

УЕБ ДИЗАЙН

УПРАЖНЕНИЕ 11

ОПИСАНИЕ НА ПРОЕКТА

Проект: Уеб сайт с RWD и flex технология. Реализира се с flex решетка и медийни заявки за четири вида устройства – телефон, таблет, компютър и HD компютър.

За изпълнение на проекта е избрана цветова схема в синьо и свободно разпространявани изображения.

Упражнението включва знания от [лекция 27](#) и [лекция 25](#).

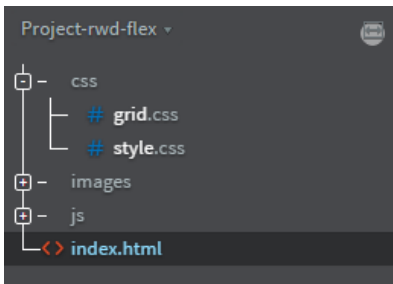
ОПИСАНИЕ НА ПРОЕКТА

Структура на проекта

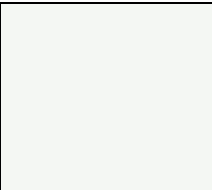
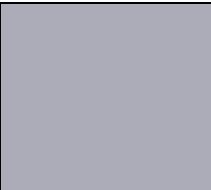




Проектът е със заглавие **Project-rwd-flex-site**. Съдържа:

1. Папка **images** – изображения за фон и проекти
2. Папка **css**
 - 2.1. стилизиращ файл **style.css**
 - 2.2. **grid.css** с flex решетка и класове
3. Папка **js**
 - 3.1. JavaScript файл **jquerymenu.js** за контролиране на менюто
 - 3.2. Папка **FlowTypeJS** с JavaScript файл за преоразмеряване на текста за различни устройства
4. Начална страница **index.html** с HTML5

Файлът **style.css** е поставен в секция **<head>** на началната страница **index.html**. В **style.css** на първи ред е свързването с **grid.css** (`@import "grid.css";`).



Цветова схема на проекта

					
Snowwhite снежнобял	Gray Сив	Darkgray Тъмносив	Orange Оранжев	Blue Син	Darkblue Тъмносин

```
:root{
  --snowwhite: rgb(244,247,243);
  --gray: rgb(171,172,184);
  --darkgray: rgb(88,89,96);
  --orange: rgb(255,153,0);
  --blue: rgb(17,89,159);
  --darkblue: rgb(43,60,91);
}
```

Използват се променливи и RGB стойности за описание на цветовете. Описани са в псевдо-клас root.

Цветовете се описват в **style.css** с класове за текста (`.text-цвят`) и класове за фона (`.bgr-цвят`).

HTML

CSS

style.css

```
31 .text-white{ color: white; }
32 .text-gray{ color: var(--gray); }
33 .text-orange{ color: var(--orange); }
34 .text-blue{ color: var(--blue); }
35 .text-darkblue{ color: var(--darkblue); }
36
37 .bgr-white{ background-color: white; }
38 .bgr-snow{ background-color: var(--snowwhite); }
39 .bgr-gray{ background-color: var(--gray); }
40 .bgr-blue{ background-color: var(--blue); }
41 .bgr-orange{ background-color: var(--orange); }
42 .bgr-darkblue{ background-color: var(--darkblue); }
43 .overlay-blue{ background-color: rgba(43,60,91,0.6); }
```

Външни и вътрешни отстояния

Външни (margin) и вътрешни (padding) отстояния са описани в **grid.css** с класове. Името на класа показва къде е отстоянието и с какъв размер е.

В името на класа:

m - означава външно отстояние (margin)

p - означава вътрешно отстояние (padding)

без други букви за всички страни

t – отстояние отгоре

b – отстояние отдолу

l – отстояние отляво

r – отстояние отдясно

0 – нулево отстояние

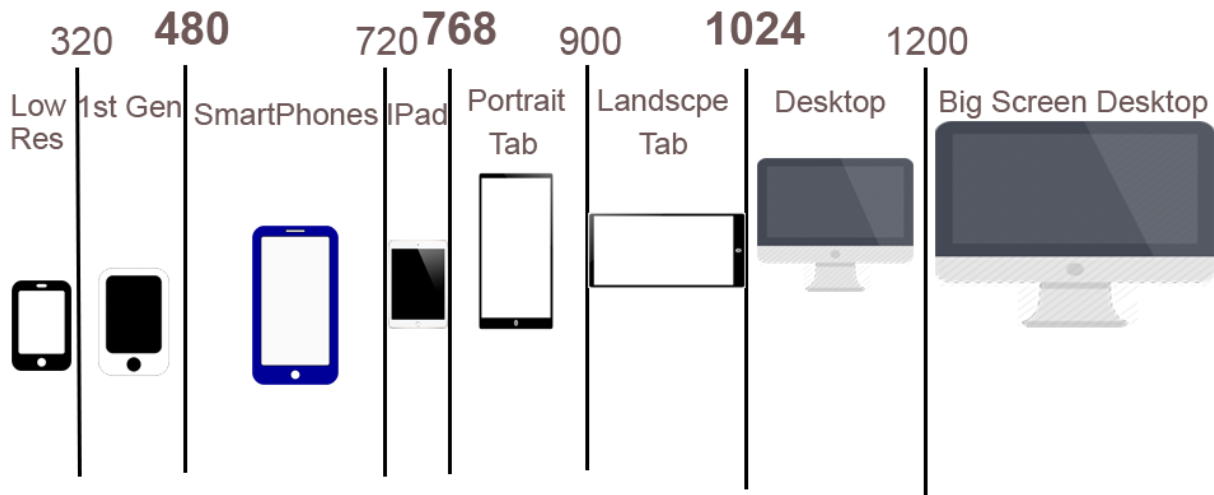
1 – отстояние 0.5rem

2 – отстояние 1.5rem

Клас **.plr-1col** дава вътрешно отстояние една колона 8.33% за компютър и 3% за телефон и таблет.

Решетка описана с flex

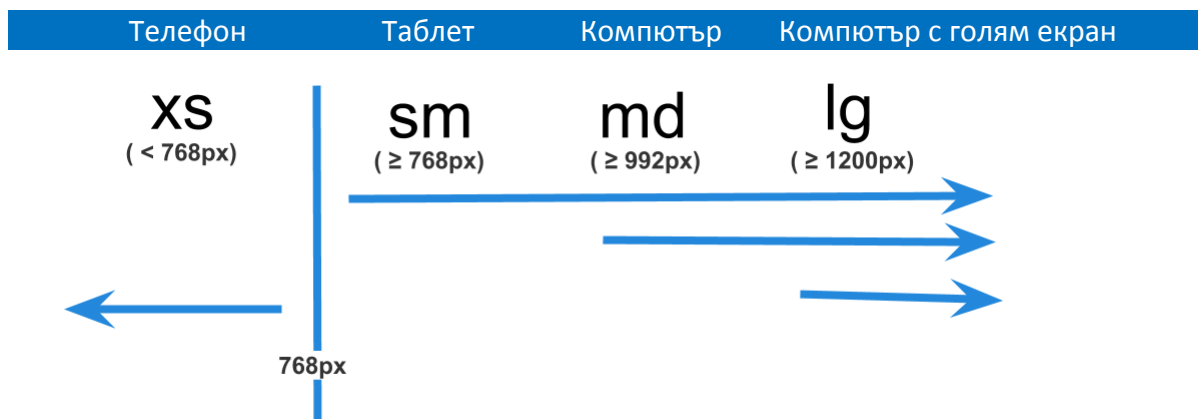
Гъвкавата решетка описва броя на заетите колони от даден елемент. В проекта има 4 решетки за компютър с голям екран, компютър с малък екран, таблет и телефон.



