[Lesson 6]

Roi Yehoshua 2018

[DAN.IT] EDUCATION

What we learnt last time?

- CSS selectors
- Selectors weight
- CSS units of measurement

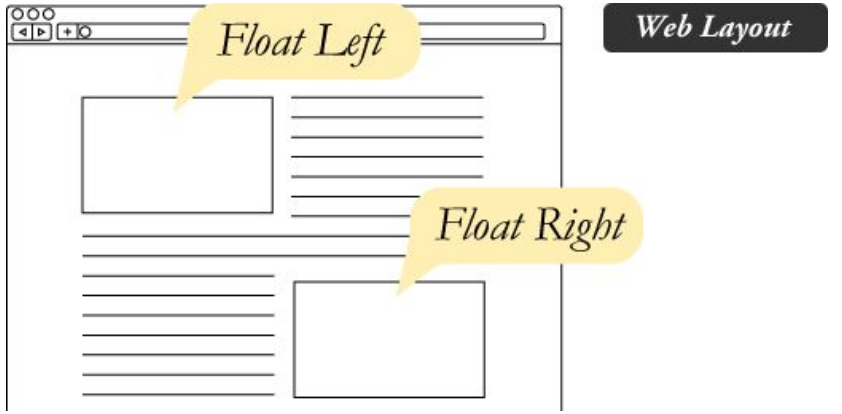
[Our targets for today]

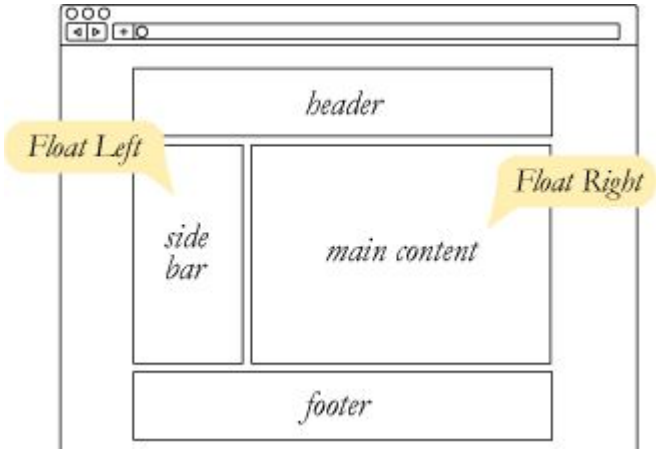
- Placing several block elements on top of one line using “float”
- How to make several columns using “float”
- Place adjacent content under floating elements using “clear: both”
- 3 different ways of including fonts into your project
- Font families: serif, sans-serif, monospace, proportional, script, fantasy
- Web safe fonts - what does it mean

[DAN.IT] EDUCATION

[Float]

- With CSS float, an element can be pushed to the left or right, allowing other elements to wrap around it
- The elements **after** the floating element will flow around it
  - The **clear** property turns off the floating
- The elements before the floating element will not be affected
- Aside from the simple example of wrapping text around images, floats can be used to create **entire web layouts**





[DAN.IT] EDUCATION

[Float]

- The following example specifies that an image should float to the **left** in a text:

```
img {
        float: left;
}
```

In this example, the image will float to the left in the paragraph, and the text in the paragraph will wrap around the image.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet, nulla et dictum interdum, nisi lorem egestas odio, vitae scelerisque enim ligula venenatis dolor. Maecenas nisl est, ultrices nec congue eget, auctor vitae massa. Fusce luctus vestibulum augue ut aliquet. Mauris ante ligula, facilisis sed ornare eu, lobortis in odio. Praesent convallis urna a lacus interdum ut hendrerit risus congue. Nunc sagittis dictum nisi, sed ullamcorper ipsum dignissim ac. In at libero sed nunc venenatis imperdiet sed ornare turpis. Donec vitae dui eget tellus gravida venenatis. Integer fringilla congue eros non fermentum. Sed dapibus pulvinar nibh tempor porta. Cras ac leo purus. Mauris quis diam velit.

- The following example specifies that an image should float to the **right** in a text:

```
img {
        float: right;
}
```

In this example, the image will float to the right in the paragraph, and the text in the paragraph will wrap around the image.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet, nulla et dictum interdum, nisi lorem egestas odio, vitae scelerisque enim ligula venenatis dolor. Maecenas nisl est, ultrices nec congue eget, auctor vitae massa. Fusce luctus vestibulum augue ut aliquet. Mauris ante ligula, facilisis sed ornare eu, lobortis in odio. Praesent convallis urna a lacus interdum ut hendrerit risus congue. Nunc sagittis dictum nisi, sed ullamcorper ipsum dignissim ac. In at libero sed nunc venenatis imperdiet sed ornare turpis. Donec vitae dui eget tellus gravida venenatis. Integer fringilla congue eros non fermentum. Sed dapibus pulvinar nibh tempor porta. Cras ac leo purus. Mauris quis diam velit.

