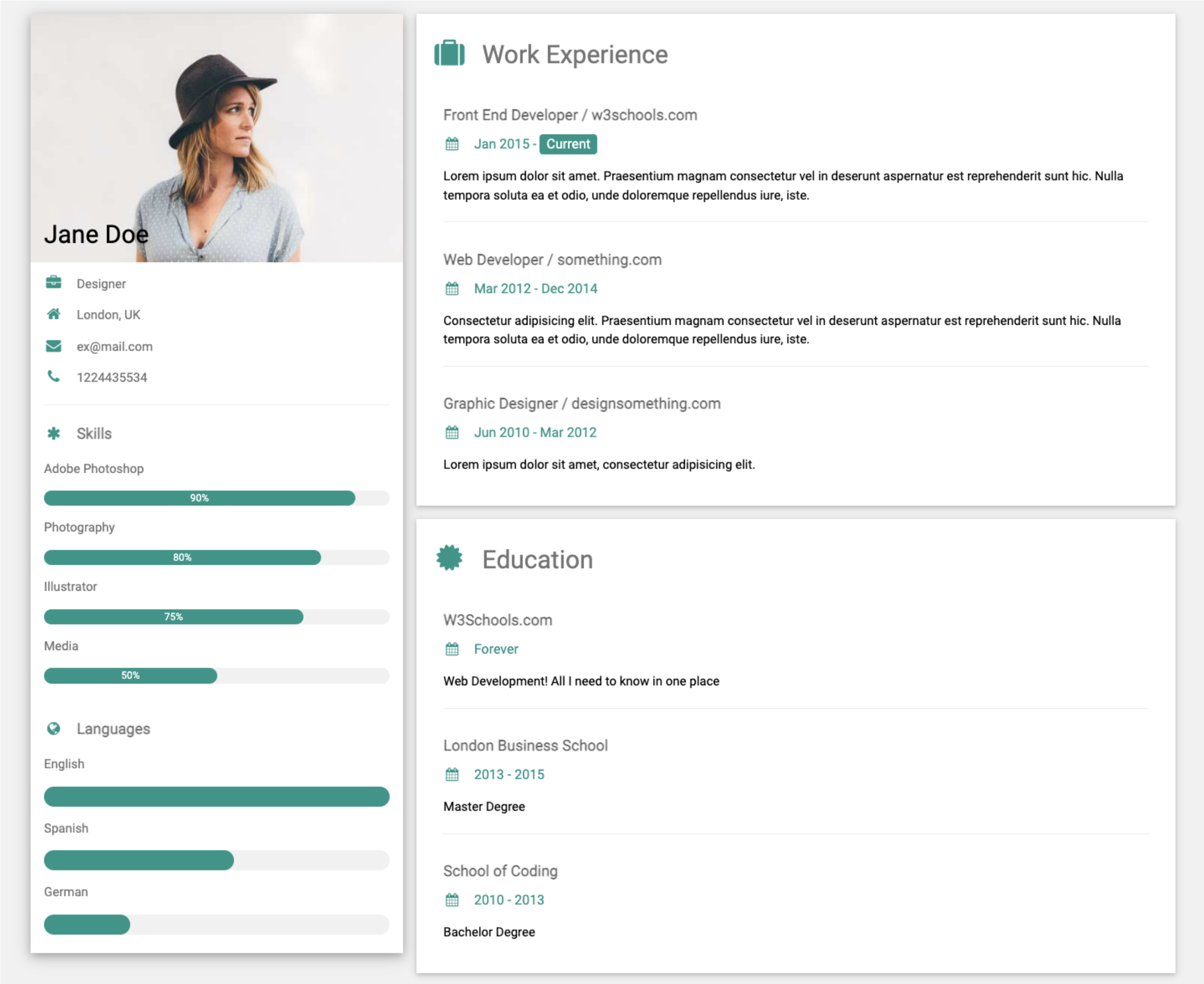
CEO & Founder, Example

Harvard University

2. Make a Layout like this one:



3. **Make a Resume Website Like this: Major Project (Might need Flex/Grid)**



Jane Doe

Designer

London, UK

ex@mail.com

1224435534

Skills

Adobe Photoshop

Photography

Illustrator

Media

Languages

English

Spanish

German

Work Experience

Front End Developer / w3schools.com

Jan 2015 - Current

Lorem ipsum dolor sit amet. Praesentium magnam consectetur vel in deserunt aspernatur est reprehenderit sunt hic. Nulla tempora soluta ea et odio, unde doloremque repellendus iure, iste.

Web Developer / something.com

Mar 2012 - Dec 2014

Consectetur adipisicing elit. Praesentium magnam consectetur vel in deserunt aspernatur est reprehenderit sunt hic. Nulla tempora soluta ea et odio, unde doloremque repellendus iure, iste.

Graphic Designer / designsomething.com

Jun 2010 - Mar 2012

Lorem ipsum dolor sit amet, consectetur adipisicing elit.

Education

W3Schools.com

Forever

Web Development! All I need to know in one place

London Business School

2013 - 2015

Master Degree

School of Coding

2010 - 2013

Bachelor Degree