телефон.

Ширина на екрана на различните устройства

Класовете на медийните заявки използват наставки към `.col-`, които задават даденото устройство. Цифрата в края на класа задава заетите колони.



Наставки и точки на прекъсване (breakpoints) за медийни заявки

Ширината на колоната се задава с свойство `flex`.

ПРИМЕР: Заети 3 колони за телефон

Internet Explorer без разширяване без стесняване ширина

```
.col-xs-3 {-ms-flex: 0 0 25%; flex: 0 0 25%;}
```

Класове и заети колони

1	2	3	4	5	6	7	8	9	10	11	12
.col-xx-1											
.col-xx-2											
.col-xx-3											
.col-xx-4											
.col-xx-5											
.col-xx-6											
.col-xx-7											
.col-xx-8											
.col-xx-9											
.col-xx-10											
.col-xx-11											
.col-xx-12											

На мястото на xx се записват инициалите на устройството (xs, sm, md, lg). При този тип решетка блоковете с структурата на сайта се долепват един до друг.

Класове от решетката за описание на блокове луфт между тях

Клас за един блок с ширина 94%

.one-xx	
---------	--

Клас за два блока с ширина 47%

.two-xx		.two-xx
---------	--	---------

Клас за три блока с ширина 30%

.three-xx		.three-xx		.three-xx
-----------	--	-----------	--	-----------

Клас за четири блока с ширина 20%

.four-xx		.four-xx		.four-xx		.four-xx
----------	--	----------	--	----------	--	----------

На мястото на xx стоят инициалите на даденото устройство (xs, sm, md, lg).

Класове на flex технология за RWD

Клас **.flex**

Flex контейнер

Клас **.flex-row**

Flex елемент 1	Flex елемент 2	Flex елемент 3	
----------------	----------------	----------------	--

```
<div class="flex flex-row">  
  <div class="p-1">Flex елемент 1</div>  
  <div class="p-1">Flex елемент 2</div>  
  <div class="p-1">Flex елемент 3</div>  
</div>
```

Клас **.flex-col**

Flex елемент 1
Flex елемент 2
Flex елемент 3

```
<div class="flex flex-col">  
  <div class="p-1">Flex елемент 1</div>  
  <div class="p-1">Flex елемент 2</div>  
  <div class="p-1">Flex елемент 3</div>  
</div>
```

Клас .h-center

	Flex элемент 1	Flex элемент 2	Flex элемент 3	
--	----------------	----------------	----------------	--

```
<div class="flex h-center">  
  <div class="p-1">Flex элемент 1</div>  
  <div class="p-1">Flex элемент 2</div>  
  <div class="p-1">Flex элемент 3</div>  
</div>
```

Клас .h-between

Flex элемент 1		Flex элемент 2		Flex элемент 3
----------------	--	----------------	--	----------------

```
<div class="flex h-between">  
  <div class="p-1">Flex элемент 1</div>  
  <div class="p-1">Flex элемент 2</div>  
  <div class="p-1">Flex элемент 3</div>  
</div>
```

Клас .h-around

	Flex элемент 1		Flex элемент 2		Flex элемент 3	
--	----------------	--	----------------	--	----------------	--

```
<div class="flex h-around">  
  <div class="p-1">Flex элемент 1</div>  
  <div class="p-1">Flex элемент 2</div>  
  <div class="p-1">Flex элемент 3</div>  
</div>
```

Клас **.v-center**

Flex элемент 1	Flex элемент 2	Flex элемент 3
----------------	----------------	----------------

```
<div class="flex v-center">  
  <div class="p-1">Flex элемент 1</div>  
  <div class="p-1">Flex элемент 2</div>  
  <div class="p-1">Flex элемент 3</div>  
</div>
```

Клас **.v-end**

Flex элемент 1	Flex элемент 2	Flex элемент 3
----------------	----------------	----------------

```
<div class="flex v-end">  
  <div class="p-1">Flex элемент 1</div>  
  <div class="p-1">Flex элемент 2</div>  
  <div class="p-1">Flex элемент 3</div>  
</div>
```

Клас **.v-stretch**

Flex элемент 1	Flex элемент 2	Flex элемент 3	
----------------	----------------	----------------	--


```
<div class="flex v-stretch">
  <div class="p-1">Flex элемент 1</div>
  <div class="p-1">Flex элемент 2</div>
  <div class="p-1">Flex элемент 3</div>
</div>
```

Клас **.v-center** и **.h-center**

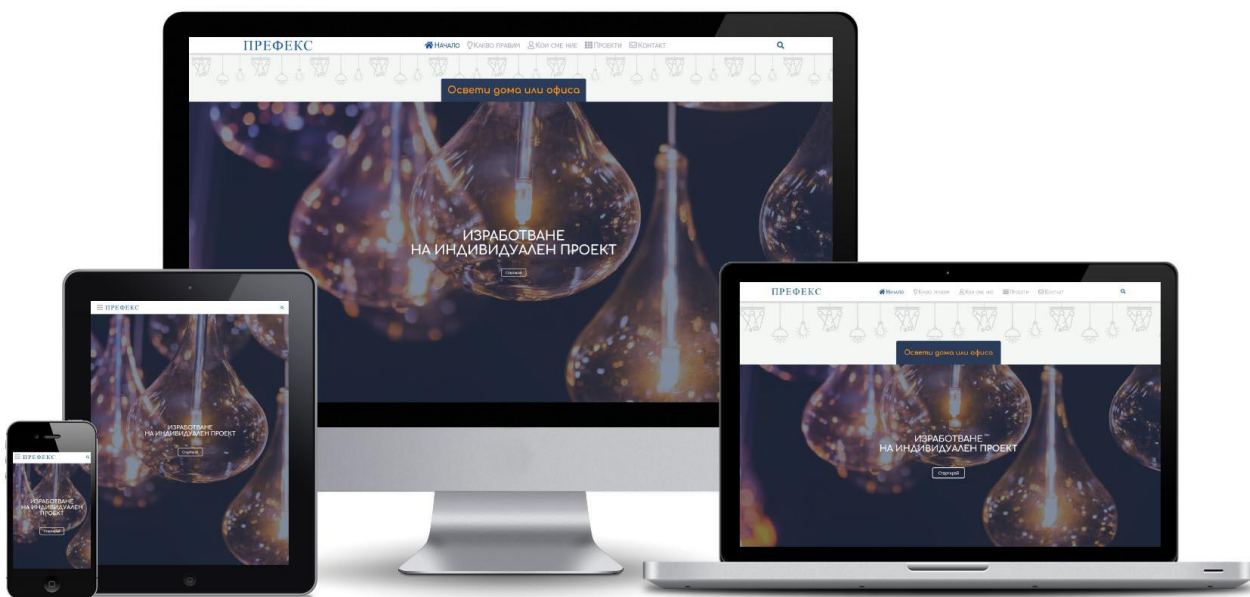


```
<div class="flex v-center h-center">
  <div class="p-1">Flex элемент 1</div>
  <div class="p-1">Flex элемент 2</div>
  <div class="p-1">Flex элемент 3</div>
</div>
```