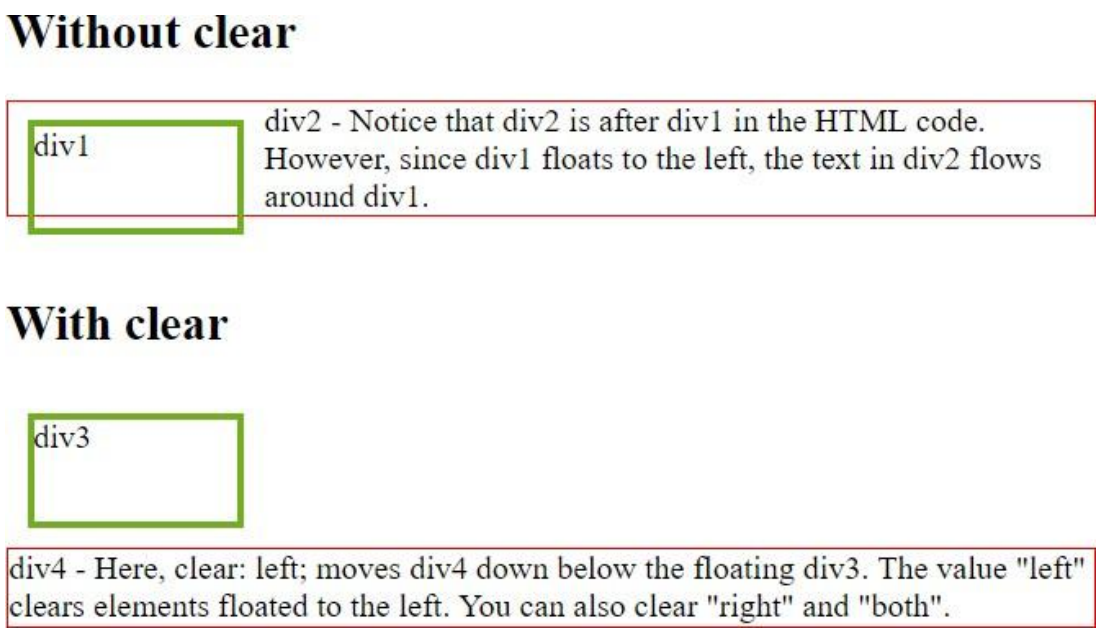
[DAN.IT] EDUCATION

The clear Property

- The clear property specifies whether an element can be next to floating elements that precede it or must be moved down (cleared) below them
- The clear property can have one of the following values:
  - none - the element is *not* moved down to clear past floating elements (default)
  - left - the element is moved down to clear past *left* floats
  - right- the element is moved down to clear past *right* floats
  - both - the element is moved down to clear past *both* left and right floats

The clear Property

```
.div1 {
        float: left;
        width: 100px;
        height: 50px;
        margin: 10px;
        border: 3px solid #73AD21;
}
.div2 {
        border: 1px solid red;
}
.div3 {
        float: left;
        width: 100px;
        height: 50px;
        margin: 10px;
        border: 3px solid #73AD21;
}
.div4 {
        border: 1px solid red;
        clear: left;
}
```

