

ВВЕДЕНИЕ В МЕДИА-ЗАПРОСЫ



СЕМЕН БОЙКО / РУКОВОДИТЕЛЬ ПРОЕКТОВ В ВЕБ-СТУДИИ «КЛИПСАЙТ»



СЕМЕН БОЙКО

руководитель проектов в Веб-студии «Клипсайт»



simonderus@gmail.com



<https://www.facebook.com/simonboycko>

ПЛАН ЗАНЯТИЯ

1. Отдельная мобильная версия сайта
2. Медиазапросы
3. Режим эмуляции устройств в веб-инспекторе
4. flex-wrap
5. align-content

ОТДЕЛЬНАЯ МОБИЛЬНАЯ ВЕРСИЯ САЙТА

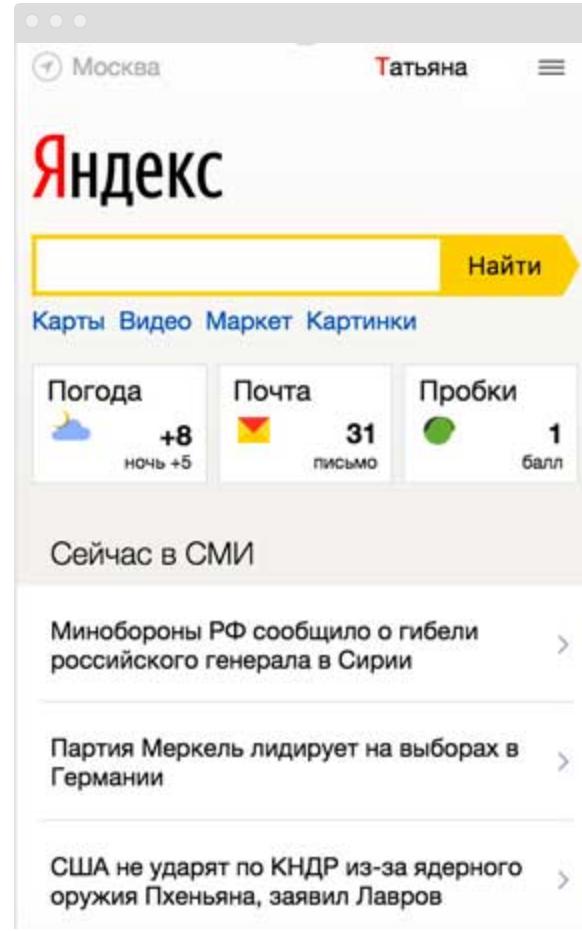
ПРИМЕР ЯНДЕКСА

Откроем главную страницу yandex.ru с ноутбука.

The screenshot shows the Yandex.ru homepage with a light gray header bar. On the left is a location indicator 'Москва' with a map pin icon. On the right are links for 'Настройка', 'Почта 31', 'Диск', and a user profile for 'Татьяна'. Below the header is a news section titled 'Сейчас в СМИ в Москве' with a timestamp '24 сентября, воскресенье 22:24'. It lists several news items, including the death of a Russian general in Syria and the German election results. To the right of the news is a small grid of five car icons. Further down, there's a promotional banner for 'Automoibili до 350 000 рублей' on 'Auto.ru'. The main search bar is centered at the top of the page, with a yellow button labeled 'Найти' (Search). Below the search bar, a message says 'Найдётся всё. Например, что такое ШАД'. To the right of the search bar is a link to 'Попробуйте быстрый браузер' (Try the fast browser). At the bottom of the page, there are three horizontal cards: one for 'PJ MASKS' with cartoon characters, one for a 'Скидка до 15%' offer with a car, and one for 'WILDBERRIES' with a purple background. At the very bottom, there are quick links for 'Погода +9 °C', 'Пробки 1 балл', 'Карта Москвы', and 'Метро Такси Расписания'.

МОБИЛЬНЫЙ ЯНДЕКС

И откроем ту же страницу с телефона.



ДВЕ РАЗНЫЕ ВЕРСИИ

Десктоп и мобильная версии, как правило, сильно отличаются. Еще один пример – [Lamoda](#)

Ваш регион доставки: г Санкт-Петербург

Женщинам Мужчинам Детям

lamoda

Войти Корзина

Осень 2017 Одежда Обувь Аксессуары Бренды Премиум Спорт Красота Распродажа %

MIDSEASON SALE
до 40%
по промокоду на осеннею коллекцию

Купить со скидкой

Дополнительно до 30%
на верхнюю одежду

50%
Любимые модели

по промокоду: MSS17

Десктоп версия

≡ lamoda

Женщинам Мужчинам Детям Бренды

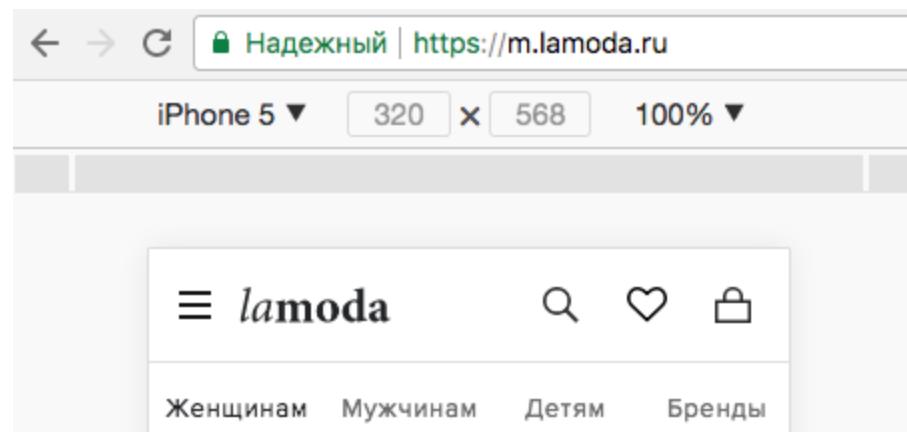
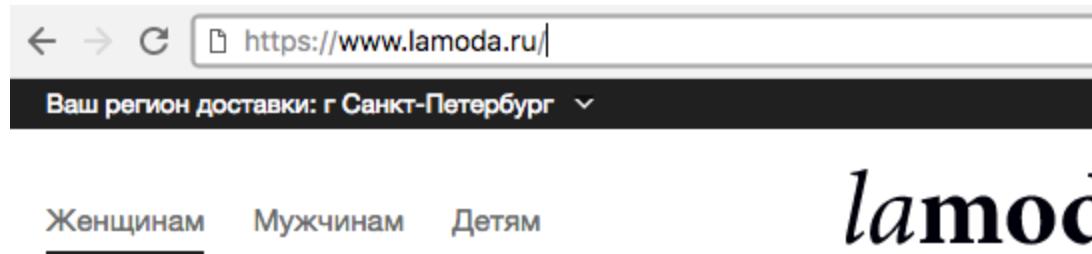
MIDSEASON SALE
40%
дополнительные скидки

Согреем скидками
Промокод: MSS17

Новинки
Мобильная
версия
Одежда
Обувь

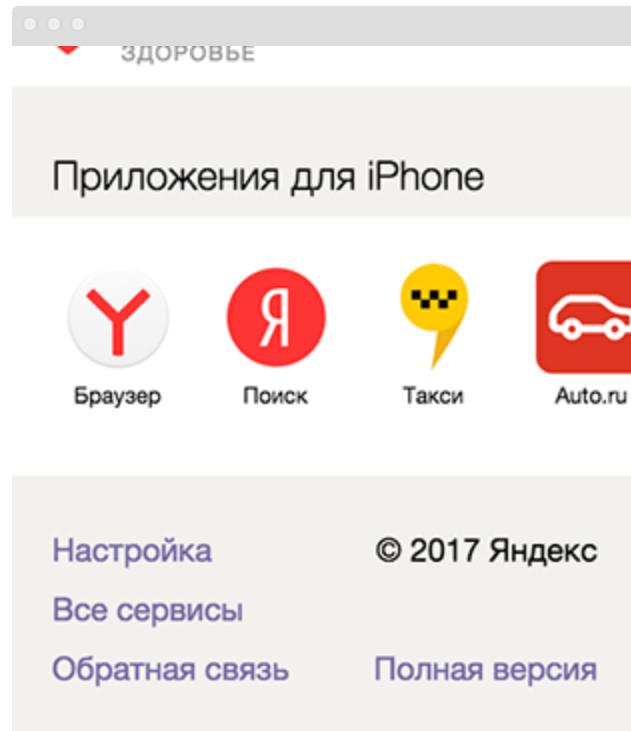
ПЕРЕАДРЕСАЦИЯ ДЛЯ МОБИЛЬНЫХ УСТРОЙСТВ

Обычно версии доступны по разным адресам. К адресу мобильной версии принято добавлять `m.` в начале адреса:



ПЕРЕКЛЮЧЕНИЕ МЕЖДУ ВЕРСИЯМИ

Принято предоставлять возможность перейти на «основную» версию сайта – например, в подвале:



ПЛЮСЫ МОБИЛЬНОЙ ВЕРСИИ

Как правило, мобильная версия имеет упрощенный внешний вид и урезанный функционал. Такой сайт быстрее загружается и не отвлекает пользователя от главной задачи и основного, самого важного контента.

Пользоваться навигацией с сенсорного устройства на мобильной версии намного удобнее.

МИНУСЫ МОБИЛЬНОЙ ВЕРСИИ

К минусам подхода можно отнести необходимость разработки, поддержки и синхронизации двух независимых версий сайта.

Также минусом является частичное повторение контента по двум разным адресам – ведь на обеих версиях основной контент будет одинаков.

А еще сложно разработать дизайн, одинаково хорошо отвечающий потребностям двух разных типов мобильных устройств – сенсорных устройств и устройств с клавиатурой (например, устройства-трансформеры).

ТЯЖЕЛЫМ ПРОЕКТАМ – МОБИЛЬНАЯ ВЕРСИЯ

В целом создание отдельной мобильной версии сайта может быть оправдано для больших «тяжелых» проектов, когда только разработав «упрощенную» версию сайта можно добиться быстрой загрузки на мобильных устройствах и получить удобный интерфейс.

МЕДИАЗАПРОСЫ

КАРТОЧКА СОТРУДНИКА

В ходе работы над одним из проектов мы сверстали карточку сотрудника.