Клас **.b-align**

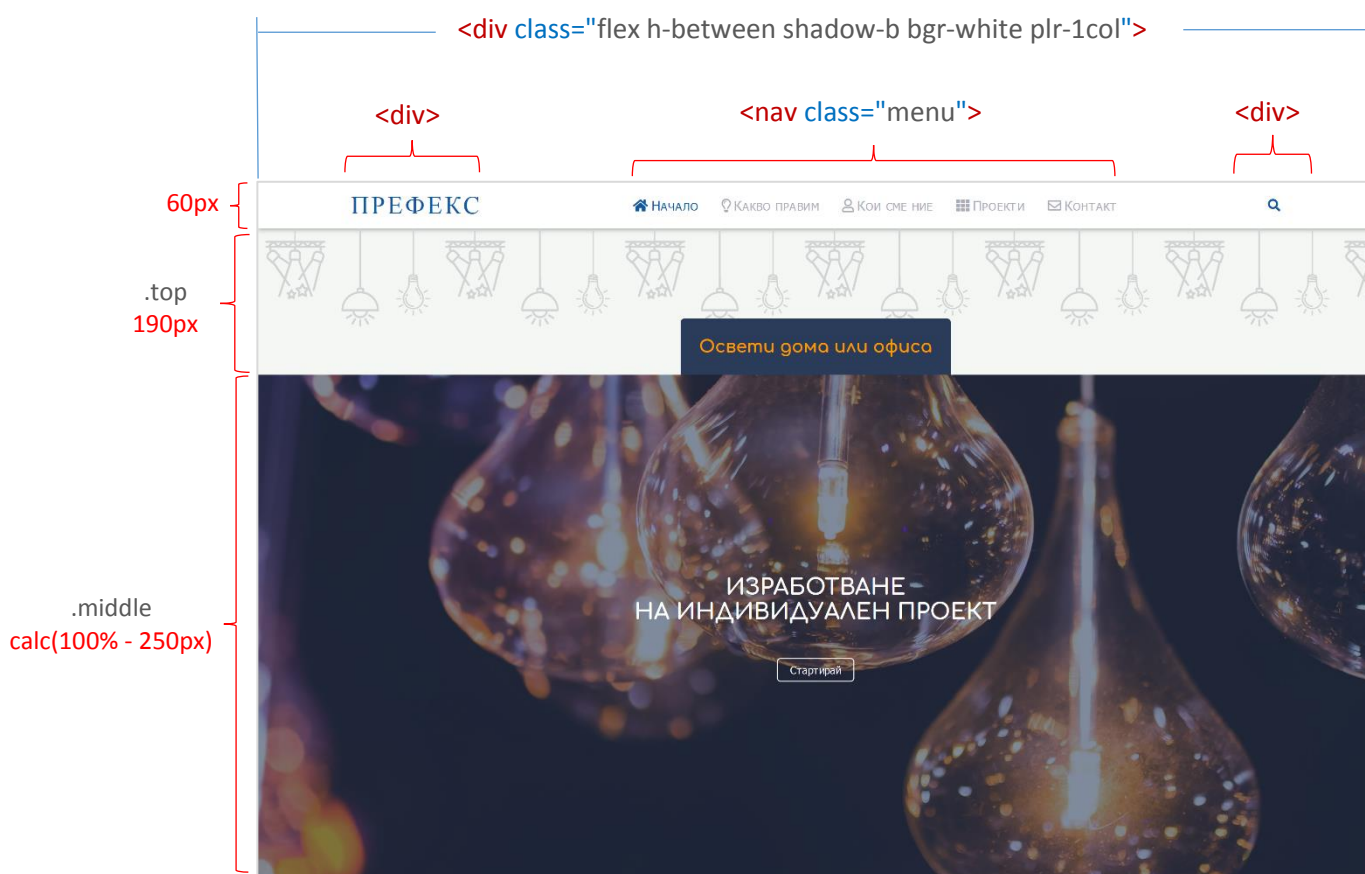


Разположение на съдържанието в сайта

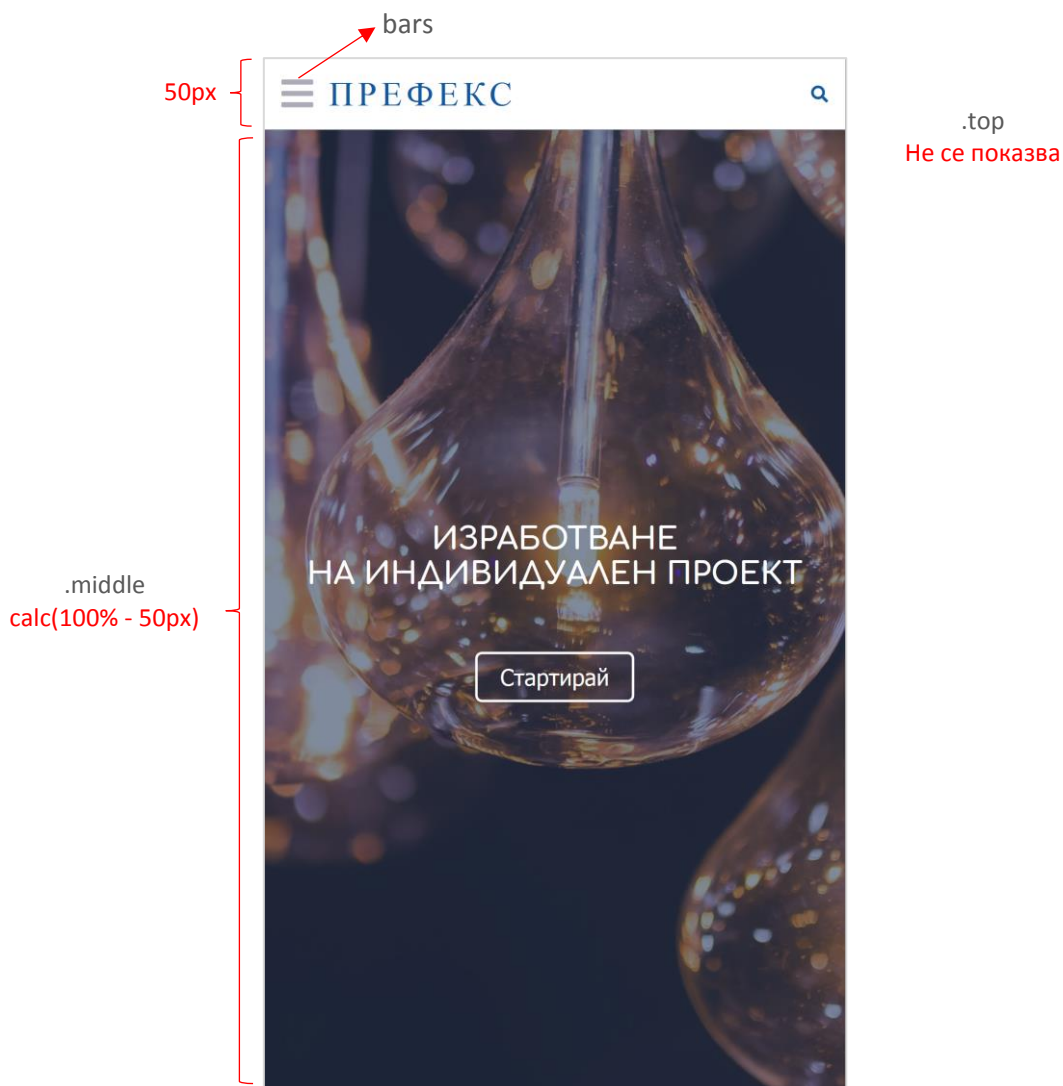


Показване на началната страница в iMac 2560x1440px, MacBook 1280x800px, iPad 768x1024px и iPhone 320x480px

Разположение в заглавната част на сайта

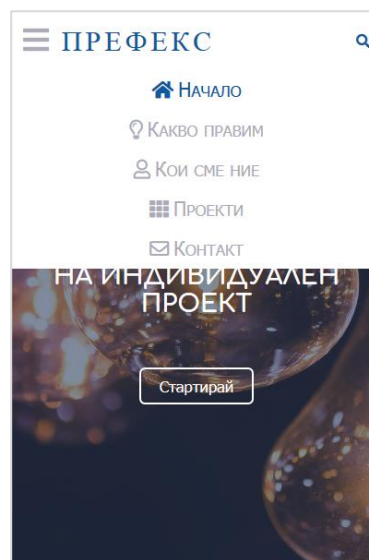


Представяне на компютър с резолюция 1440x900px



Представяне на телефон с резолюция 414x736px

Когато представянето е на таблет и телефон, клас **active** към неподредения списък скрива хоризонталното меню. Показва се иконка **bars**. При