

UPD – в прошлой методичке забыла *position: sticky; top: 0*

```
header {  
  display: flex;  
  align-items: center;  
  justify-content: space-between;  
  padding: 20px 10%;  
  background: #1b242f;  
  position: sticky;  
  top: 0;  
}
```

Добавление эффектов при наведении, анимаций и адаптивности в проект

Откроем созданный на уроке 1 проект.

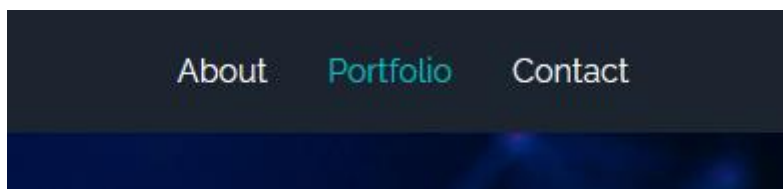
Начнем с изменения цвета ссылок в навигации при наведении

В файле `style.css` найдем раздел, где описывали стили для ссылок в шапке. В принципе, порядок правил в файле не важен, но лучше не делать вперемешку, для понятности. Ищем, где стили для `header a` и модифицируем следующим образом:

```
header a {  
  color: white;  
  text-decoration: none;  
  transition: .3s;  
}  
  
header a:hover {  
  color: #04c2c9;  
}
```

Здесь для ссылки добавилось свойство *transition* – это длительность перехода=0.3сек

Для ссылки при наведении меняется цвет:



Далее добавим псевдоэлемент к ссылке в навигации. Ищем, где у нас стили для `nav a`, и добавляем:

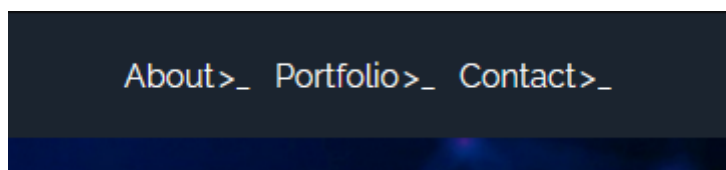
```

✓ nav a {
  position: relative;
}
✓ nav a::after {
  content: ">_";
  position: absolute;
  margin-left: 2px;
  left: 100%;
}

```

Здесь ссылке добавили относительное позиционирование, чтобы мы смогли абсолютно позиционировать псевдоэлемент.

::after будет содержать символы ">_" и быть абсолютно позиционирован справа от ссылки, и отступ, чтобы сильно не липло.



Теперь мы хотим, чтобы этот элемент был только при наведении на ссылку, при этом анимирован. Добавим следующее:

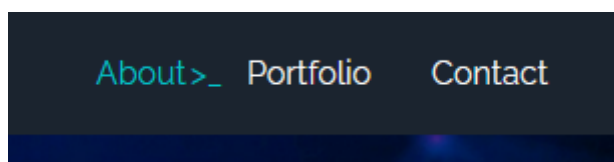
```

✓ nav a:not(:hover)::after {
  opacity: 0;
  transform: translateY(-50%);
}
✓ nav a:hover::after {
  transition: .3s;
}

```

У ссылок, которые НЕ в состоянии наведения, псевдоэлемент будет прозрачным и сдвинут вверх на половину высоты.

Плавность перехода была добавлена только для момента наведения, чтобы сильно не мельтешило. Но можно поэкспериментировать.



Еще вариант псевдоэлемента: блок за блоком. Находим стили для .about-img и добавляем:

```

.about-img, .about-img img {
  position: relative;
}
.about-img::before {
  content: '';
  display: block;
  background-color: #04c2c9;
  position: absolute;
  top: 10px;
  left: 10px;
  width: 100%;
  height: 100%;
}

```

Опять же относительное позиционирование для блока и картинки в нем. Если картинке не сделать relative, то псевдоэлемент ее закроет.

За блоком с картинкой будет псевдоблок шириной и высотой по 100%, сдвинутый на 10px по вертикали и горизонтали:

Who am I?

Full-stack web developer. Skilled in development of LMS, training platforms, landings, API for apps. Winner of WorldSkills Kazakhstan 2019, WorldSkills Eurasia 2020, WorldSkills Asia 2021. Bronze medalist and Best of the nation in EuroSkills 2021. Deputy Chief Expert for Web Development Competence.

Also student of Computer Science faculty.

What is my next purpose? I think that this is learning more in IT, improving skills, and try to create something great :)



Теперь добавим эффекты на галерею. У нас будет фильтр на картинке при наведении, также трансформация и появление текста. Сначала модифицируем html, добавив теги с текстом в ячейки после картинки:

```

1  class="container gallery">
2  Just gallery demo</h2>
3  class="grid">
4  <div class="cell">
5      Hello from cell</div>
7  </div>
8  <div class="cell">
9      Look at img</div>
11 </div>
12 <div class="cell">
13     Another text</div>
15 </div>
16 <div class="cell">
17     А если по-русски?</div>
19 </div>
20 <div class="cell">
21     И зачем это</div>
23 </div>
24 <div class="cell">
25     А вот и все</div>
27 </div>

```

Теперь найдем стили для .gallery. Добавим следующее. Это позволит картинке не вылезать за пределы бокса и абсолютно позиционировать текст поверх картинки:

```

.gallery .grid .cell {
  overflow: hidden;
  position: relative;
}

```

Найдем существующие стили для картинок и добавим *transition*:

```

.gallery .grid .cell img {
  width: 100%;
  height: 100%;
  object-fit: cover;
  transition: .4s;
}

```

Добавим эффект при наведении на блок. Картинка будет увеличена и затемнена:

```

.gallery .grid .cell:hover img {
  transform: scale(1.1);
  filter: brightness(.25);
}

```

Стилизуем .cell-text и отключим отображение без наведения:

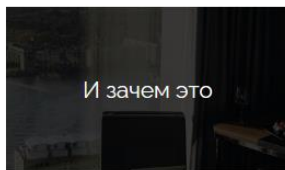
```

.gallery .grid .cell .cell-text {
  position: absolute;
  color: white;
  font-size: 1.3rem;
  top: 0;
  left: 0;
  width: 100%;
  height: 100%;
  display: flex;
  align-items: center;
  justify-content: center;
}
.gallery .grid .cell:not(:hover) .cell-text {
  display: none;
}

```

Текст будет абсолютно растянут на всю площадь блока и центрован.