ИВАН ИВАНОВ

HTML-верстальщик

Возраст: 26

Образование: Высшее

Стаж: 5 лет

ВЕРСТКА

Начальный

Средний

Опытный

Эксперт

ДИЗАЙН

Начальный

Средний

Опытный

Эксперт

JAVASCRIPT

Начальный

Средний

Опытный

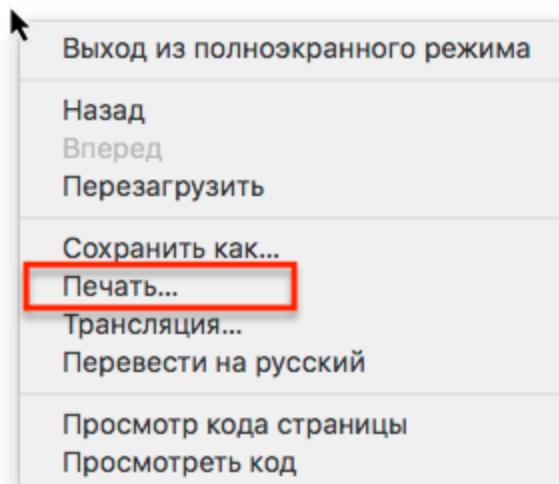
Эксперт

[Live Demo](#)

ПЕЧАТЬ ИЗ БРАУЗЕРА

Через некоторое время к нам подходит HR-менеджер и просит сделать вывод карточки на печать.

В браузере мы выбираем в выпадающем по правому клику мыши меню пункт «Печать» или нажимаем `Ctrl + P` / `⌘ + P`.



ПЕЧАТАЕМ КАРТОЧКУ

Видим, что верстка выглядит не так, как надо:

The screenshot shows a print preview dialog on the left and a resume card template on the right. The print dialog includes fields for printer selection (Printer), page range (All), and layout (Portrait). The resume card displays a photo of a man, Ivan Ivanov, an HTML layout specialist, aged 26, with 5 years of experience. It lists skills in Design, CSS, and JavaScript, each with four levels: Beginner, Intermediate, Advanced, and Expert.

Печать

Всего: 1 страница

Отмена Сохранить

Принтер Сохранить как PDF
Изменить

Страницы Все
Например: 1–5, 8, 11–13

Раскладка Книжная

+ Дополнительные настройки

Печатать с помощью системного
диалогового окна (⌘P)

Открыть PDF для просмотра

**ИВАН
ИВАНОВ**
HTML-
верстальщик

Возраст: 26
Образование: Высшее
Стаж: 5 лет

ВЕРСТКА

Начальный	Средний	Опытный	Эксперт
-----------	---------	---------	---------

ДИЗАЙН

Начальный	Средний	Опытный	Эксперт
-----------	---------	---------	---------

JAVASCRIPT

Начальный	Средний	Опытный	Эксперт
-----------	---------	---------	---------

ПРОБЛЕМЫ ПРИ ВЫВОДЕ НА ПЕЧАТЬ

Получилось, что при печати страницы наши прогресс-бары полностью не видны. Это произошло потому, что при печати страницы свойство `background` игнорируется. Поэтому нам нужно сделать альтернативный способ вывода информации.

ЗАМЕНИМ ГРАФИК НА ТЕКСТ

Например, при печати можно заменить график на текстовое описание опыта. Для этого нужно добавить в HTML само описание. Но! Оно также отобразится и при обычном просмотре браузера.

ОТДЕЛЬНЫЙ CSS

Очевидно, что нам нужно написать отдельную версию CSS для обычного просмотра и для печати. Но как же можно это делать? Для решения этой задачи в CSS существуют *медиазапросы*.



Медиазапросы – это возможность браузера выбрать устройства, которые отвечают заданным параметрам, и применить к ним отдельный CSS.

Первым параметром, по которому можно отобрать устройства, – тип устройства или медиатип.



МЕДИАТИПЫ

Существуют следующие медиатипы:

screen – Если указать `screen`, то браузер применит CSS только для устройств в обычном режиме просмотра.

print – При значении `print` браузер применит CSS только для устройств в режиме печати страницы.

speech – Если указать `speech`, то браузер применит CSS только для устройств в режиме озвучивания текста.

all – При этом значении браузер применит CSS для устройств в режиме обычного просмотра, печати или озвучивания текста. Также это значение является значением по умолчанию.

ВНЕДРЕНИЕ МЕДИАЗАПРОСОВ

Для того, чтобы начать использовать медиазапросы, нужно добавить атрибут `media` к тегу `<link>` и указать значение нужного типа устройства. Например, укажем `print` :

```
<link  
  href="for-print.css"  
  rel="stylesheet"  
  media="print">
```

В итоге браузер загрузит файл `for-print.css` только при печати страницы.

ДОБАВИМ ТЕКСТОВОЕ ОПИСАНИЕ

Теперь вернемся к нашему примеру. Добавим текстовое описание для навыков:

```
1 <div class="about__skill">
2   <span class="about__skill-name">Верстка</span>
3   <div class="about__skill-progress">
4     <span class="about__skill-value">Эксперт</span>
5     <div class="about__skill-bar" style="width: 90%"></div>
6   </div>
7   <div class="about__skill-captions">
8     <span class="about__skill-caption">Начальный</span>
9     <span class="about__skill-caption">Средний</span>
10    <span class="about__skill-caption">Опытный</span>
11    <span class="about__skill-caption">Эксперт</span>
12  </div>
13 </div>
```

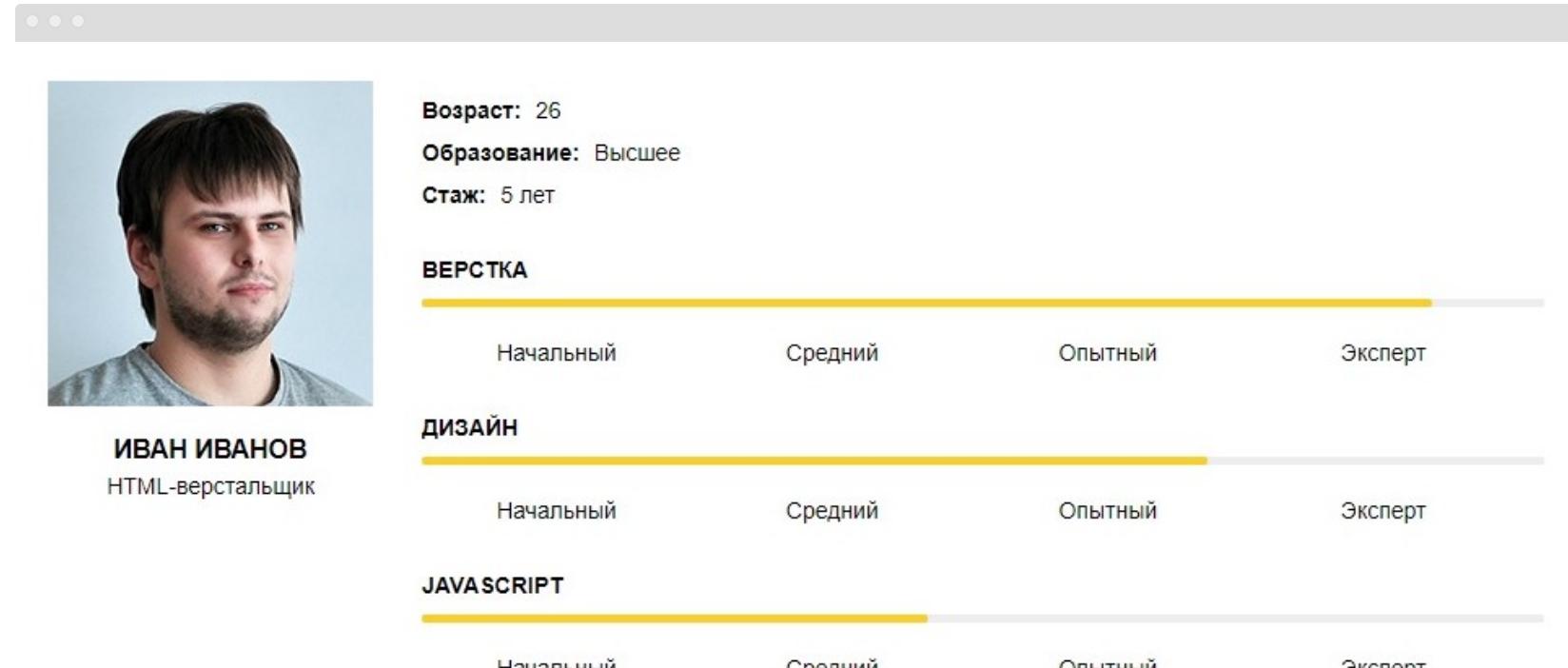
СКРЫВАЕМ ОПИСАНИЕ

Текстовое описание требуется только при печати страницы, поэтому для отображения в обычном режиме мы их скроем. Добавим стили в файл style.css

```
1 | .about__skill-value {  
2 |   display: none;  
3 | }
```

РЕЗУЛЬТАТ ИЗМЕНЕНИЙ

В итоге страница отобразится так:



[Live Demo](#)

СКРЫВАЕМ ПРОГРЕСС-БАР

А теперь самое интересное. В печатной версии страницы нам нужно скрыть графики и подписи к ним. Для этого создадим отдельный CSS-файл `print.css` и добавим в него следующий CSS:

```
1 | .about__skill-bar, .about__skill-captions {  
2 |   display: none;  
3 | }
```

ПОКАЗЫВАЕМ ТЕКСТОВОЕ ОПИСАНИЕ

Также нам нужно отобразить текстовое описание. В том же файле `print.css` добавляем стили:

```
1 .about__skill-bar, .about__skill-captions {  
2     display: none;  
3 }  
4  
5 .about__skill-value {  
6     display: inline-block;  
7 }
```