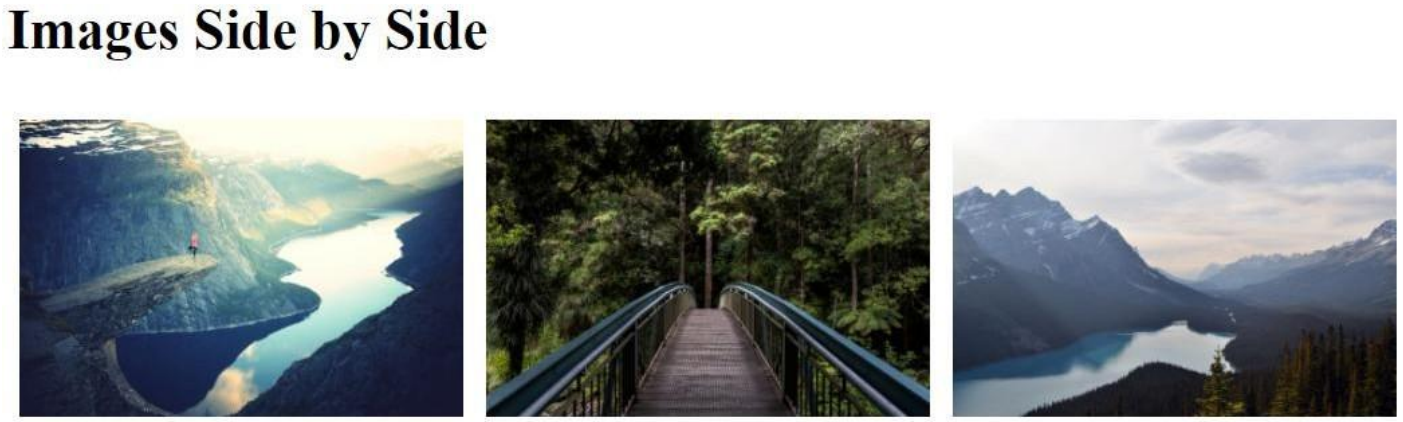


[Images Side By Side]

→ With the float property, it is easy to float images side by side:

```
.img-container {
        float: left;
        width: 33.33%;
        padding: 5px;
        box-sizing: border-box;
}

img {
        width: 100%;
}
```




```
<div class="img-container">
        <img src="images/fjords.jpg" alt="Fjords"/>
</div>
```

[DAN.IT] EDUCATION

[Exercise (1)]

- Use CSS to create an image gallery with 4 pictures
- Add a description text below each image
- When the user hovers over an image, its borders will change color to black
- Clicking on an image will open a new page displaying the image in its full size

Fjords

A forest

Northern lights

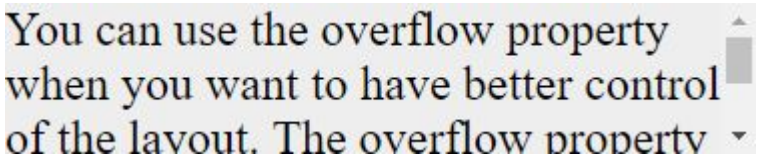
Mountains

[DAN.IT] EDUCATION

Overflow

- The overflow property controls what happens to content that is too big to fit into a specified area
- It has the following values:
  - visible - Default. The overflow is not clipped. It renders outside the element's box
  - hidden - The overflow is clipped, and the rest of the content will be invisible
  - scroll - The overflow is clipped, but a scrollbar is added to see the rest of the content
  - auto - If overflow is clipped, a scrollbar should be added to see the rest of the content
- The overflow property only works for block elements with a specified height

```
div {
        width: 250px;
        height: 50px;
        background-color: #eee;
        overflow: auto;
}
```



[Exercise (2)]

→ You are given a list of hyperlinks:

```
<ul>
      <li><a href="#home" class="active">Home</a></li>
      <li><a href="#news">News</a></li>
      <li><a href="#contact">Contact</a></li>
      <li><a href="#about">About</a></li>
</ul>
```

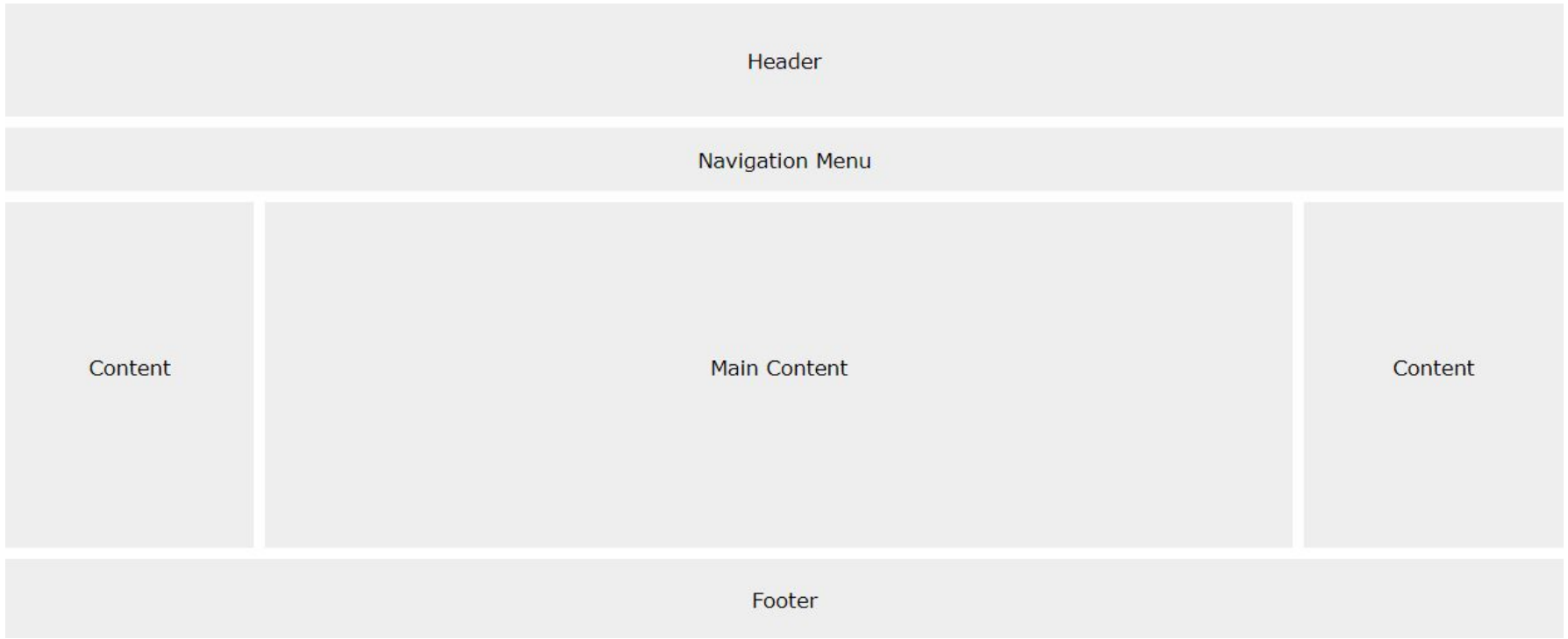
→ Use float to create the following horizontal menu:

→ Try not to specify the width of each <li> (hint: use the overflow property)

[DAN.IT] EDUCATION

[Website Layout]

→ A website is often divided into headers, menus, content and a footer:



[DAN.IT] EDUCATION

[Header]

→ A header is usually located at the top of the website (or right below a top navigation menu). It often contains a logo or the website name:

```
header {
	background-color: #f1f1f1;
	padding: 20px;
	text-align: center;
}
```


```
<header>
	<h1>Header</h1>
</header>
```

[DAN.IT] EDUCATION

[Navigation Bar]

→ A navigation bar contains a list of links to help visitors navigating through your website:

```
nav {
    overflow: hidden;
    background-color: #333;
}
    nav a {
        float: left;
        display: block;
        color: #f2f2f2;
        text-align: center;
        padding: 15px;
        text-decoration: none;
    }
        nav a:hover {
            background-color: #ddd;
            color: black;
        }
```

```
<nav>
    <a href="#">Link</a>
    <a href="#">Link</a>
    <a href="#">Link</a>
</nav>
```

[DAN.IT] EDUCATION

[Content]

- → The layout in this section, often depends on the target users
- → The most common layout is one (or combining them) of the following:
  - → **1-column** (often used for mobile browsers)
  - → **2-column** (often used for tablets and laptops)
  - → **3-column layout** (only used for desktops)

