

Відкривши HTML-документ з архіву, ви побачите готовий каркас посадкової сторінки. У тезі `<head>` ми підключили таблиці стилів, які знадобляться вам для роботи:

**normalize.css** - файл для скидання стандартних стилів браузера (ми докладно розповідали про нього в одному з уроків першої частини підручника);

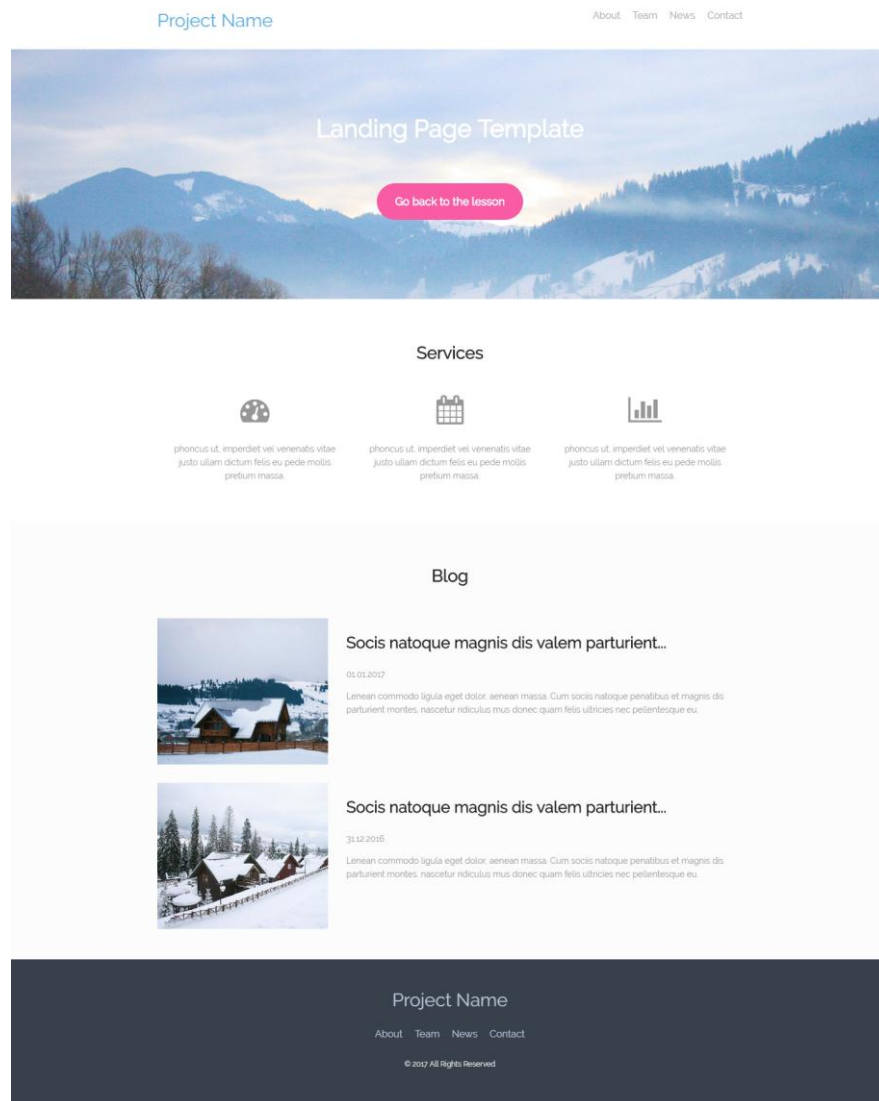
**шрифт Raleway** від Google Fonts;

іконочні шрифт **FontAwesome**;

**style.css** - файл, куди ви будете записувати всі стилі для Лендінзі.

Тіло документа розділено на секції - шапка сайту (**header**) з навігацією (**nav**), великий блок на всю ширину (**jumbotron**), блок сервісів (**services**), записи блогу (**blog**) і підвал сайту (**footer**). Кожна секція містить відповідний контент.

Макет сторінки виглядає наступним чином:



## Таблиця стилів

Відкрийте файл **style.css** з папки «css». Ви побачите, що документ структурований за допомогою коментарів - для зручності ми розділили його на частини відповідно до структури HTML-сторінки.