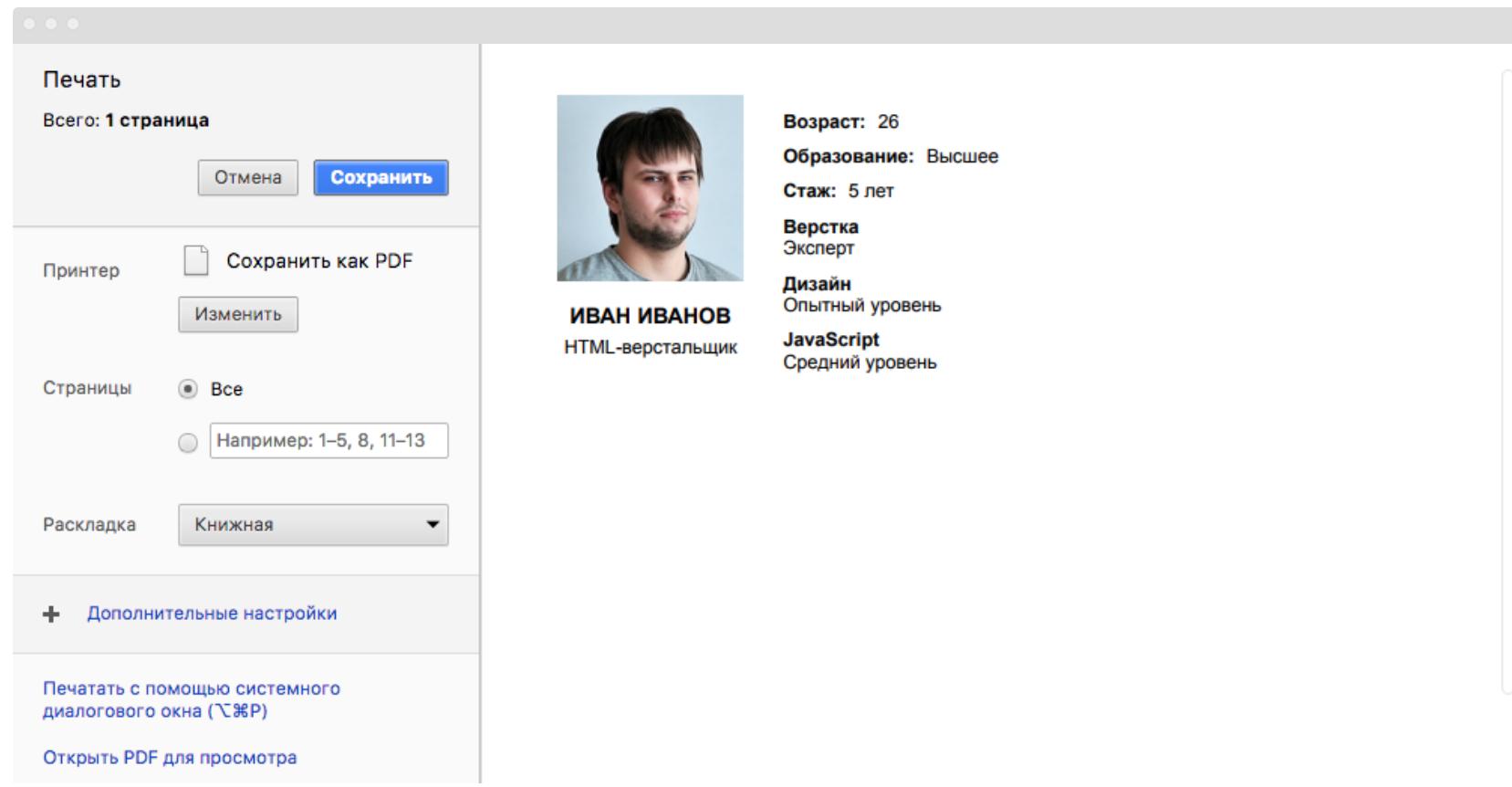
ПОДКЛЮЧАЕМ СТИЛИ ДЛЯ ПЕЧАТИ

Теперь подключим файл `print.css` только для печати. Как мы ранее рассматривали, для этого нужно добавить к тегу `<link>` атрибут `media`, в котором указать медиатип – `print`.

```
<link href="style.css" rel="stylesheet">
<link href="print.css" rel="stylesheet" media="print">
```

ПЕЧАТАЕМ СТРАНИЦУ

Работает! Графики скрылись, а на их место добавилось текстовое описание.



The screenshot shows a print dialog box from a web browser. On the left, the dialog includes fields for printer selection (set to 'Принтер'), page range (set to 'Все'), orientation ('Книжная'), and additional settings. On the right, the content of the page to be printed is displayed, featuring a profile picture of a man with dark hair and a beard, followed by his name 'ИВАН ИВАНОВ' and title 'HTML-верстальщик'. Below this, his details are listed: 'Возраст: 26', 'Образование: Высшее', 'Стаж: 5 лет', 'Верстка Эксперт', 'Дизайн Опытный уровень', and 'JavaScript Средний уровень'.

Печать

Всего: 1 страница

Отмена Сохранить

Принтер Сохранить как PDF

Изменить

Страницы Все
Например: 1–5, 8, 11–13

Раскладка Книжная

+ Дополнительные настройки

Печатать с помощью системного диалогового окна (⌘P)

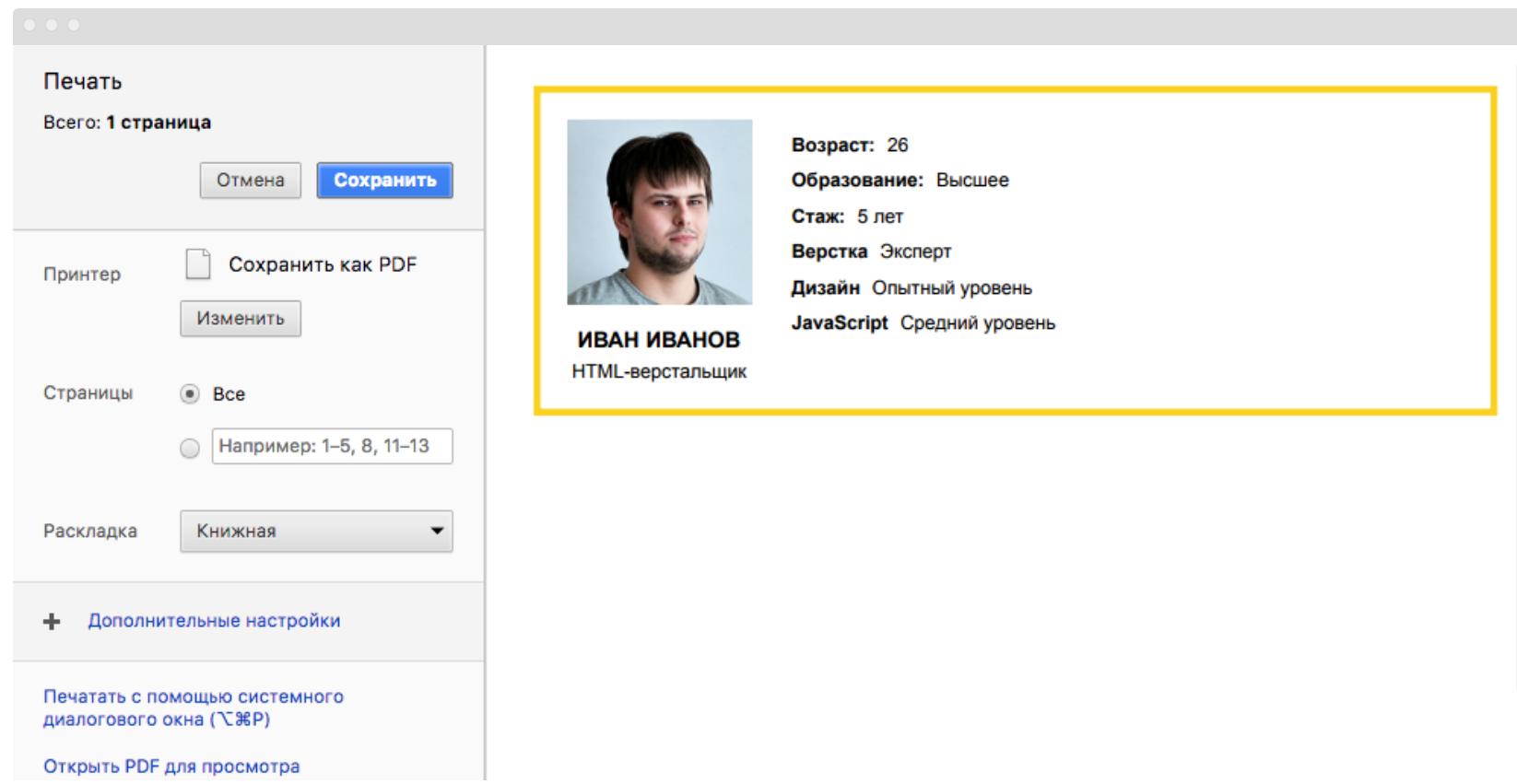
Открыть PDF для просмотра

ИВАН ИВАНОВ
HTML-верстальщик

Возраст: 26
Образование: Высшее
Стаж: 5 лет
Верстка
Эксперт
Дизайн
Опытный уровень
JavaScript
Средний уровень

НАВОДИМ ЛОСК

Отлично! Теперь данные сотрудника понятно отображаются как при обычном просмотре, так и при печати.



[Live Demo](#)

РЕЖИМ ЭМУЛЯЦИИ УСТРОЙСТВ В ВЕБ-ИНСПЕКТОРЕ

ТЕСТИРОВАНИЕ МОБИЛЬНОЙ ВЕРСТКИ

В процессе создания верстки нам необходимо проверить результат – как же выглядит наш код, в частности, на экране телефона. Но тестировать верстку на реальных устройствах чаще всего возможности нет, или это крайне затруднительно.

ВЕБ-ИНСПЕКТОР GOOGLE CHROME

Получить представление о том, как верстка работает на экранах мобильных устройств, нам поможет режим эмуляции в веб-инспекторе.

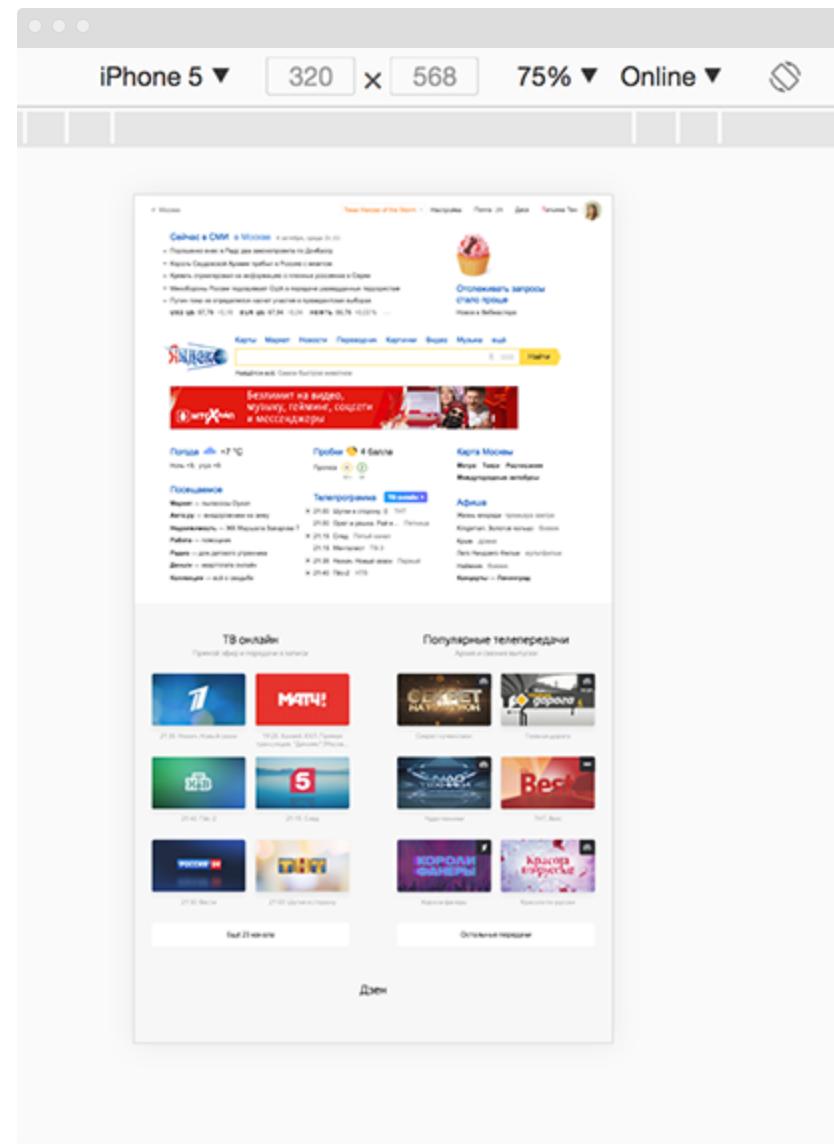
Данный режим есть во всех современных браузерах, однако давайте рассмотрим принцип тестирования на примере Google Chrome.