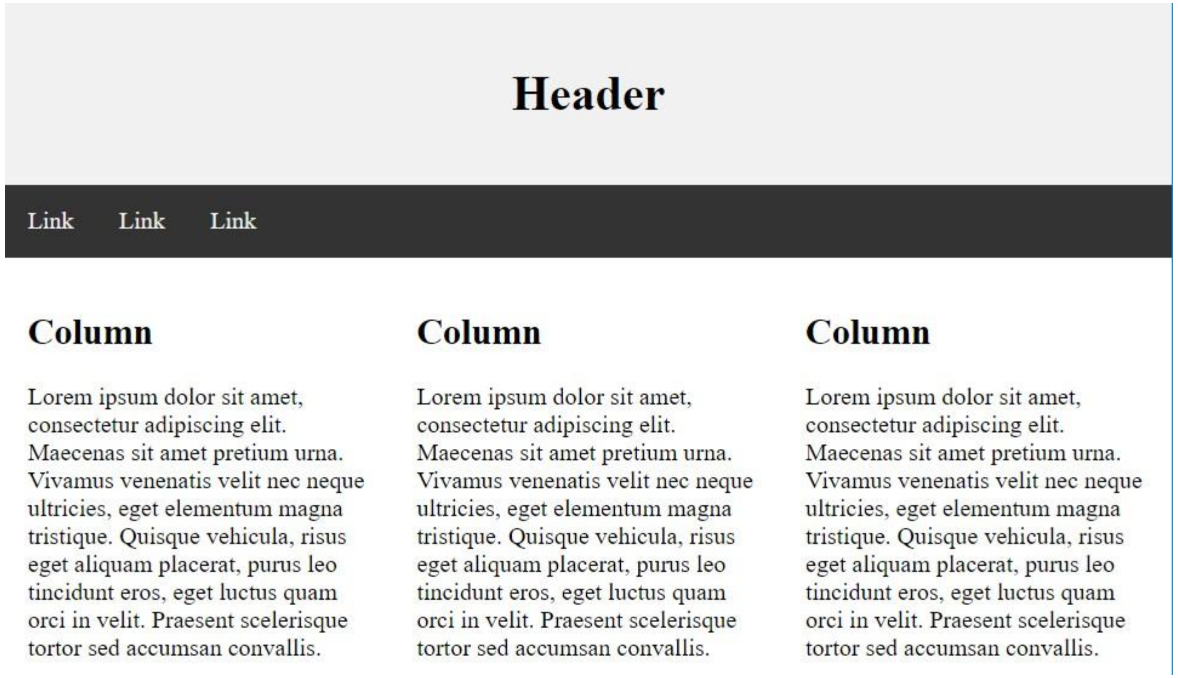
[DAN.IT] EDUCATION

Content

→ For example, we will create a 3-column layout:

```
.column {
	float: left;
	width: 33.33%;
	padding: 15px;
}
```

```
<main>
	<div class="column">
		<h2>Column</h2>
		<p>Lorem ipsum...</p>
	</div>
	<div class="column">
		<h2>Column</h2>
		<p>Lorem ipsum...</p>
	</div>
	<div class="column">
		<h2>Column</h2>
		<p>Lorem ipsum...</p>
	</div>
</main>
```



To create a 2-column layout, change the width to 50%, to create a 4-column layout, use 25%, etc.

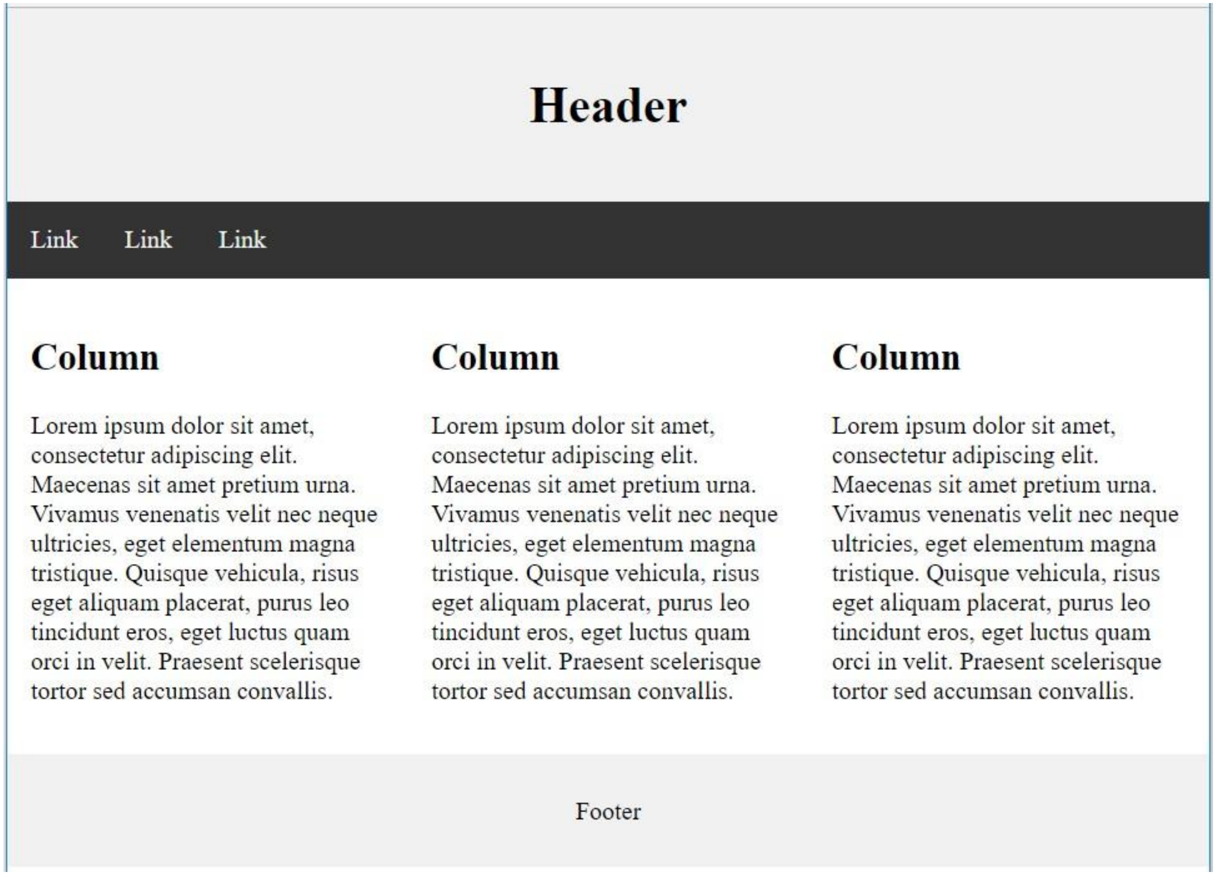
DAN.IT EDUCATION

[Footer]