Записуйте кожен стиль у відповідний розділ: стилі для шапки сайту розміщуйте під коментарем `/* Header */`, стилі для блогу - під коментарем `/* Blog */`, базові стилі і стилі для повторюваних елементів сторінки - під коментарем `/* General */` і т. д.

### Основні стилі CSS

На нашій сторінці є повторювані елементи, а також теги, яким потрібно задати певний стиль, після чого він буде застосований до всіх елементів відразу.

Почнемо з основного тексту на сайті.

Згідно дизайну макета, основний (і єдиний) шрифт на сторінці - це **Raleway**, а всі параграфи `<p>` повинні мати розмір шрифту 14 пікселів, колір #999 і висоту рядка 1.6. Запишемо ці дані в нашу таблицю стилів, в розділ **General**:

```
body {
    font-family: Raleway, Arial, Helvetica, sans-serif; /* крім
Raleway додайте кілька запасних шрифтів */
}
p {
    font-size: 14px;
    color: #999;
    line-height: 1.6;
}
```

У нас є один заголовок `<h1>`, а також кілька заголовків `<h2>` і `<h3>`. Згідно макету, заголовок `h1` має розмір шрифту 40 пікселів і білий колір. Решта заголовків мають колір #282828. Розмір шрифту для заголовків другого рівня - 28 пікселів, а для `h3` - 26 пікселів.

```
h1 {
    font-size: 40px;
    color: #fff;
}
h2 {
    font-size: 28px;
}
h3 {
    font-size: 26px;
}
h2, h3 {
    color: #282828;
}
```

Крім тексту у нас є ще кілька повторюваних елементів. Наприклад, клас `.container` - це блок, з обмеженою шириною, в який поміщається контент. Ширина нашого контенту повинна займати 90% екрану, але при цьому контейнер не повинен розтягуватися ширше, ніж 960

пікселів. Тут нам стане в нагоді властивість `max-width`. Крім того, цей контейнер повинен розміщуватися по центру сторінки, для чого ми скористаємося властивістю `margin`.

```
.container {
    width: 90%;
    max-width: 960px;
    margin-left: auto;
    margin-right: auto;
}
```

Наступні повторювані елементи - блок з класом `.brand` і навігаційне меню.

Ці елементи присутні в хедері і футері. Вони не будуть виглядати абсолютно ідентично, проте частина стилів для них будуть загальними.

Отже, обидва елементи `.brand` матимуть однаковий розмір шрифту:

```
.brand {
    font-size: 30px;
}
```

А обидва навігаційних меню будуть горизонтальними і теж матимуть однаковий розмір шрифту. До речі, для створення горизонтального меню ми задіємо властивість `float`, застосувавши її до пунктів меню:

```
.menu li {
    float: left;
    font-size: 16px;
    list-style: none; /* забираємо стандартні маркери */
}
.menu {
    padding-left: 0; /* забираємо стандартний відступ зліва списку */
    margin: 0; /* забираємо стандартні поля списку */
}
```

Заберемо з навігації стандартне підкреслення посилань:

```
.menu li a {
    text-decoration: none;
}
```

З основними стилями ми розібралися. Збережіть зміни в таблиці CSS і оновіть файл `index.html` в браузері. Ви вже можете бачити, як почала видозмінюватися веб-сторінка. Тепер час перейти до оформлення секцій.

### CSS для шапки сайту

Наш хедер матиме білий фон і відступ зверху в розмірі 30 пікселів. Запишемо це в таблицю стилів:

```
header {
    background-color: #fff;
    padding-top: 30px;
}
```

Блок з назвою проекту буде приліплений до лівої сторони контейнера, мати колір # 59abe3 і відступ внизу в 30 пікселів:

```
header .brand {
    float: left;
    color: #59abe3;
    margin-bottom: 30px;
}
```

Зверніть увагу, що ми записали стиль саме для того елемента .brand, який знаходиться в шапці сайту, і не торкалися .brand, який розташований в футері, оскільки він матиме інший колір і розташування. Надалі ми будемо продовжувати використовувати переваги селекторів нащадків.

Навігаційне меню ми приклеїмо до правого боку шапки, додамо внизу поле висотою 30 пікселів, а також зробимо відступи по 20 пікселів зліва для всіх пунктів меню, щоб вони не зливалися в одне ціле:

```
header .menu {
    float: right;
    margin-bottom: 30px;
}
header .menu li {
    padding-left: 20px;
}
```

Встановимо колір тексту #aaa для пунктів меню.