натискане на иконката клас **active** изчезва и се появява менюто подредено вертикално. Скриването и появяването на клас **active** към неподредения списък се контролира от файлът **jquerymenu.js** с библиотека **jQuery**.

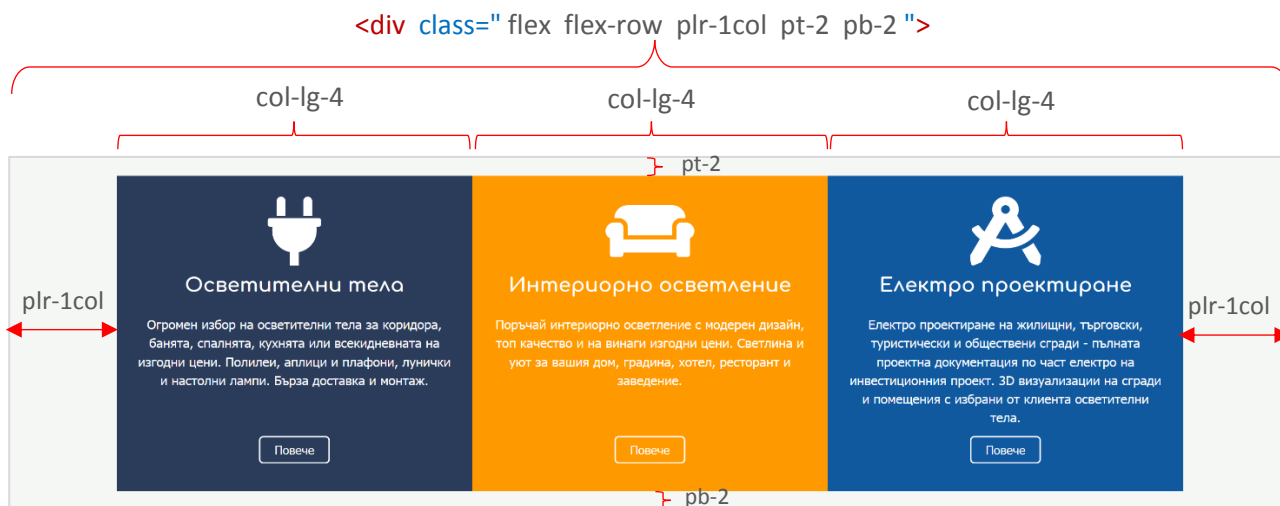


Библиотеката е заредена към сайта в секция **<head>** на **index.html**.

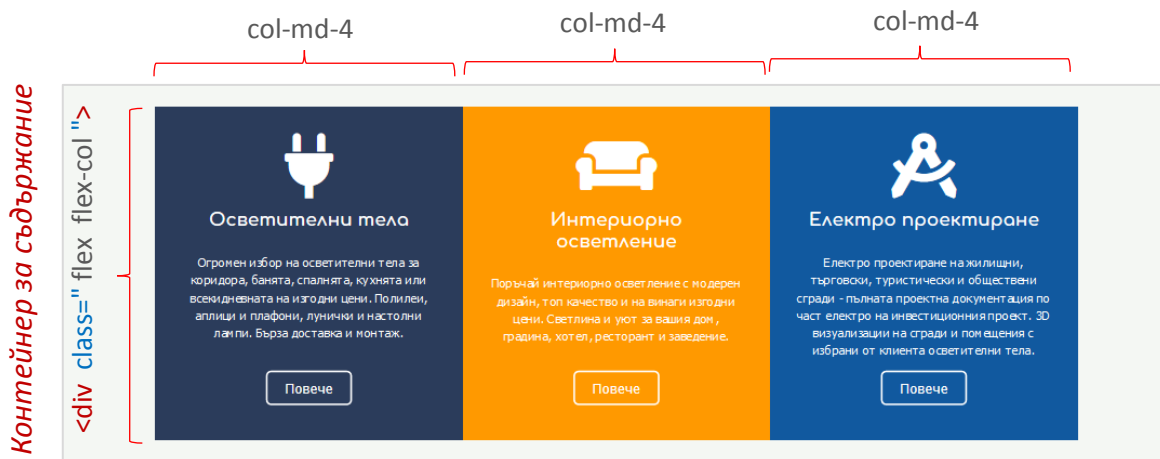
```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
```

Разположение на съдържание в три блока на ред
Три блока с различен фон и съдържание, което се разполага вертикално едно по друго.