→ The footer is placed at the bottom of your page. It often contains information like copyright and contact info:

```css
footer {
        background-color: #f1f1f1;
        padding: 10px;
        text-align: center;
        clear: both; /* clear the floats */
}
```

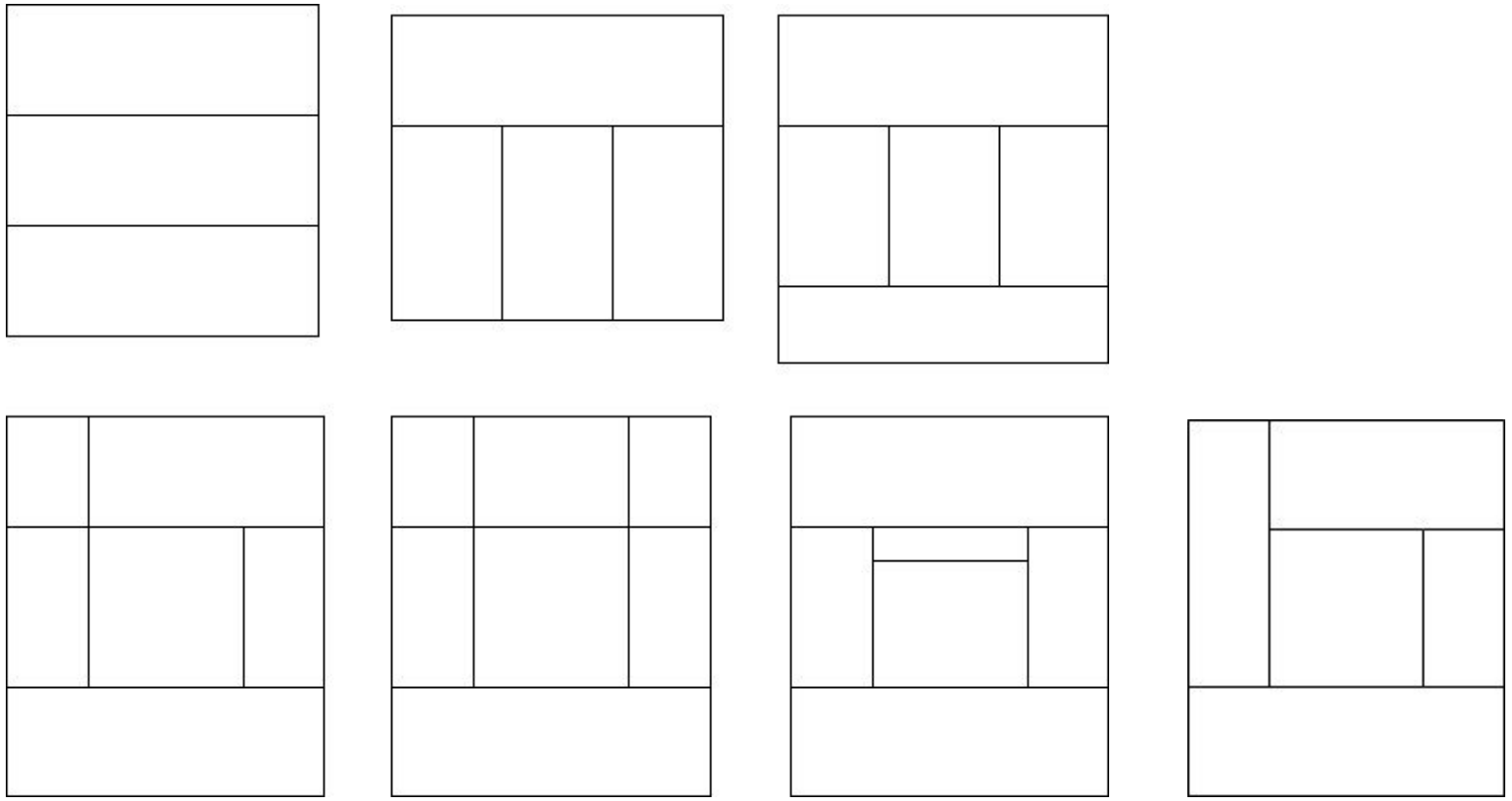
```html
<footer>
        <p>Footer</p>
</footer>
```



[DAN.IT] EDUCATION

[Exercise (3)]