Цей стиль ми застосуємо саме до тегів <a>, а не до <li>, в іншому випадку ми не побачимо результат. Справа в тому, що тег <a> не буде наслідувати колір тексту від батька, і посилання продовжать відображатися в стандартному синьому кольорі.

```
header .menu li a {
    color: #aaa;
}
```

## CSS для блоку Jumbotron

Як правило, jumbotron (джамботрон) на сайті - це великий, розтягнутий на всю ширину сторінки блок. Фоном для нашого джамботрона послужить зображення в хорошій якості, розтягнуте на всю ширину, а текстовий вміст буде перебувати в обмежуючому блоці .container.

Фонове зображення ви знайдете в папці «img».

```
.jumbotron {
    background-image: url(../img/jumbotron.jpg);
    background-size: cover;
    background-position: top center;
    background-repeat: no-repeat;
}
```

Ми додамо ще кілька рядків до стилю джамботрона - для створення вертикальних відступів по 80 пікселів і установки мінімальної висоти в 250 пікселів, для вирівнювання внутрішнього

вмісту по центру блоку, а також для скидання обтікання, яке було задано навігаційному меню:

```
.jumbotron {
    background-image: url(../img/jumbotron.jpg);
    background-size: cover;
    background-position: top center;
    background-repeat: no-repeat;
    padding-top: 80px;
    padding-bottom: 80px;
    min-height: 250px;
    text-align: center;
    clear: both;
}
```

Стиль для заголовка джамботрона ми додали раніше, тому відразу перейдемо до створення яскравої кнопки, розташованої під ним. Якщо бути точніше, це звичайне посилання з класом `.button`.

По-перше, оскільки тег `<a>` є рядковим елементом, то першим ділом застосуємо до нього властивість `display` із значенням `inline-block`. Таким чином кнопка буде відображатися як блок, а поведінка залишиться, як у рядка.

Текст кнопки буде білим і без підкреслення, шрифт - жирний 18 пікселів, верхній `margin` - 40 пікселів, а колір фону - `#22a7f0`. Щоб це більше було схоже на кнопку, необхідно додати відступи з чотирьох сторін - в нашому випадку по 20 пікселів вертикально і по 30 пікселів горизонтально. Також ми округлимо кути елемента за допомогою властивості `border-radius`:

```
.jumbotron .button {
    display: inline-block;
    color: #fff;
    text-decoration: none;
    font-weight: bold;
    font-size: 18px;
    margin-top: 40px;
    background-color: #fa5ba5;
    padding: 20px 30px;
    border-radius: 30px;
}
```

### CSS для блоку Services

Дана секція буде включати в себе три блоки, розташованих в один ряд. У цьому нам знову допоможе властивість `float`. Всі блоки матимуть однакову ширину, зазначену в процентах, і в сумі вони займуть всю ширину контейнера. Розділивши число 100 на 3, отримуємо 33,333333. Саме його ми записуємо в стиль для класу `.service`. Додатково задамо кожному блоку внутрішні відступи в розмірі 15 пікселів і зовнішні вертикальні відступи в 30 і 50 пікселів:

```
.services .service {
    width: 33.333333%;
    float: left;
    padding-left: 15px;
```

```
padding-right: 15px;
margin-top: 30px;
margin-bottom: 50px;
}
```

Зараз ви можете оновити сторінку і побачити, як третій блок з'їхав вниз. Здогадаєтеся, чому так сталося? Згадайте заняття, де ми розбирали, як браузер обчислює розміри елемента і що робить властивість `box-sizing`. Ми додали нашим блокам властивість `padding`, і сума їх ширин перевищила ширину контейнера, тому один з них не поміщається в ряд. Додайте до стилю елемента `.service` властивість `box-sizing` зі значенням `border-box`, після чого оновіть сторінку в браузері. Ви побачите, що тепер всі блоки вишикувалися так, як треба.