Контейнер на трите блока



Представяне на компютър с резолюция 1440x900px



Представяне на компютър с резолюция 1024x768px

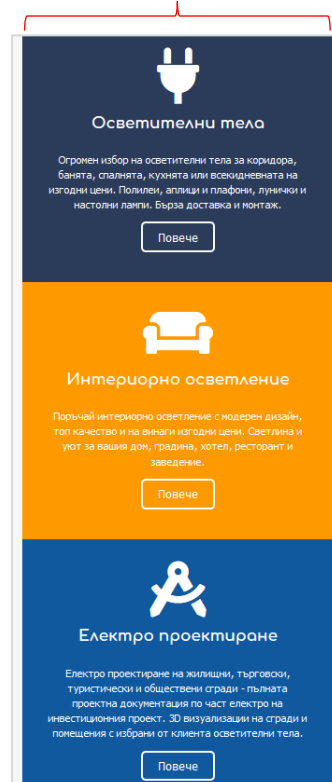
Бутоните отдолу на съдържанието се подреждат на една линия и стоят в края на контейнера с клас **flex-button** и благодарение на неговото свойство `margin-top: auto;`.

col-sm-12



Представяне на таблет с резолюция 800x1280px

col-xs-12



Представяне на телефон с резолюция 360x740px

Разположение на съдържание в два блока на ред
Ляв блок със заглавие и текст и десен блок с изображение с маска.

Контейнер на двата блока

```
<div class="flex flex-row bgr-darkblue plr-1col ">
```

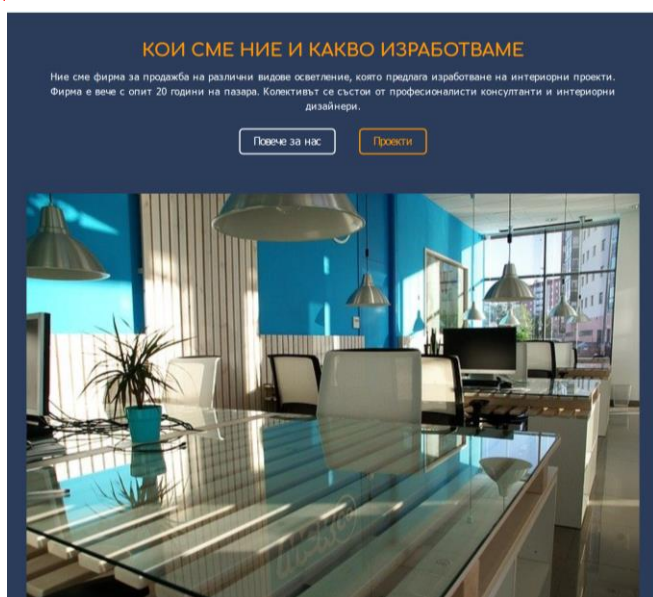
col-lg-6

col-lg-6



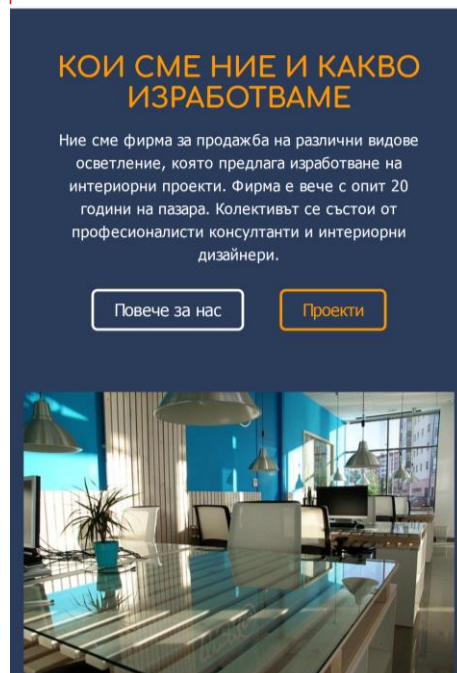
Представяне на компютър с резолюция 1440x900px

col-sm-12



Представяне на таблет с резолюция 800x1280px

col-xs-12



Представяне на телефон с резолюция 360x740px

Разположение на съдържание в четири блока на ред
Четирите блока се разполагат на един ред с разстояние между тях.

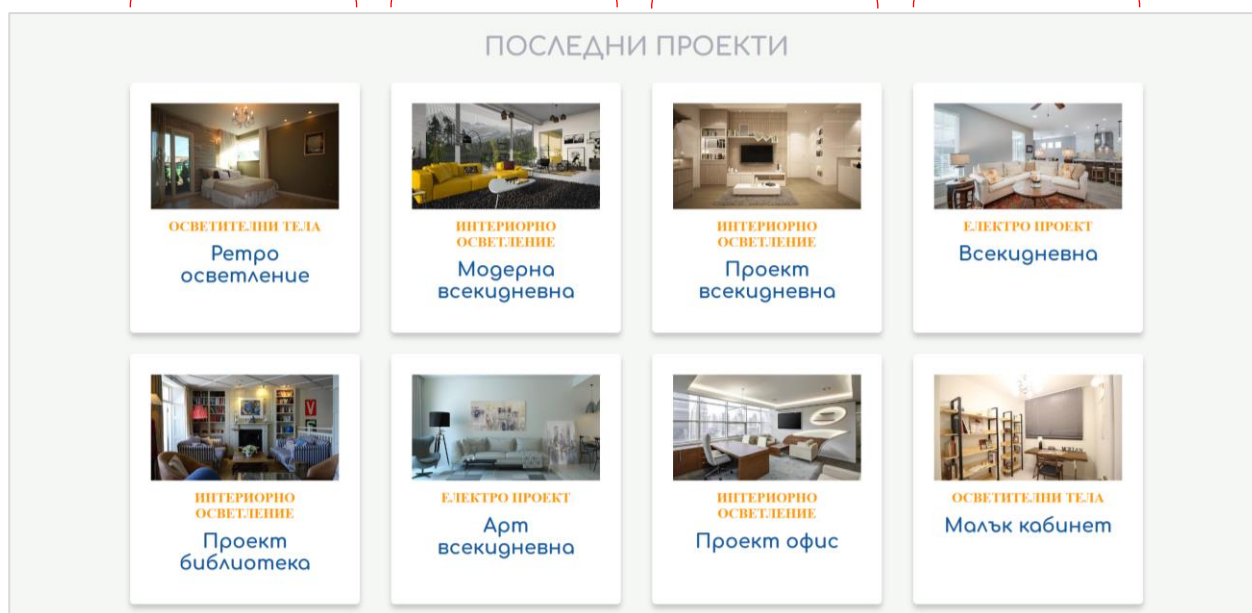
Контейнер на четири блока
<div class="flex h-around mt-2 plr-1col ">