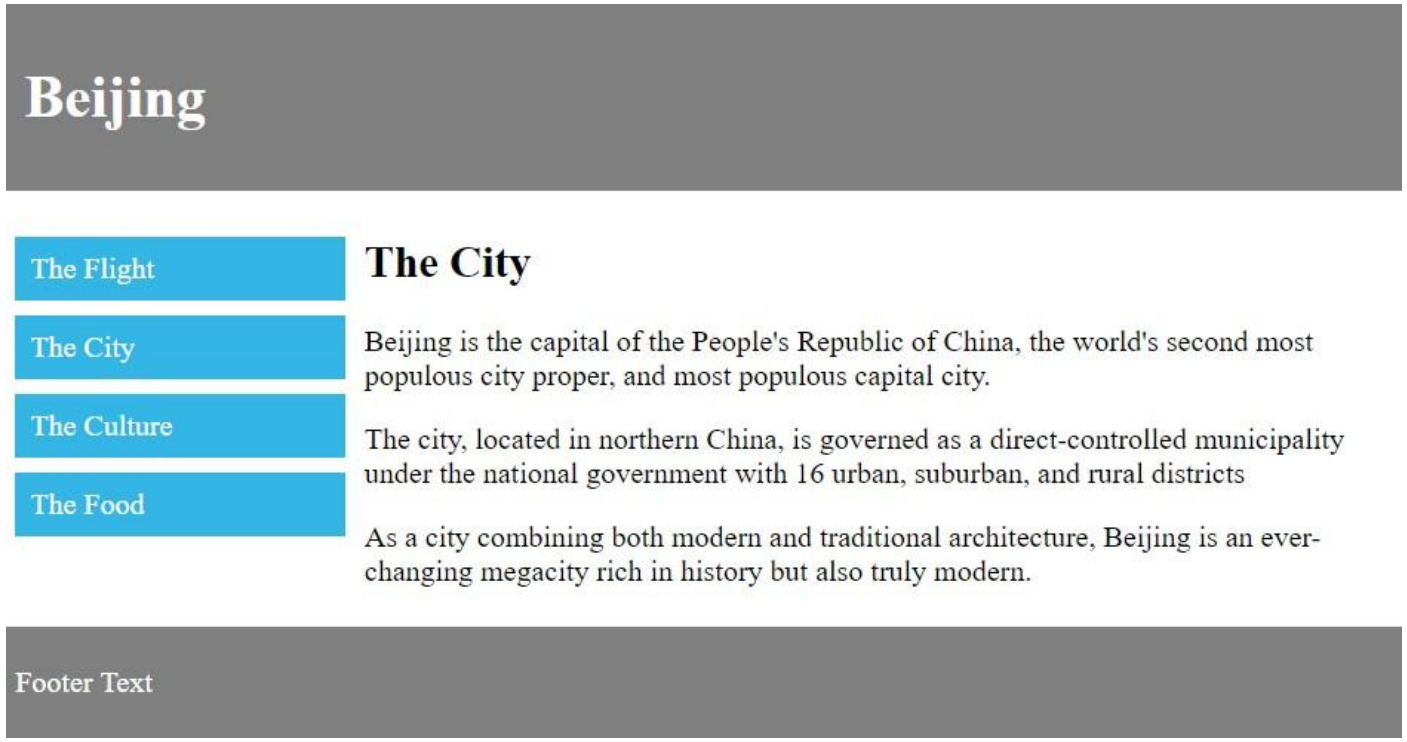
→ Create the following page layouts using only the float property:



[DAN.IT] EDUCATION

[Exercise (4)]

→ Create the following page using the page layout techniques we've learned:



[DAN.IT] EDUCATION

CSS Fonts

- The CSS font properties define the font family, boldness, size, and the style of a text
- There are two types of font family names:
  - **generic family** - a group of font families with a similar look (like "Serif" or "Monospace")
  - **font family** - a specific font family (like "Times New Roman" or "Arial")

Generic family	Font family	Description
Serif	Times New Roman Georgia	Serif fonts have small lines at the ends on some characters
Sans-serif	Arial Verdana	"Sans" means without - these fonts do not have the lines at the ends of characters
Monospace	Courier New Lucida Console	All monospace characters have the same width

- On computer screens, sans-serif fonts are considered easier to read than serif fonts.

Font Family

- The font family of a text is set with the font-family property→
- The font-family property should hold several font names as a "fallback" system
  - If the browser does not support the first font, it tries the next font, and so on.
  - Start with the font you want, and end with a generic family, to let the browser pick a similar font in the generic family, if no other fonts are available
- **Note**: If the name of a font family is more than one word, it must be in quotation marks, like: "Times New Roman"

```
p.serif {
    font-family: "Times New Roman", Times, serif;
}
p.sansserif {
    font-family: Arial, Helvetica, sans-serif;
}
```

This paragraph is shown in the Times New Roman font.

This paragraph is shown in the Arial font.