ВКЛЮЧАЕМ РЕЖИМ ЭМУЛЯЦИИ

Чтобы включить эмулятор устройств, нужно включить инструменты разработчика Google Chrome и кликнуть на иконку выбора режима эмуляции в веб-инспекторе:

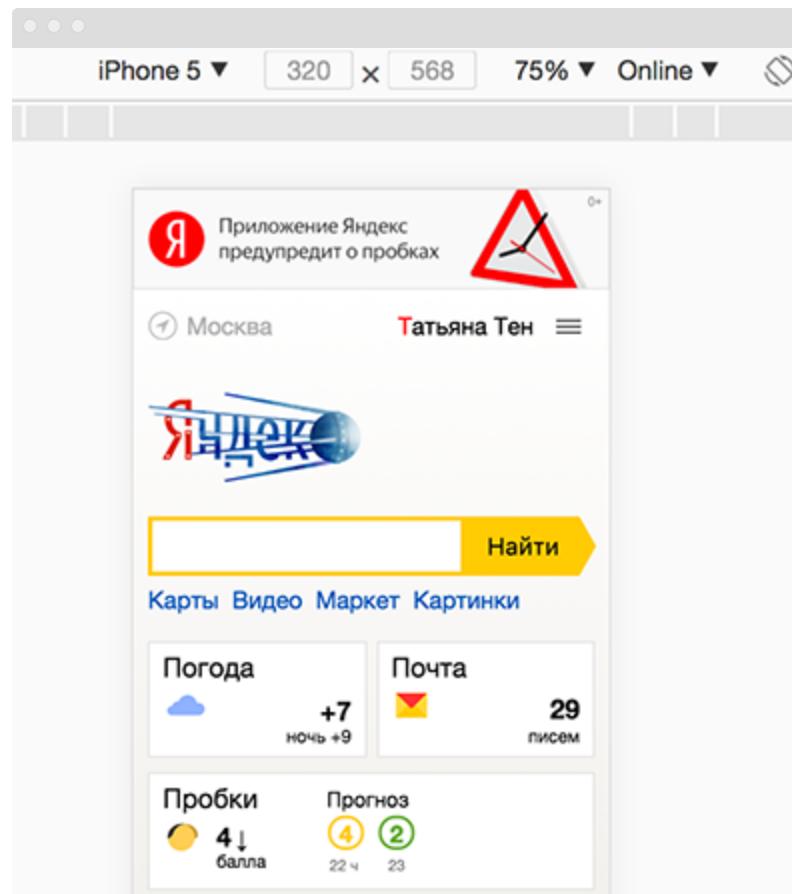


ИНТЕРФЕЙС ЭМУЛЯТОРА



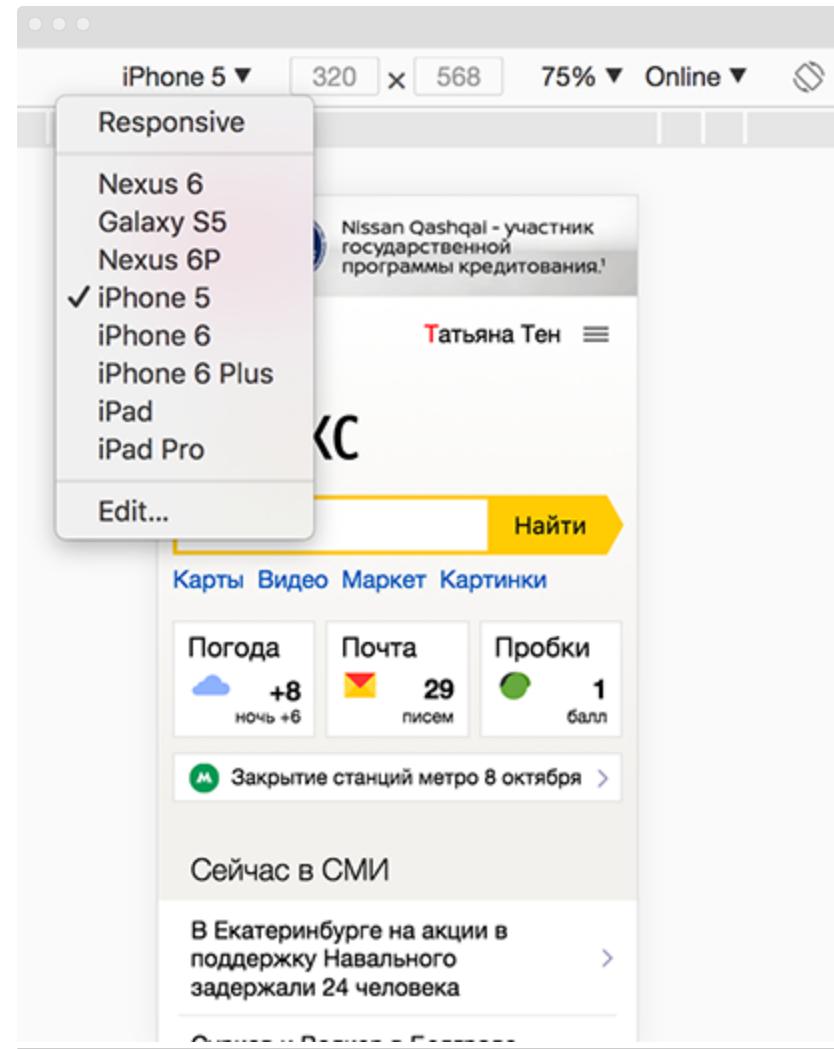
МОБИЛЬНАЯ ВЕРСИЯ

Окно имитирует внешний вид мобильного устройства. Но это все та же десктопная версия, только уменьшенная. Чтобы увидеть мобильную версию – нужно обновить страницу:



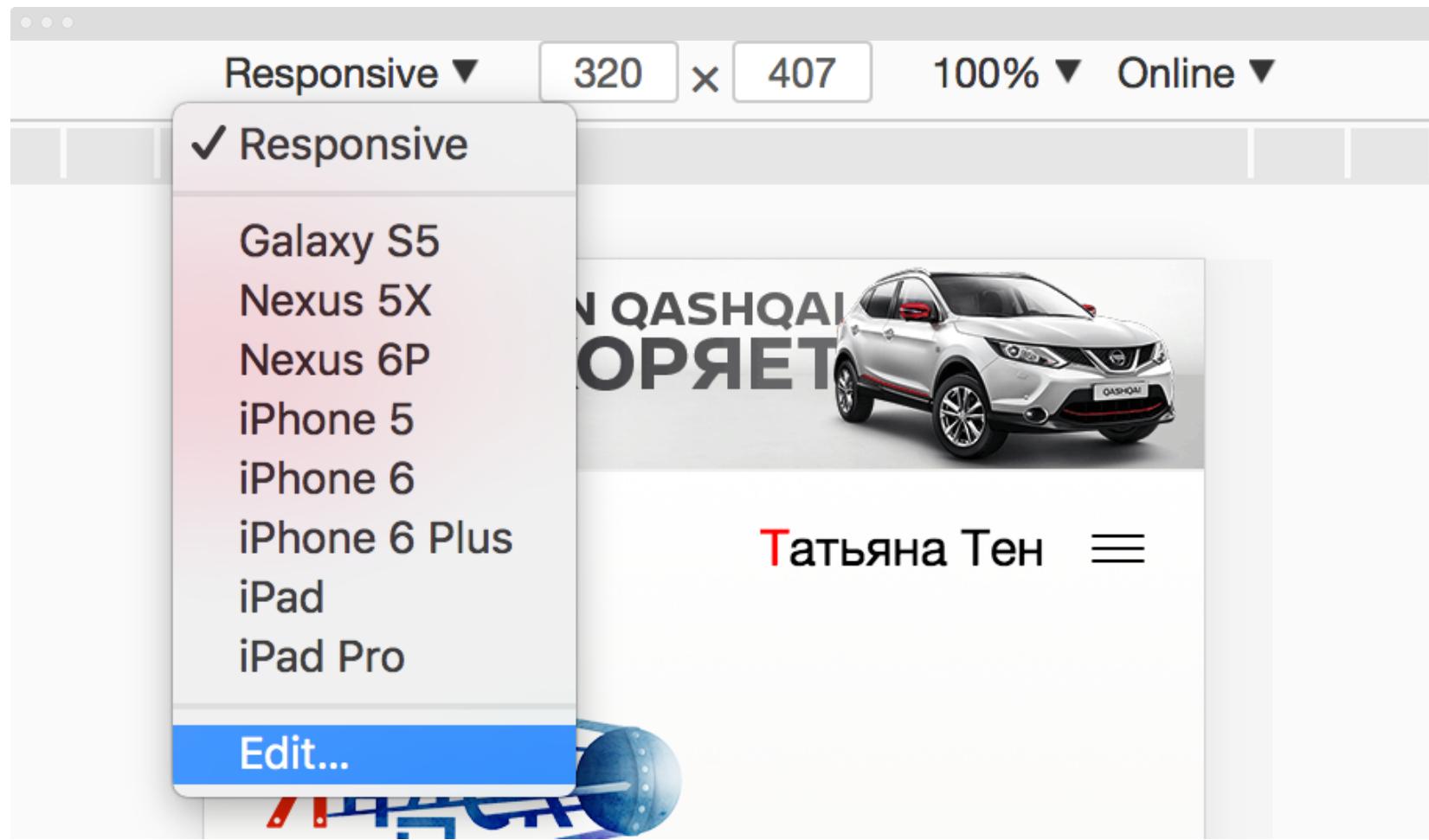
ВЫБОР МОБИЛЬНОГО УСТРОЙСТВА

Можно выбрать из списка какое устройство эмулировать:



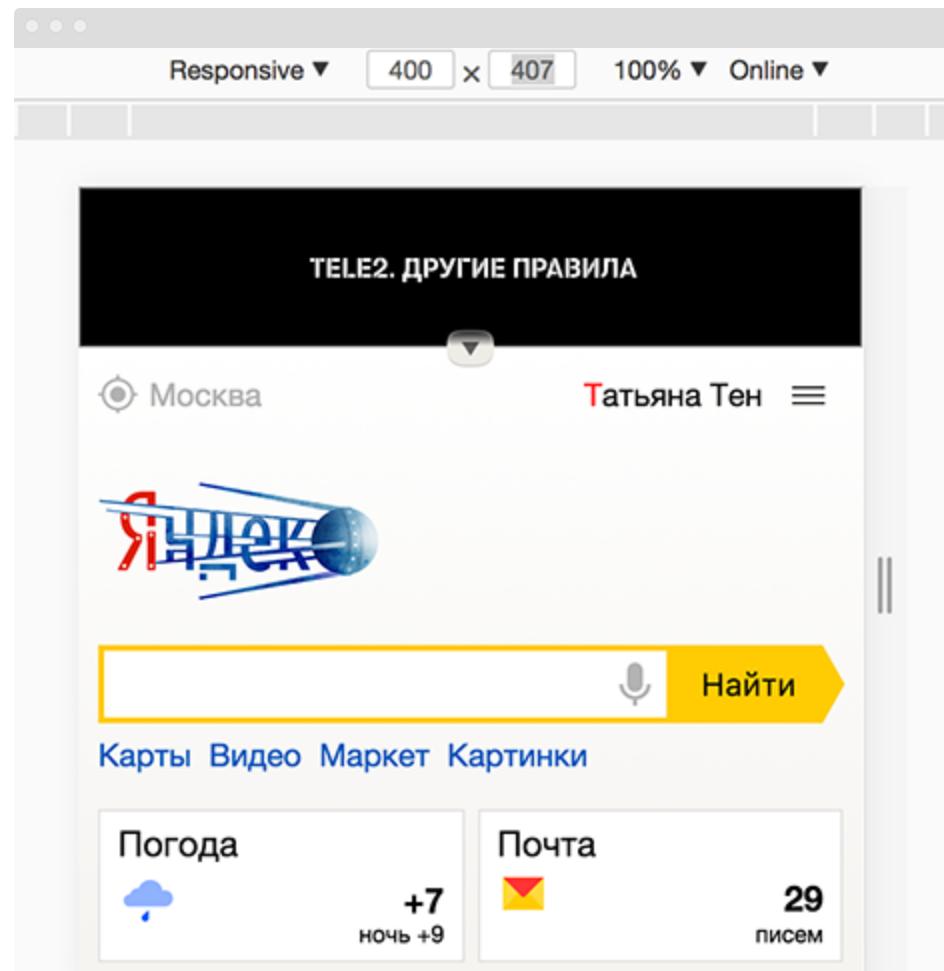
ДОБАВЛЯЕМ СВОИ УСТРОЙСТВА

Это не все доступные устройства, список можно отредактировать, кликнув на **Edit**:



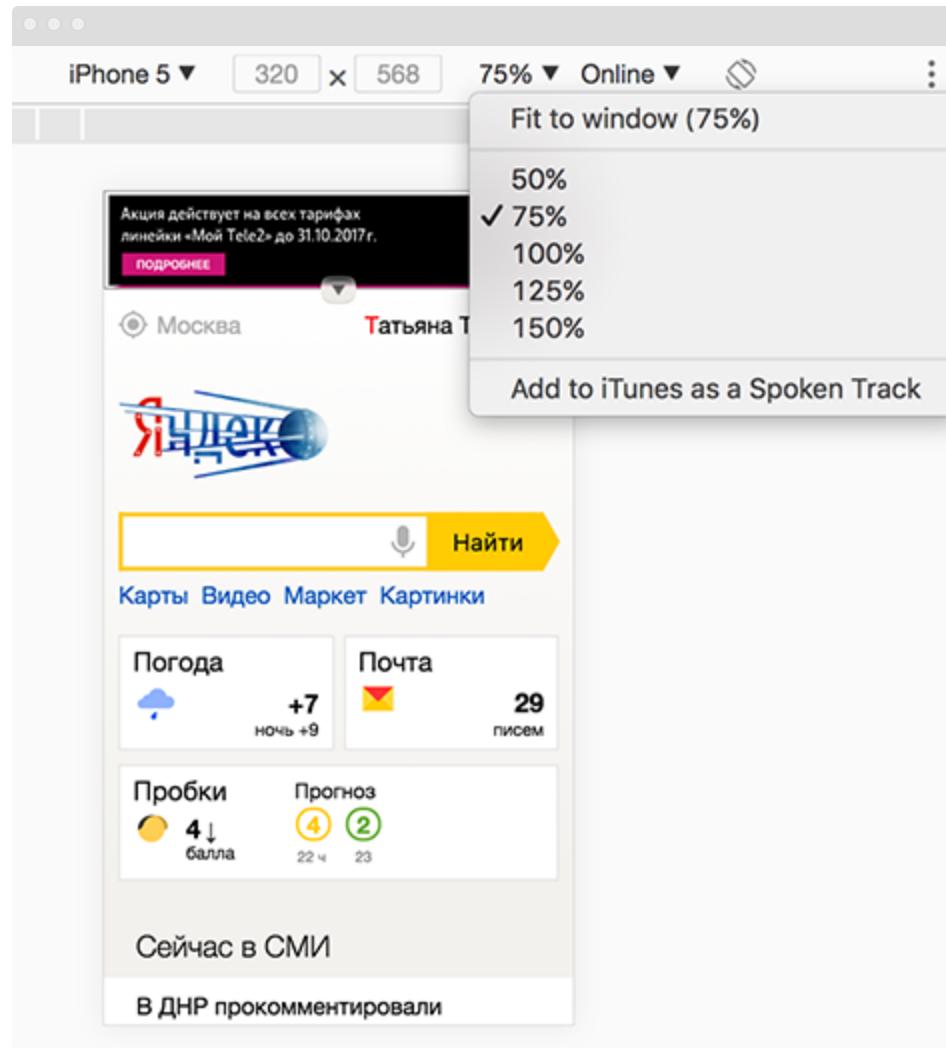
ПРОИЗВОЛЬНЫЕ РАЗМЕРЫ ЭКРАНА

Также можно задать произвольные размеры экрана, выбрав тип устройства **Responsive**:



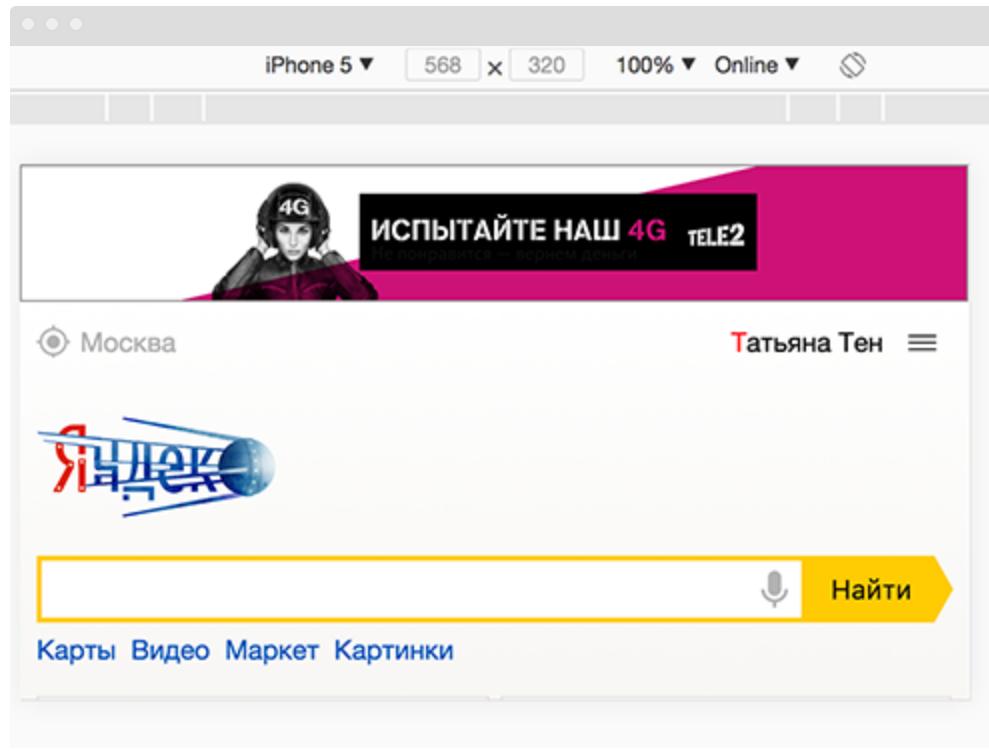
МЕНЯЕМ МАСШТАБ

Помимо этого, есть возможность изменять масштаб:



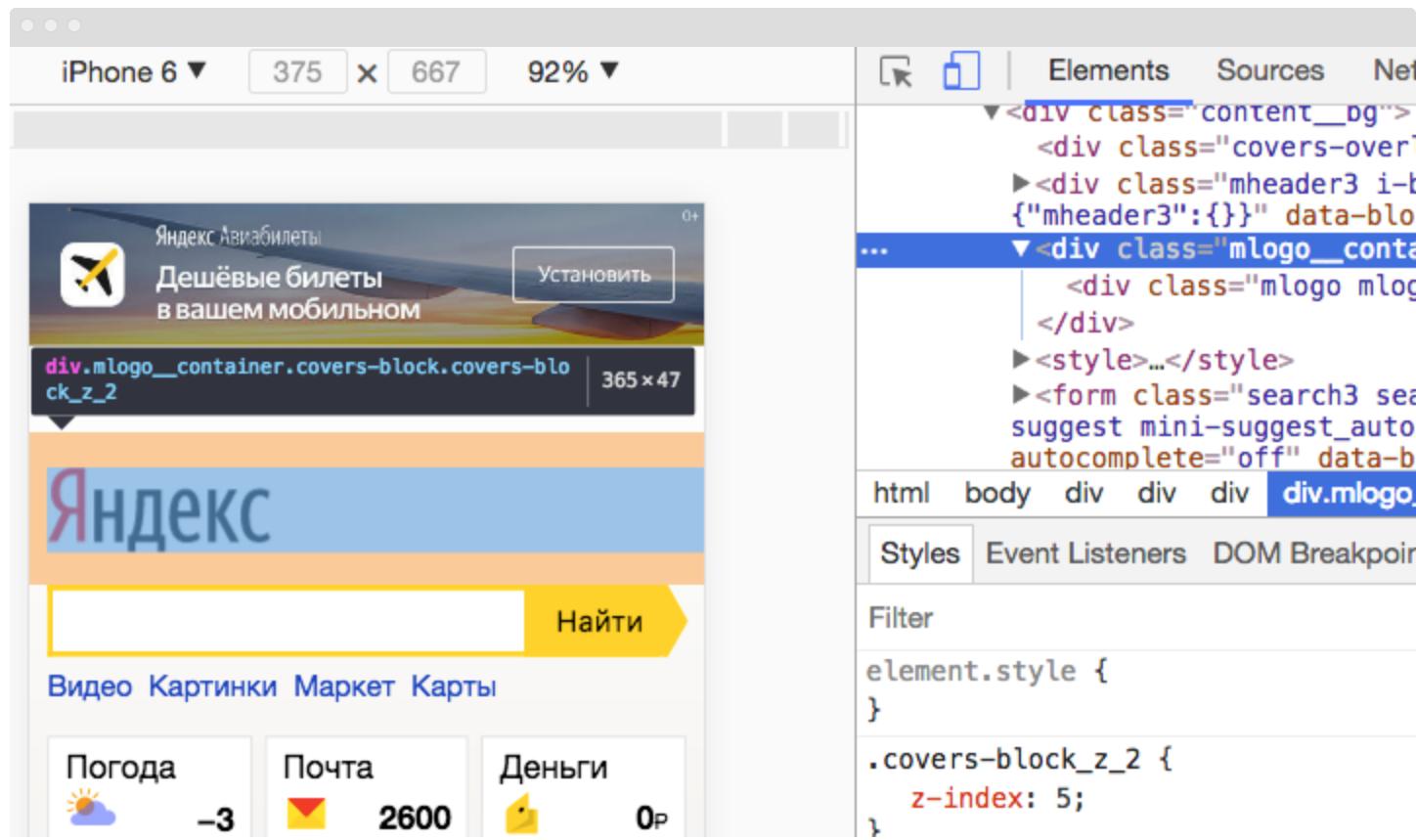
МЕНЯЕМ ОРИЕНТАЦИЮ ЭКРАНА

К тому же, мы можем изменить ориентацию эмулируемого устройства:



ВЗАЙМОДЕЙСТВУЕМ СО СТРАНИЦЕЙ

В режиме эмуляции можно не просто увидеть, как будет выглядеть верстка на разных устройствах, а еще и инспектировать любые элементы и менять стили на лету – делать все, что и обычно:



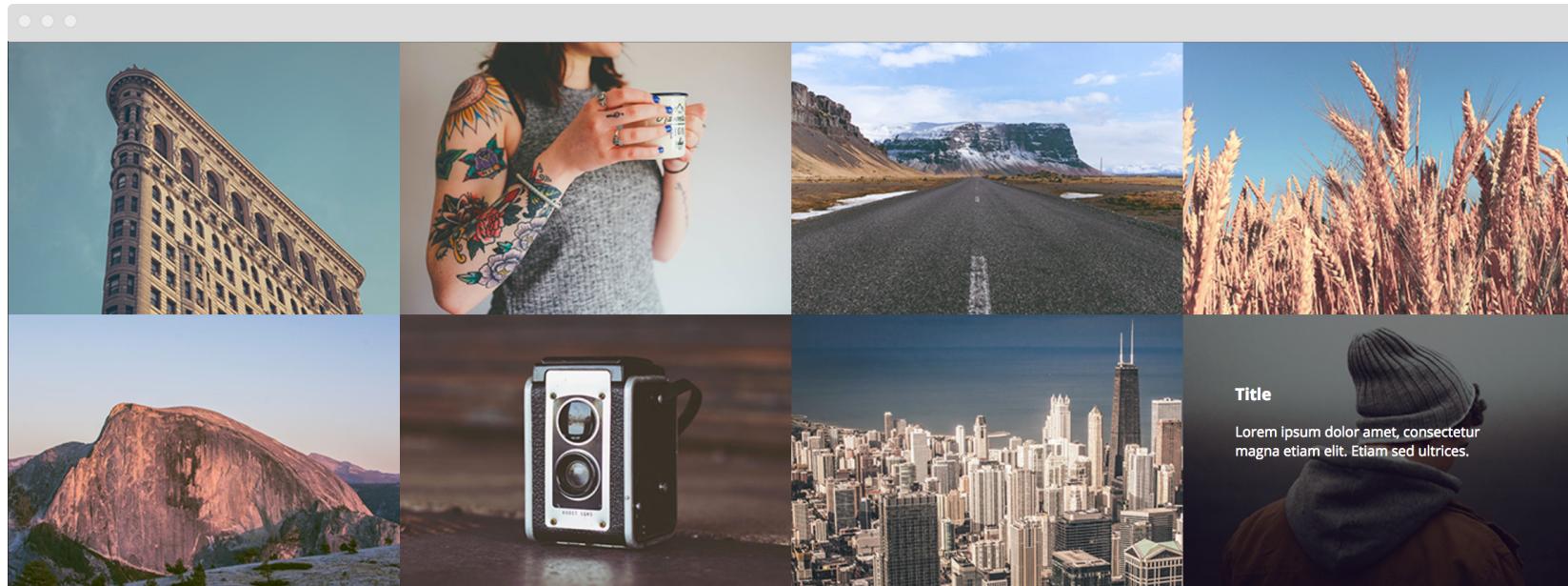
ПРЕДПОЧТЕНИЕ РЕАЛЬНЫМ УСТРОЙСТВАМ

Однако нужно помнить, что эмулятор не может передать всю специфику отображения верстки на реальных устройствах, поэтому, чтобы полностью быть уверенными в том, что верстка выглядит правильно на всех мобильных устройствах, лучше всего протестировать ваш код на реальных устройствах.

flex-wrap

ВЕРСТАЕМ ПОРТФОЛИО

Допустим, нам пришла задача сверстать блок портфолио:



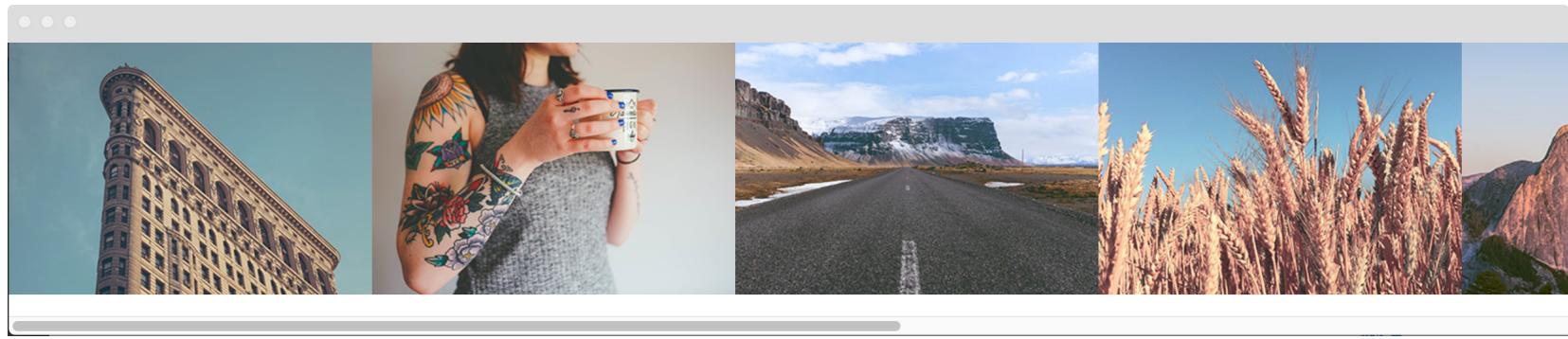
ИСПОЛЬЗУЕМ flex

Реализовывать блок мы решили при помощи флексбоксов:

```
1 .portfolio {  
2   display: flex;  
3   width: 1000px;  
4 }  
5  
6 .portfolio__item {  
7   width: 250px;  
8 }
```

НЕОЖИДАННЫЙ РЕЗУЛЬТАТ

Вместо того, чтобы переноситься на вторую строку, блоки выстроились в одну линию, и появилась горизонтальная прокрутка.



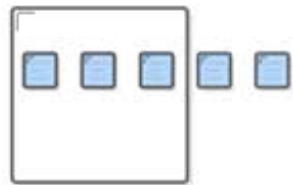
[Live Demo](#)

flex-wrap для многострочности

По умолчанию во flex-контейнере нет многострочности. Но на это можно повлиять, используя свойство `flex-wrap`.

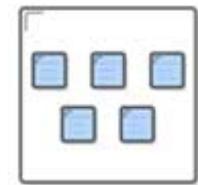


Свойство `flex-wrap` позволяет указать, является ли flex-контейнер односсточным или многострочным.



NO WRAPPING

FLEX-WRAP: NOWRAP;



WITH WRAPPING

FLEX-WRAP: WRAP;

Если контейнер односсточный, то flex-элементы не могут переходить на новую строку.

А если многострочный, то flex-элементы сначала будут располагаться в одну строку, а если они не поместятся, то могут переноситься, подобно тексту, на следующие строки.

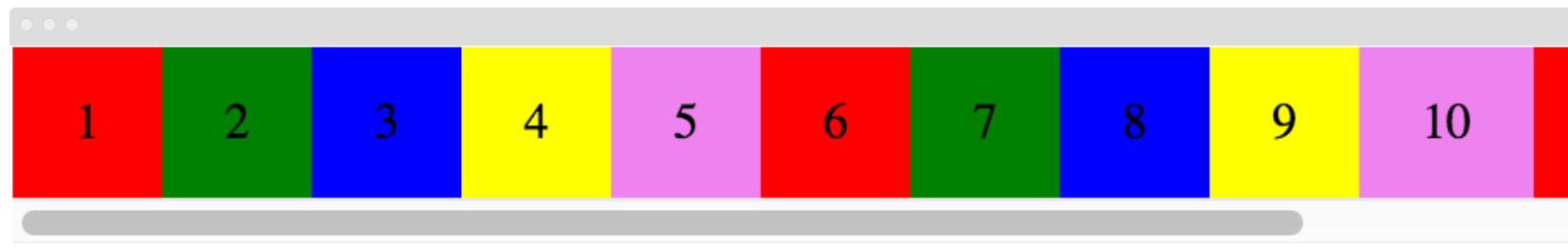
ДОСТУПНЫЕ ЗНАЧЕНИЯ

Свойство `flex-wrap` имеет следующие значения:

- `nowrap`;
- `wrap`;
- `wrap-reverse`.