four-lg

four-lg

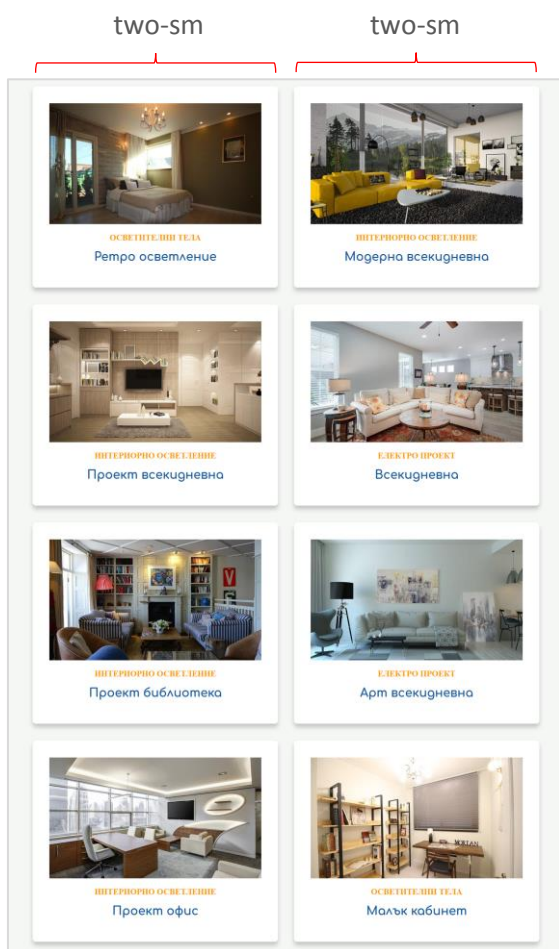
four-lg

four-lg

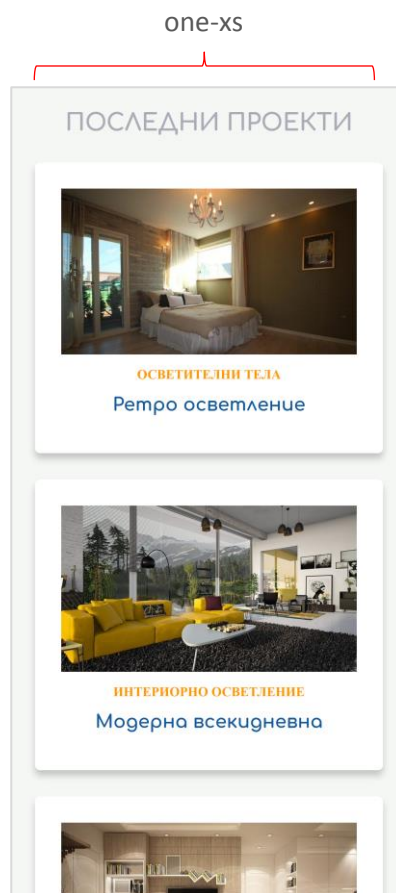


Представяне на компютър с резолюция 1440x900px

Клас **four-ig** създава ширина на блоковете 22%. Подредането им в контейнера с ширина 100% става с клас **h-around**. Блоковете се разполагат в контейнера с равно разстояние между тях.



Представяне на таблет с резолюция 800x1280px



Представяне на телефон с резолюция 360x740px

Клас **two-sm** създава ширина на блоковете 47%, клас **one-xs** 94%.

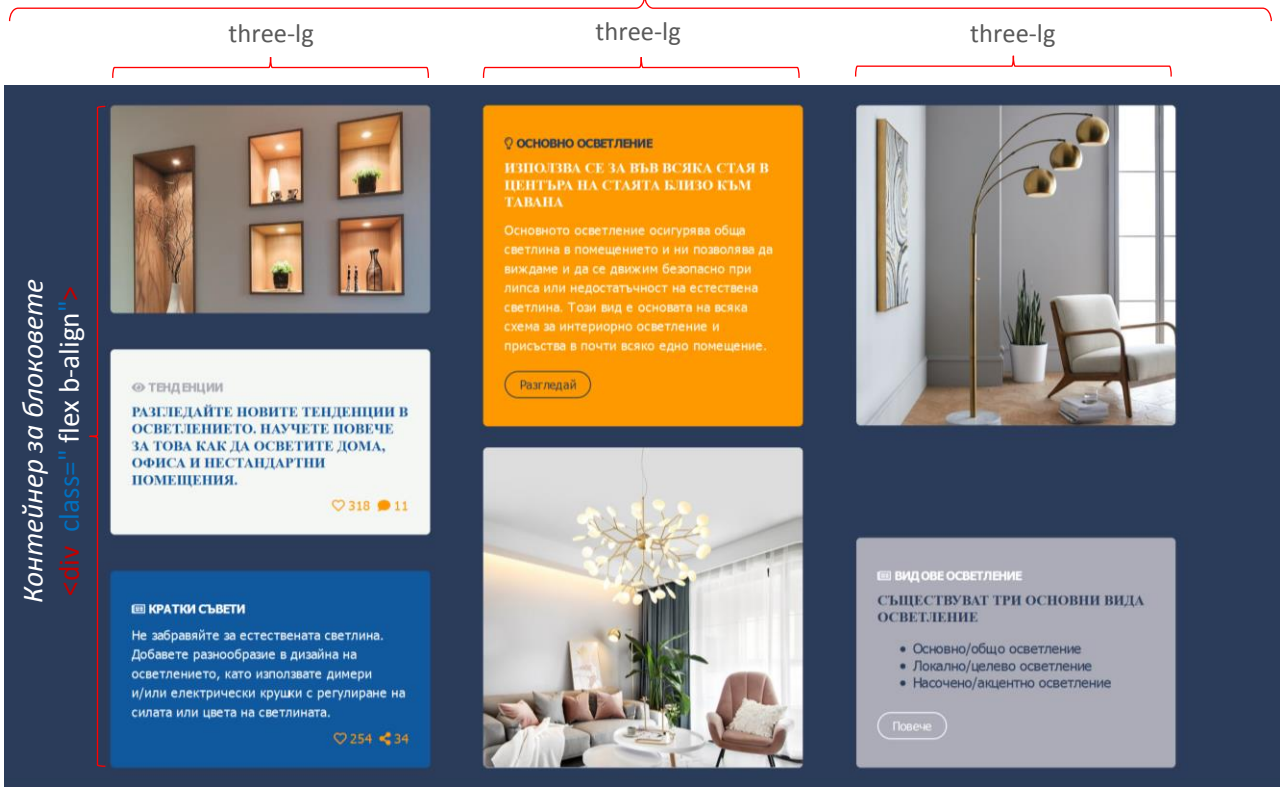
Разположение на съдържание в три колони с блокове в тях

Блоковете се разполагат в колоните по висината на контейнера и отстоят на равно отстояние един от друг.