Font Style

- The font-style property is mostly used to specify italic text
- This property has three values:
  - normal - The text is shown normally
  - italic - The text is shown in italics
  - oblique - The text is "leaning" (oblique is very similar to italic, but less supported)

```
p.normal {
        font-style: normal;
}
p.italic {
        font-style: italic;
}
p.oblique {
        font-style: oblique;
}
```

This is a paragraph in normal style.

This is a paragraph in italic style.

This is a paragraph in oblique style.

[DAN.IT] EDUCATION

Font Size

- The font-size property sets the size of the text
- The font-size value can be an absolute (using px or pt), or relative size (using em or %)
- To maximize accessibility, it is generally best to use values that are relative to the user's default font size
- To allow users to resize the text (in the browser menu), many developers use em
  - 1em is equal to the current font size
  - The default text size in browsers is 16px. So, the default size of 1em is 16px.
  - The size can be calculated from pixels to em using this formula: *pixels*/16=*em*

```
h1.larger {
      font-size: 2.5em; /* 40px/16=2.5em */
}
```

This heading has a default font size (2em)

This heading has a larger font size (2.5em)

[DAN.IT] EDUCATION

Font Weight

- The font-weight property specifies the weight of a font. Possible values:
  - Normal - Normal font weight. Same as 400.
  - Bold - Bold font weight. Same as 700.
  - Lighter - One font weight lighter than the parent element (among the available weights of the font)
  - Bolder - One font weight heavier than the parent element (among the available weights of the font)
  - 100, 200, 300, 400, 500, 600, 700, 800, 900 - Numeric font weights
- Some fonts are only available in normal and bold

```
p.normal {
        font-weight: normal;
}
p.light {
        font-weight: lighter;
}
p.thick {
        font-weight: bold;
}
p.thicker {
        font-weight: 900;
}
```

This is a paragraph.

This is a paragraph.

This is a paragraph.

This is a paragraph.

[DAN.IT] EDUCATION

Font Style

- The font-style property is mostly used to specify italic text
- This property has three values:
  - normal - The text is shown normally
  - italic - The text is shown in italics
  - oblique - The text is "leaning" (oblique is very similar to italic, but less supported)

```
p.normal {
        font-style: normal;
}
p.italic {
        font-style: italic;
}
p.oblique {
        font-style: oblique;
}
```

This is a paragraph in normal style.

This is a paragraph in italic style.

This is a paragraph in oblique style.

[DAN.IT] EDUCATION

Using custom font

- The @font-face directive allows using custom fonts which are not present in user's OS
- This directive has two mandatory values:
  - font-family - name of the font that will be used across the project
  - src - path to the file with font
- Optional properties:
  - font-style - style used in this particular file (e.g. italic)
  - font-weight - weight of the font in file (100-900 or lighter-bold)
  - Other: unicode-range, font-variant, font-feature-settings, font-variation-settings, font-stretch

```
@font-face {
    font-family: MyUniqueFont;
    src: url('fonts/MyUniqueFont.ttf');
}


p.normal {
        font-family: MyUniqueFont;
}
```

[DAN.IT] EDUCATION

Google fonts

- Google Fonts service provides the easiest way to include custom font from the available list into the project
- https://fonts.google.com/
- To include the needed font, select it, choose font properties (font-weight, language), and link to project using <link> in HTML file or @import in CSS

```
<link href="https://fonts.googleapis.com/css?family=Lato:400,700,900i" rel="stylesheet">

OR

@import url('https://fonts.googleapis.com/css?family=Lato:400,700,900i');
```

```
body {
      font-family: Lato, sans-serif;
}
```

[DAN.IT] EDUCATION

CSS Icons

- The simplest way to add an icon to your page is with an icon library, such as Font Awesome
- Add the name of the specified icon class to any inline HTML element (like)
  - No downloading or installation is required!

```
<!DOCTYPE html>
<html>
<head>
	<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/4.7.0/css/font-awesome.min.css">
</head>
<body>
	<span class="fa fa-heart"></span>
	<span class="fa fa-car"></span>
	<span class="fa fa-file"></span>
	<span class="fa fa-bars"></span><br/>
	<span class="fa fa-cloud"></span>
	<span class="fa fa-cloud" style="font-size:24px;color:red;"></span>
	<span class="fa fa-cloud" style="font-size:36px;color:lightblue;"></span>
</body>
</html>
```

[DAN.IT] EDUCATION

[Control questions]

1. What is the default value of “display” property for elements with “float: right”?
2. How floating elements affect height of parent element?
3. How elements going after tags with “float” property are displayed?
4. How does “clear: right” work?
5. Which element has to contain “clear: both” if you want to fix the problem with overlapping floating elements?
6. What fonts do we call “web safe”?
7. What is the difference between serif and sans-serif fonts?
8. How can you add custom font on a page?

[DAN.IT EDUCATION]