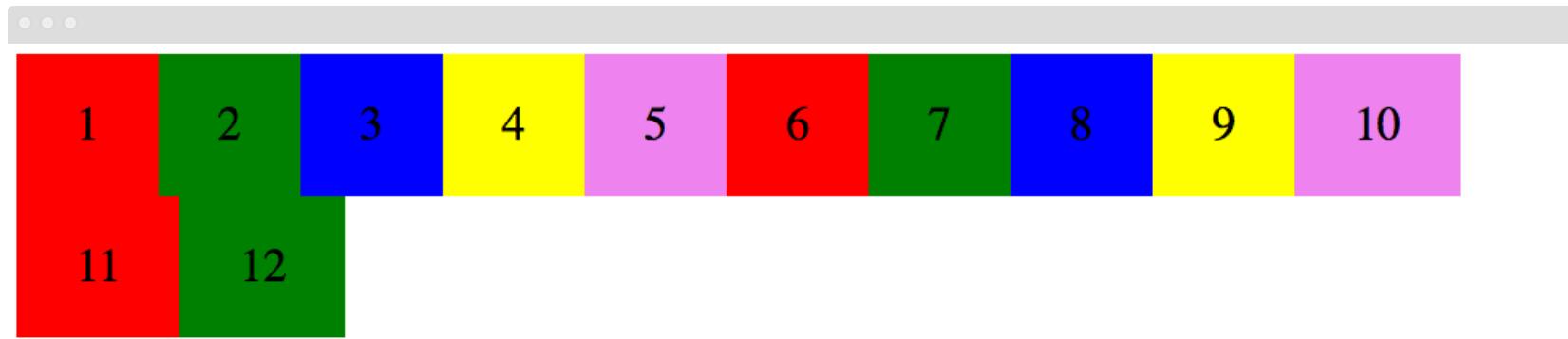
nowrap

Данное значение установлено по умолчанию и его можно не указывать. В этом случае элементы выстраиваются в одну строку, и если их будет достаточно много, то вызывают переполнение контейнера и выходят за его пределы.



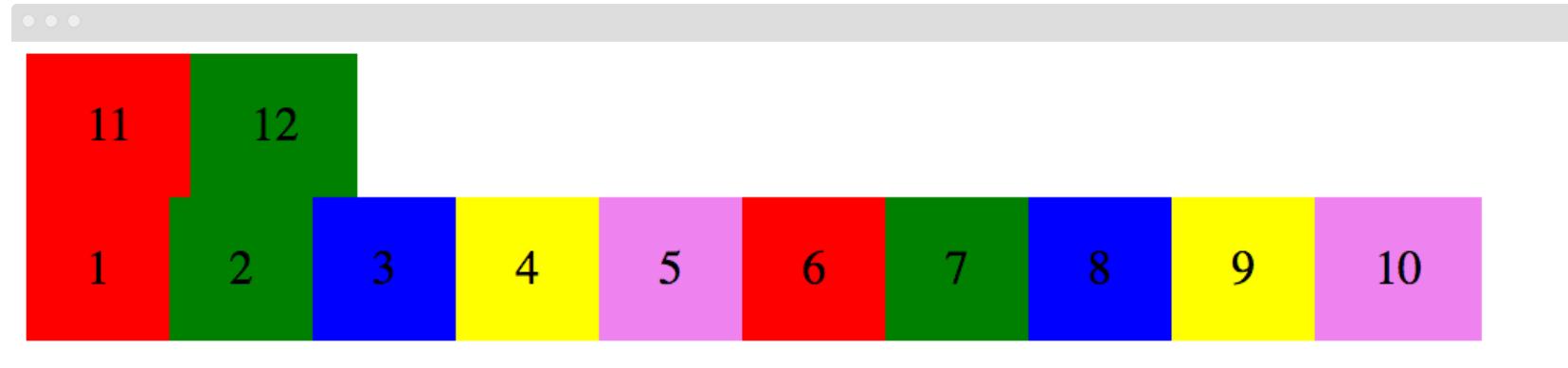
wrap

Значение `wrap` позволяет элементам переноситься на следующую строку:



wrap-reverse

Значение `wrap-reverse` также позволяет блокам переноситься, но при этом меняется направление дополнительной оси, благодаря чему порядок строк по вертикали изменяется на «снизу вверх»:



align-content



Свойство `align-content` позволяет выровнять строки с контентом внутри многострочного flex-контейнера. Рассмотрим возможные значения этого свойства.



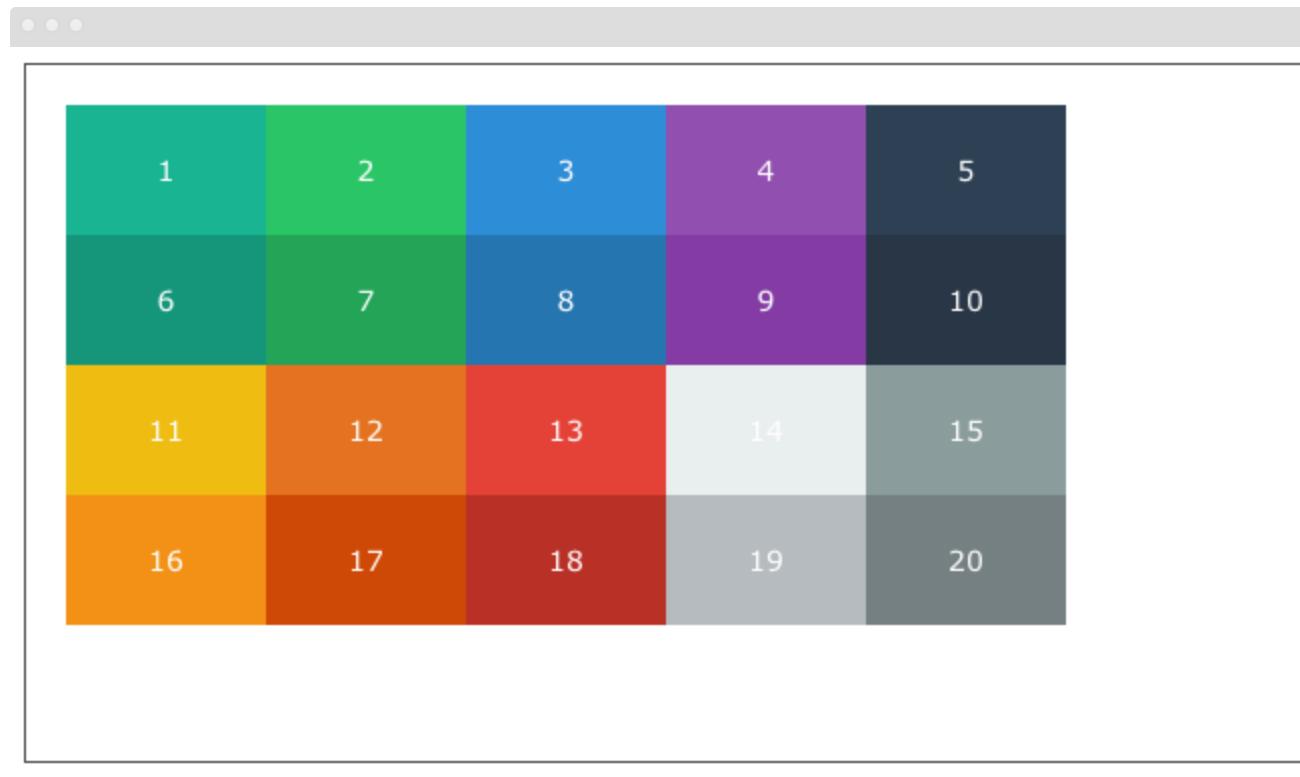
`align-content`

ДОСТУПНЫЕ ЗНАЧЕНИЯ

- `flex-start`;
- `center`;
- `flex-end`;
- `space-between`;
- `space-around`;
- `stretch`.

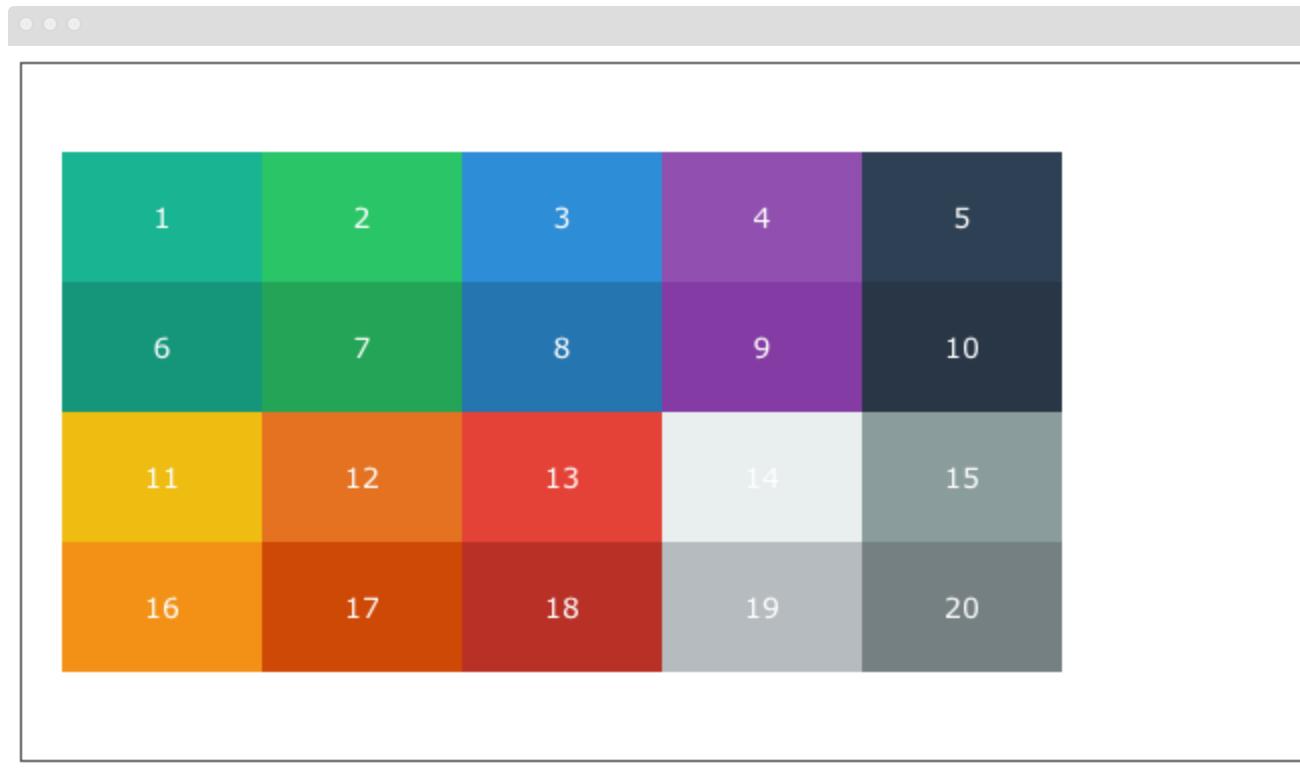
flex-start

При `align-content: flex-start` flex-элементы выравниваются, располагаясь от левого верхнего угла (то есть, от начала) контейнера:



center

При `align-content: flex-center` flex-элементы располагаются по центру контейнера:



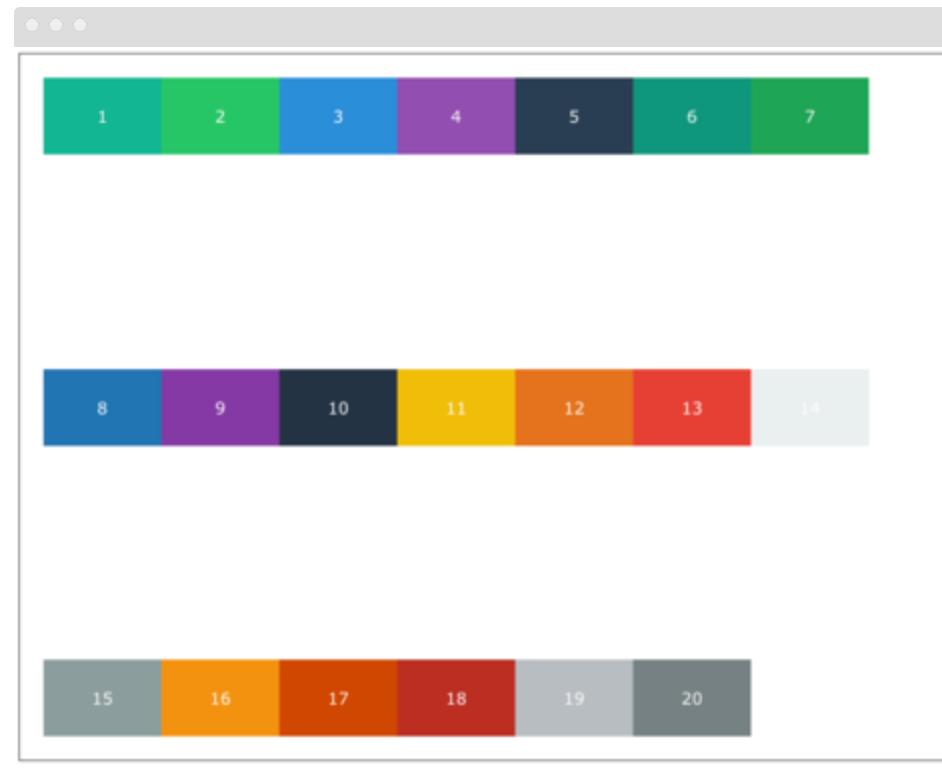
flex-end

При `align-content: flex-end` flex-элементы выравниваются от правого нижнего угла (от конца) контейнера:



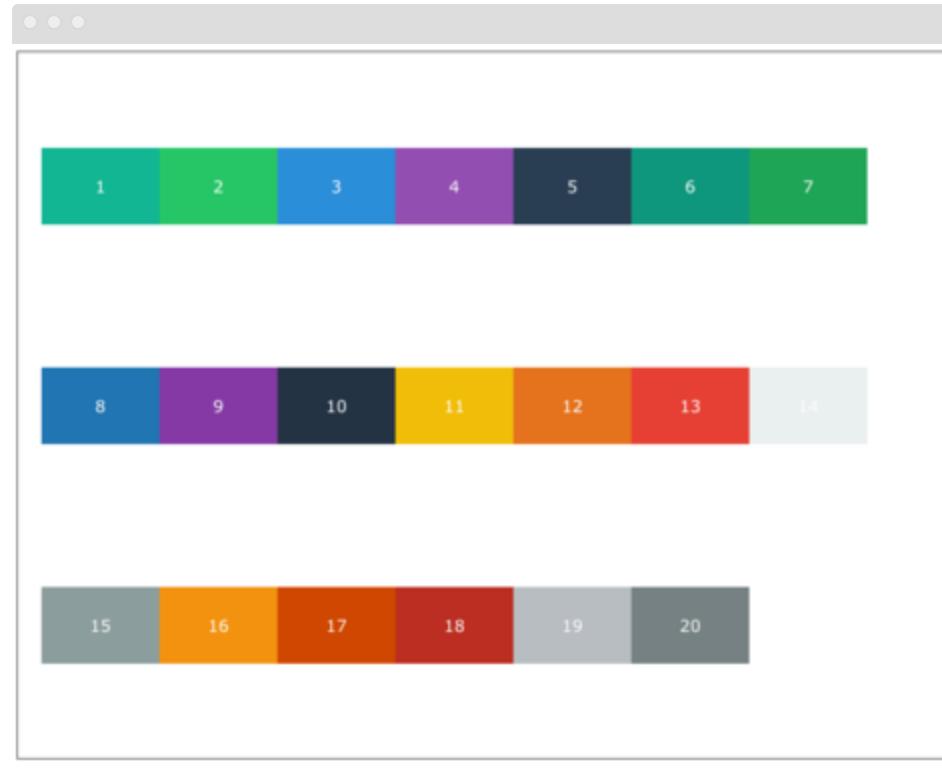
space-between

Строчки распределяются по контейнеру равномерно, так, что первая и последняя из них находятся у верхнего и нижнего края контейнера безо всякого отступа:



space-around

Строчки распределяются по контейнеру равномерно, так что как между ними, так и от верхнего края контейнера до первой строчки и от нижнего края контейнера до последней строчки – равное пространство:



stretch

flex-элементы растягиваются по высоте, при этом строки тянутся пропорционально количеству контента таким образом, чтобы занимать все доступное пространство. Это значение является значением по умолчанию, его можно не указывать:



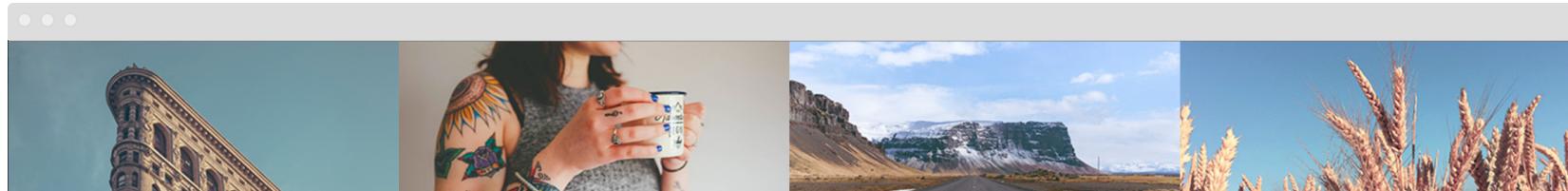
ВНОСИМ ПРАВКИ В ВЕРСТКУ

Вернемся к нашему примеру и попробуем исправить его:

```
1 .portfolio {  
2   display: flex;  
3   flex-wrap: wrap;  
4   align-content: stretch;  
5 }
```

ФИНАЛЬНЫЙ РЕЗУЛЬТАТ

Теперь верстка выглядит как надо:



[Live Demo](#)

ИТОГИ

ОТДЕЛЬНАЯ МОБИЛЬНАЯ ВЕРСИЯ САЙТА

- Один из подходов к решению проблемы адаптации сайта к разным устройствам – разработка отдельной мобильной версии помимо версии для десктопов.
- Когда пользователь заходит на сайт, определяется тип устройства. И если это мобильное устройство – происходит переадресация на мобильную версию.
- Плюсы: упрощенный внешний вид и урезанный функционал позволяют сайту быстрее загружаться и предоставляют пользователю только самый важный контент, удобная навигация с сенсорных устройств.
- Минусы: необходимость разработки, поддержки и синхронизации двух независимых версий, повторение контента по двум разным адресам, сложность разработки дизайна.

МЕДИАЗАПРОСЫ

- Медиазапросы – возможность браузера выбрать устройства, отвечающие заданным параметрам, и применить к ним отдельный CSS.
- Для их использования нужно добавить атрибут `media` к тегу `<link>` и указать значение типа устройства.
- Медиатипы:
 - `screen` (обычный режим просмотра),
 - `print` (режим печати),
 - `speech` (режим озвучивания текста),
 - `all` (по умолчанию – режим обычного просмотра, печати или озвучивания текста).

РЕЖИМ ЭМУЛЯЦИИ УСТРОЙСТВ В ВЕБ-ИНСПЕКТОРЕ

- Для тестирования мобильной верстки можно использовать режим эмуляции в веб-инспекторе. Он есть во всех современных браузерах, в том числе в Google Chrome.
- Для включения эмулятора нужно выбрать режим эмуляции в веб-инспекторе инструментов разработчика Google Chrome.
- Важно помнить: эмулятор не может передать всю специфику отображения верстки на мобильных устройствах, поэтому для полной уверенности лучше протестировать код на реальных устройствах.

flex-wrap

- По умолчанию во flex-контейнере нет многострочности. Но на это можно повлиять с помощью свойства `flex-wrap`.
- Оно позволяет указать, является ли flex-контейнер односторонним или многострочным.
- Если контейнер односторонний, то flex-элементы не могут переходить на новую строку, если многострочный, то сначала они будут располагаться в одну строку, а если не поместятся, то могут переноситься на следующие строки.
- Значения свойства `flex-wrap`:
 - `nowrap` (по умолчанию),
 - `wrap` (позволяет перенос на следующую строку),
 - `wrap-reverse` (позволяет перенос, меняет порядок строк на «снизу вверх»).

align-content

- Свойство `align-content` позволяет выровнять строки с контентом внутри многострочного flex-контейнера.
- Значения свойства `align-content` определяют, каким образом выравниваются строки внутри flex-контейнера:
 - `flex-start` (от начала контейнера),
 - `center` (по центру контейнера),
 - `flex-end` (от конца контейнера),
 - `space-between` (распределяются по контейнеру равномерно, расстояние между строками одинаковое),
 - `space-around` (распределяются равномерно так, чтобы пространство между двумя соседними строками было одинаковым),
 - `stretch` (равномерно растягиваются).



Задавайте вопросы и напишите отзыв о лекции!

СЕМЕН БОЙКО



simonderus@gmail.com



<https://www.facebook.com/simonboycko>