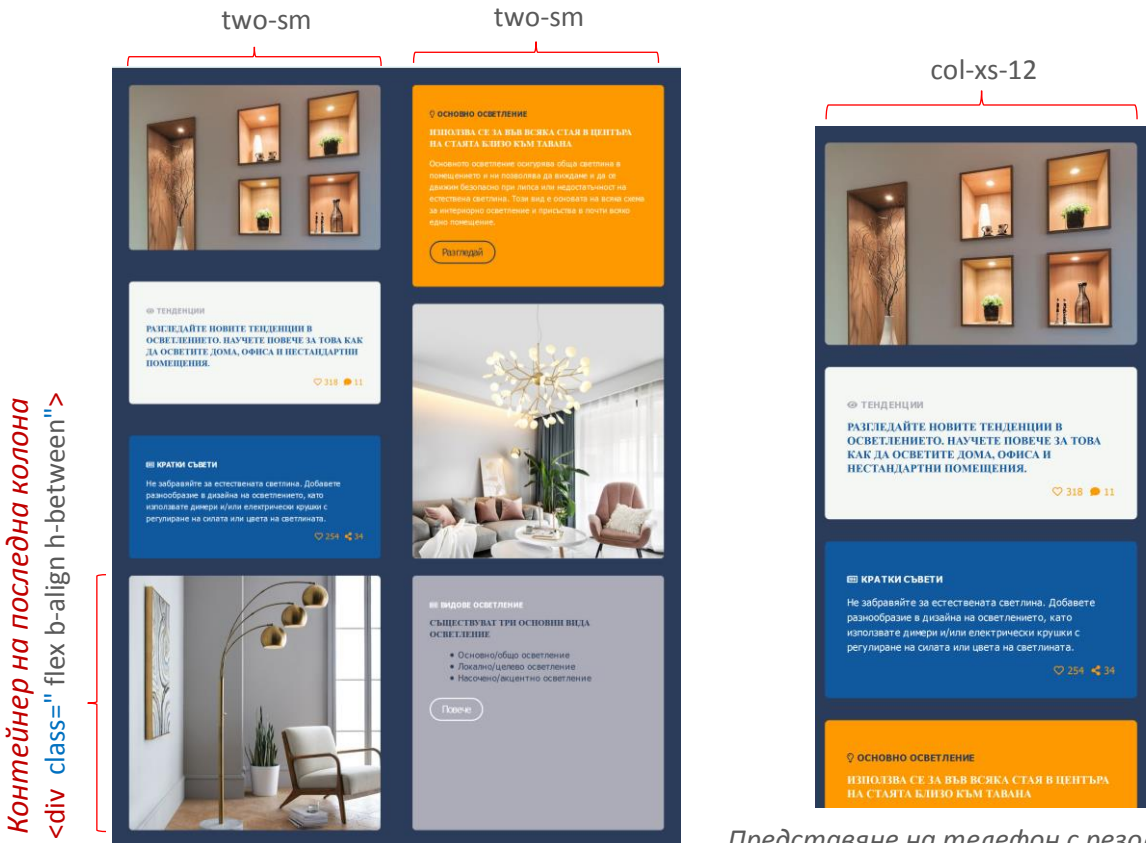
Колоните за компютър са с ширина 30%, за таблет 47% и за телефон 100%. Всеки блок е различна височина, фон и има различно съдържание. Подравняването на блоковете вертикално в контейнера на колоната става с клас **b-align**.

Контейнер с три колони

```
<div class="flex h-between plr-1col pt-2 bgr-darkblue ">
```



Представяне на компютър с резолюция 1440x900px



Представяне на таблет с резолюция 800x1280px

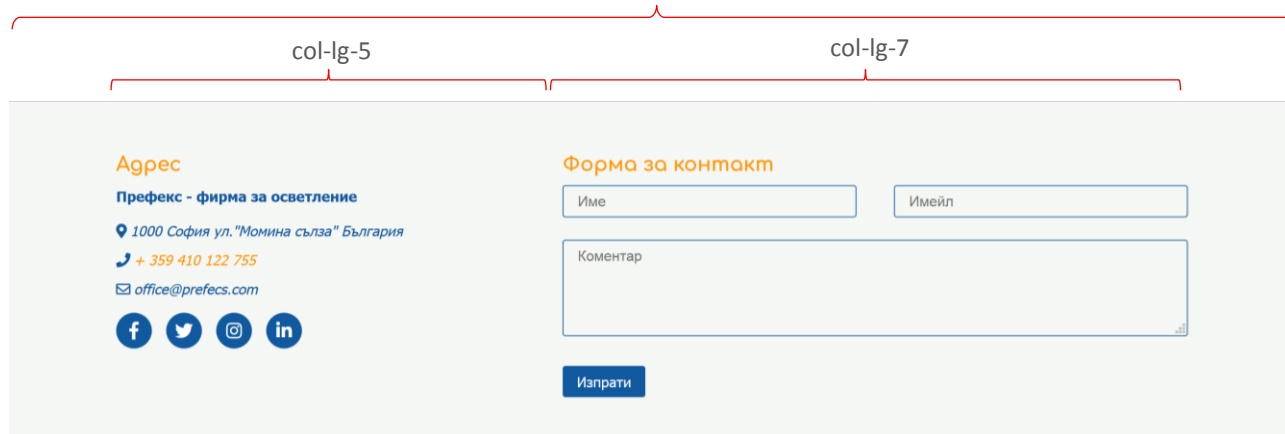
Представяне на телефон с резолюция 360x740px

Разположение на съдържание в две колони

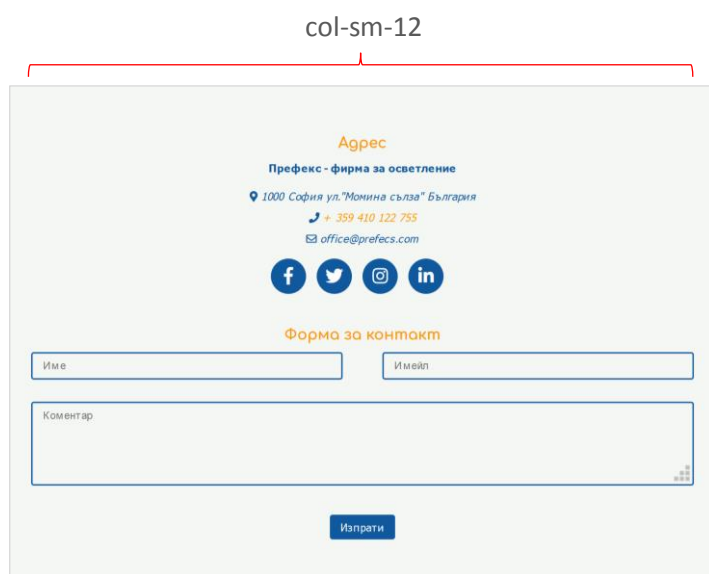
Колоните са с различна ширина при представяне на компютър и с ширина 100% за таблет и телефон.