Just gallery demo



Добавим эффект для фокусированных элементов ввода. Найдем стили для `.contact input`, `.contact select`, `.contact textarea` и добавим:

```

.contact input:focus, .contact select:focus, .contact textarea:focus {
  border-bottom: 1px solid #04c2c9;
  padding-bottom: 9px;
}

```

Добавили нижнюю границу, а чтобы высота не прыгала, уменьшили нижний отступ на 1 пиксель (с 10 до 9):

What is your name?

Mr. Somebody

Where should I answer to?

thebest@domain.com

Choose your topic

Work

А для футера придумать самостоятельно.

Для практики добавим пару анимаций. В конце файла стилей добавим:

```
@keyframes appear-from-left {
  0% {
    opacity: 0;
    transform: translateX(-100%);
  }
}

@keyframes appear-from-bottom {
  0% {
    opacity: 0;
    transform: translateY(100%);
  }
}

@keyframes appear-from-opacity {
  0% {
    opacity: 0;
  }
}
```

Это описывает 3 анимации: появления слева, снизу и через прозрачность.

Найдем стили для .card. Добавим свойство анимации:

```
.card {
  padding: 50px;
  background: linear-gradient(to top, black, transparent);
  text-align: center;
  color: white;
  animation: appear-from-opacity 2s;
}
```

Найдем стили для секции `.about` и добавим:

```
.about-text {  
  animation: appear-from-left .75s .75s backwards;  
}  
.about-img {  
  animation: appear-from-bottom .75s 1s backwards;  
}
```

Анимация для этих блоков будет запускаться с отставанием в `.75` и `1` секунду соответственно. С помощью *backwards* они изначально будут иметь состояние как на 1 кадре анимации.

Добавление адаптивности

Добавляем и `.menu-toggler #menu-check` в header:

```
<header>
  <a href="#" class="logo">A. Dina</a>
  <label for="menu-check" class="menu-toggler">
    <span></span>
    <span></span>
    <span></span>
  </label>
  <input type="checkbox" id="menu-check">
  <nav>
    <a href="#">About</a>
    <a href="#">Portfolio</a>
    <a href="#">Contact</a>
  </nav>
</header>
```

Это иконка меню и флажок переключения. В конце `style.css` стилизуем их. Иконку скроем на десктопной версии.

```
#menu-check {
  display: none;
}
.menu-toggler {
  display: grid;
  grid-template-rows: repeat(3, 2px);
  grid-gap: 7px;
  width: 25px;
  cursor: pointer;
}
.menu-toggler span {
  background: white;
}

@media(min-width: 768px) {
  /*
  // Стили, больше планшетных
  */
  .menu-toggler {
    display: none;
  }
}
```

Добавим переключение по флажку на версии меньше планшетной:


```

}
@media(max-width: 767px) {
  header {
    position: relative;
  }
  nav {
    background: inherit;
    position: absolute;
    top: 100%;
    left: 0;
    width: 100%;
    flex-direction: column;
    align-items: flex-start;
    padding: 10px 10%;
  }
  nav a {
    padding: 10px 0;
  }
  #menu-check:not(:checked) + nav {
    display: none;
  }
}

```

Вот что получилось на чистом CSS:



Добавим стили для блока обо мне в том же медиазапросе (max-width: 767px):

```

    .about {
      display: block;
    }
    .about-text, .about-img {
      width: 100%;
    }
    .about-text {
      margin-bottom: 1.5rem;
    }
  }
}

```

Блоки встают в одну колонку, к тексту добавили отступ.

Who am I?

Full-stack web developer. Skilled in development of LMS, training platforms, landings, API for apps. Winner of WorldSkills Kazakhstan 2019, WorldSkills Eurasia 2020, WorldSkills Asia 2021. Bronze medalist and Best of the nation in EuroSkills 2021. Deputy Chief Expert for Web Development Competence.

Also student of Computer Science faculty.

What is my next purpose? I think that this is learning more in IT, improving skills, and try to create something great :)



Меняем стили сетки там же:

```

    .gallery .grid {
      grid-template-columns: repeat(3, 1fr);
    }
    .gallery .grid .cell:first-child {
      grid-column: 1 / 3;
      grid-row: 1;
    }
    .gallery .grid .cell:nth-child(3) {
      grid-row: 2 / 4;
    }
    .gallery .grid .cell:last-child {
      grid-column: 2 / 4;
    }
  }
}

```

Just gallery demo



Еще один медиазапрос для мобилок:

```
@media(max-width: 576px) {  
  .gallery .grid {  
    grid-template-columns: 1fr;  
  }  
  .gallery .grid .cell {  
    grid-row: unset !important;  
    grid-column: unset !important;  
  }  
}
```

Все блоки сетки в одну колонку.

Just gallery demo