Додамо стилі для самого блоку `.services` - зробимо верхній відступ в 50 пікселів і центруємо весь текстовий вміст:

```
.services {
  padding-top: 50px;
  text-align: center;
}
```

Як іконки блоків ми використовували іконочний шрифт FontAwesome. Ми змінимо колір іконок на більш відповідний нам. Кожна ікона представляє собою векторний малюнок, перефарбувати який не складе труднощів. Ми просто запишемо стиль, як ніби задаємо колір для тексту:

```
.services .fa {
  color: #979797;
  margin-bottom: 15px; /* и добавим небольшой отступ */
}
```

## CSS для блоку Blog

Перш за все необхідно скасувати обтікання для блоку `.blog`. Він буде мати вертикальні внутрішні відступи в 50 пікселів, а також легкий сірий фон, відповідно до макету:

```
.blog {
  clear: both;
  padding-top: 50px;
  padding-bottom: 50px;
  background-color: #fcfcfc;
}
```

Вирівнюємо заголовок цього блоку по центру:

```
.blog h2 {
  text-align: center;
}
```

Кожному блоку з класом `.post` дамо внутрішній відступ в 30 пікселів зверху і поведінку рядково-блочного елемента:

```
.blog .post {
  padding-top: 30px;
```

```
    display: inline-block;
}
```

Для мініатюр постів задамо обтікання і зовнішній відступ 30 пікселів з правого боку:

```
.blog .post img {
    float: left;
    margin-right: 30px;
}
```

Після чого доповніть стиль елементів .post рядком clear: both, завдяки якому кожен блок буде відображатися коректно і не з'їжджати.

### CSS для футера

Для основного блоку підвалу знадобиться скинути обтікання, задати вертикальні відступи в 50 пікселів, вирівнювання тексту по центру і темно-сірий фон:

```
footer {
    clear: both;
    padding-top: 50px;
    padding-bottom: 50px;
    text-align: center;
    background-color: #38404b;
}
```

Встановимо колір тексту для елемента .brand (як уже згадувалося, він буде відрізнятися від стилю, який застосовується до елемента .brand в шапці сайту):

```
footer .brand {
    color: #bcc9dd;
}
```

Для пунктів меню в футері задамо відступи і колір посилань:

```
footer .menu li {
    padding-left: 10px;
    padding-right: 10px;
}
footer .menu li a {
    color: #bcc9dd;
}
```

Зверніть увагу: щоб властивість text-align: center, застосована до тегу <footer>, вплинула на блок з меню, необхідно зробити його рядково-блоковим:

```
footer .menu {
    display: inline-block;
    margin-top: 30px; /* также добавим отступы */
    margin-bottom: 15px;
}
```

І, нарешті, напишемо невеликий стиль для останнього елемента - .copyright:

```
footer .copyright {
    font-size: 12px;
    color: #eee;
}
```

## Адаптивність

Давайте перевіримо, наскільки наша верстка добре адаптується під різні розміри екрану.

Перше, що кидається в очі - це три блоки `.service`, кожен з яких має ширину 33,333333% від ширини контейнера. На вузьких екранах текст в цих блоках стає важко читати.

У медіазапиті можна вказати, при якій ширині екрану який стиль відображати.

Отже, наші блоки починають виглядати погано на екранах шириною менше 576 пікселів (в основному, це телефони в портретній орієнтації). Запишемо спеціальний стиль для даної ширини (помістіть цей запис в самий низ таблиці стилів):

```
@media (max-width: 575px) {  
  .services .service {  
    width: 100%;  
    float: none;  
  }  
}
```

Тепер, якщо ширина екрана користувача буде менш ніж 576 пікселів, кожен блок `.service` матиме ширину 100% і не мати обтікання.

Продовжуємо перевіряти нашу верстку на адаптивність. Чітко видно, що на маленькі екрани нормально не поміщаються блоки `.post`. Мініатюра поста надто велика для невеликого екрану і наступний за нею текст некрасиво обтікає її.

Виправимо це наступним стилем (просто додайте код до вже створеного медіазапиту):

```
.blog .post img {  
  float: none;  
  display: block;  
  margin-right: auto;  
  margin-left: auto;  
}  
.blog .post {  
  max-width: 280px;  
  margin-left: auto;  
  margin-right: auto;  
  display: block;  
}
```

Наостанок додамо ще пару рядків коду, які допоможуть усім зображенням на сторінці стати адаптивними. Увага: цей стиль додаємо в розділ `General`!

```
img {  
  display: block;  
  max-width: 100%;  
  height: auto;  
}
```