Контейнер с две колони

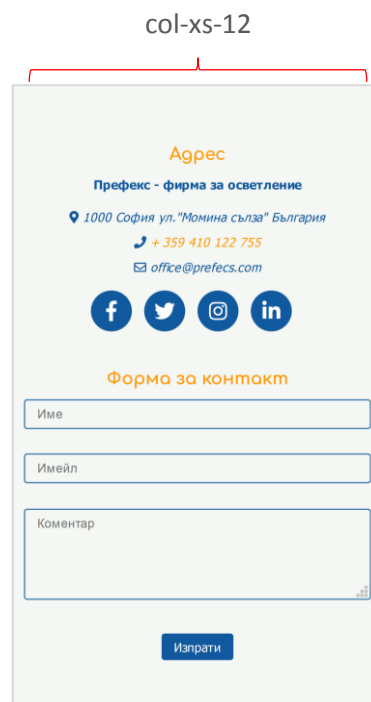
```
<div class="flex mt-2 plr-1col pt-2 pb-2 text-blue ">
```



Представяне на компютър с резолюция 1440x900px



Представяне на таблет с резолюция 800x1280px



Представяне на телефон с резолюция 360x740px

Съдържанието се центрира само за таблет и телефон с клас **text-center-s**. Той принадлежи към медийна заявка за малки устройства. Полета на формуляра са деца на flex контейнер, който обхваща полета на един ред.

Ефекти

Сянка към елементите

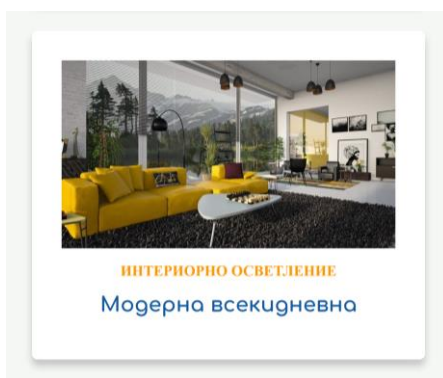
Три вида сянка – отдолу (клас `shadow-b`), отгоре (клас `shadow-t`) и от всички страни (клас `shadow`).

HTML

CSS

style.css

```
303 .shadow-b{
304     box-shadow: 0 2px 6px rgba(0,0,0,0.3);
305 }
306 .shadow-t{
307     box-shadow: 0 -2px 6px rgba(0,0,0,0.3);
308 }
309 .shadow{
310     box-shadow: 0 5px 6px rgba(0,0,0,0.2);
311 }
```



Сянка от всички страни с клас `shadow`

Закръгления на елементите

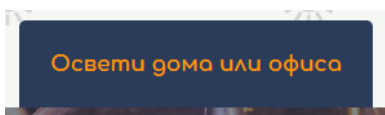
Буква **r**- в името на класа означава закръгление (rounding). Число **1** в името на класа дава закръгление `0.3rem`, число **2** съответно `1.1rem`.

HTML

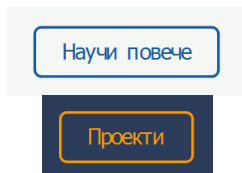
CSS

style.css

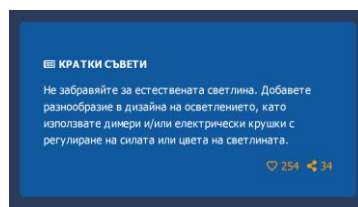
```
312 .r-1{border-radius: 0.3rem;}
313 .r-tlr-1{border-radius: 0.3rem 0.3rem 0 0;}
314 .r-blr-1{border-radius: 0 0 0.3rem 0.3rem;}
315 .r-2{border-radius: 1.1rem;}
```



r-blr-1



r-1



r-1



r-2

Ефект за увеличение

Клас **zoom** създава увеличение на елементите с 107%. Увеличението се извършва с метод `scale()` за 0.2s.

HTML

CSS

style.css

```
316 ▾ .zoom{
317     -webkit-transition: transform 0.2s ease;
318     transition: transform 0.2s ease;
319 }
320 ▾ .zoom:hover{
321     -ms-transform: scale(1.07, 1.07);
322     -webkit-transform: scale(1.07, 1.07);
323     transform: scale(1.07, 1.07);
324 }
```

Ефект overlay

Ефектът се прилага с клас **overlay** към `<div>`, който съдържа изображение и бутон. Бутонът е прозрачен. При преминаване над `<div>` елемента, изображението става 30% прозрачно, а бутонът става плътен и може да бъде натиснат.

HTML

CSS

```
147 ▾ <div class="overlay mb-2">
148     
149     <button class="btn-white r-2">Виж</button>
150 </div>
```

HTML

CSS

```
325 ▾ .overlay{
326     position: relative;
327 }
328 ▾ .overlay img{
329     opacity: 1;
330     -webkit-transition: all 0.5s ease;
331     transition: all 0.5s ease;
332     backface-visibility: hidden;
333 }
334 ▾ .overlay:hover img{
335     opacity: 0.3;
336 }
337 ▾ .overlay button{
338     opacity: 0;
339     position: absolute;
340     top: 50%;
341     left: 50%;
342     -ms-transform: translate(-50%, -50%);
343     -webkit-transform: translate(-50%, -50%);
344     transform: translate(-50%, -50%);
345 }
346 ▾ .overlay:hover button{
347     opacity: 1;
348 }
349 }
```

Маска на изображение

Маскираните изображения се виждат само за компютър. Клас **clip-par** състав маска успоредник и се изпълнява в медийна заявка за устройства над 992px.

HTML

CSS

```
105 @media(min-width:992px){  
106   .clip-par{  
107     clip-path: polygon(25% 0%, 100% 0%, 75% 100%, 0% 100%);  
108   }  
109 }
```



`clip-path: polygon(25% 0%, 100% 0%, 75% 100%, 0% 100%);`

Съдържание

Структура на проекта	2
Цветова схема на проекта	2
Външни и вътрешни отстояния	3
Решетка описана с flex	4
Класове от решетката за описание на блокове луфт между тях	5
Класове на flex технология за RWD	6
Разположение на съдържанието в сайта	10
Разположение в заглавната част на сайта	10
Разположение на съдържание в три блока на ред	12
Разположение на съдържание в два блока на ред	13
Разположение на съдържание в четири блока на ред	14
Разположение на съдържание в три колони с блокове в тях.....	15
Разположение на съдържание в две колони	17
Ефекти	18
Сянка към елементите.....	18
Закръгления на елементите.....	18
Ефект за увеличение	19
Ефект overlay.....	19
Маска на изображение.....	20