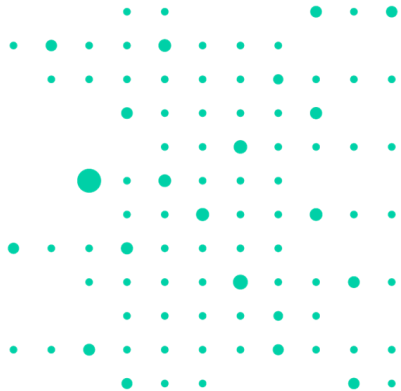
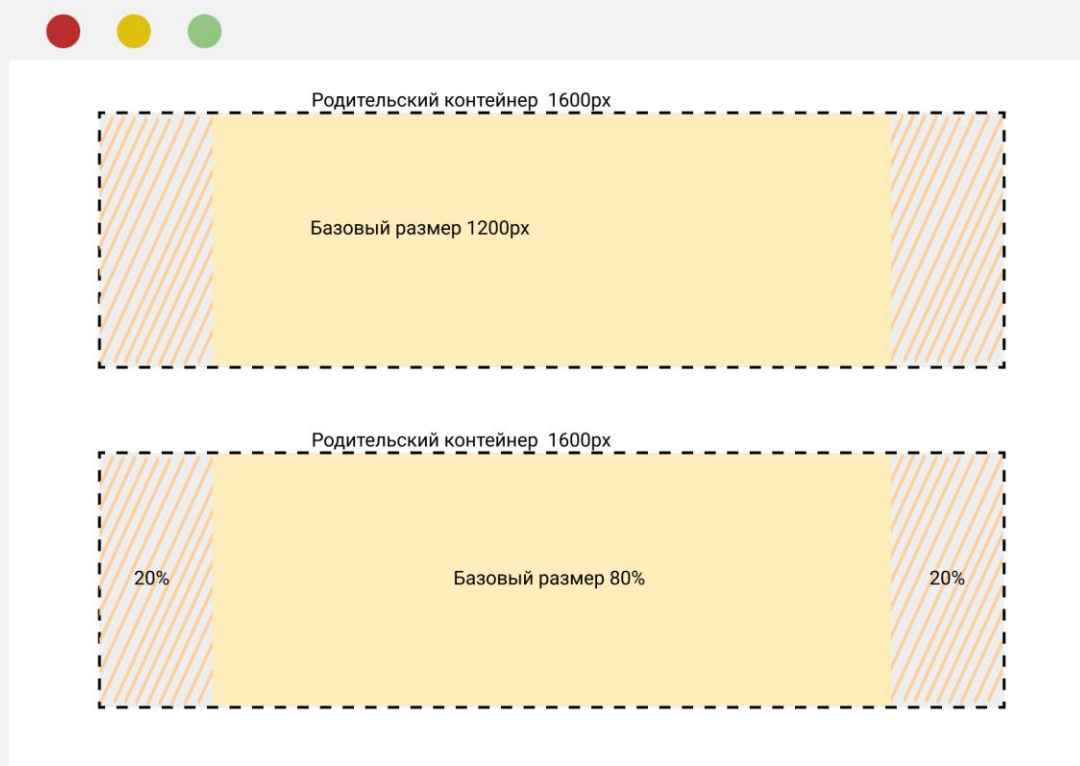




**Advanced adaptive,  
using Flexbox, Grid**

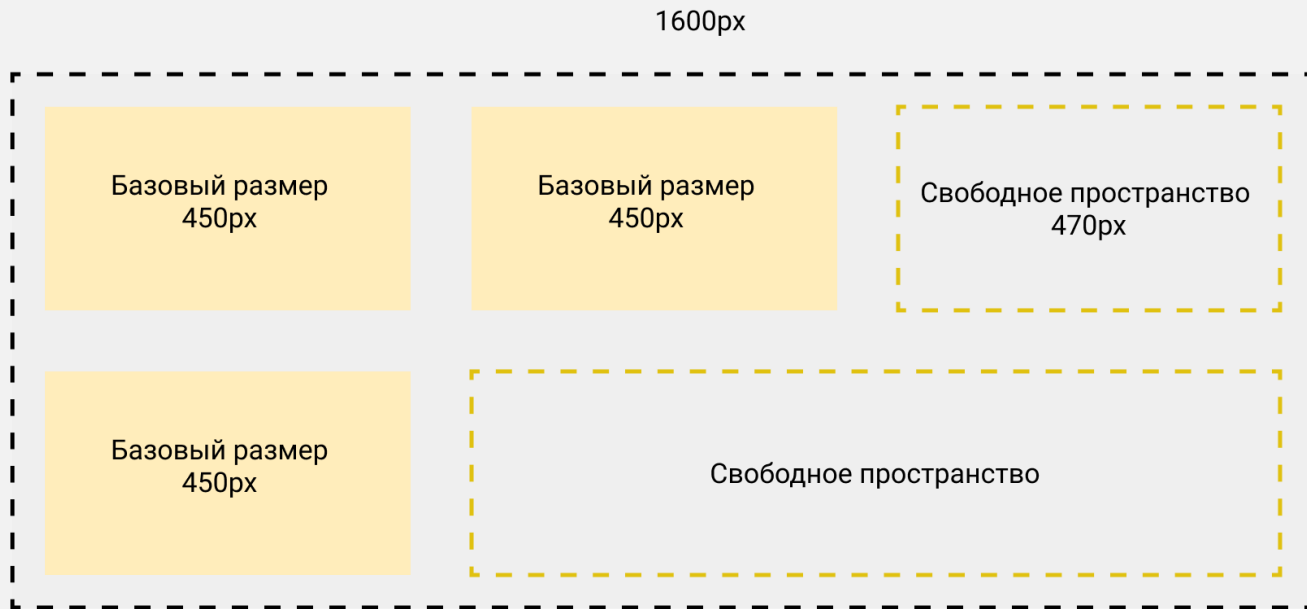


# ПЕРЕХОД ОТ ТОЧНЫХ ЗНАЧЕНИЙ PX НА ПРОЦЕНТЫ



# FLEX-WRAP: WRAP И ФИКСИРОВАННЫЙ БАЗОВЫЙ РАЗМЕР

---



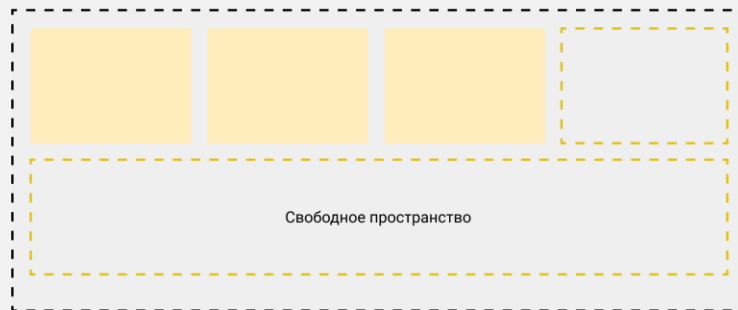
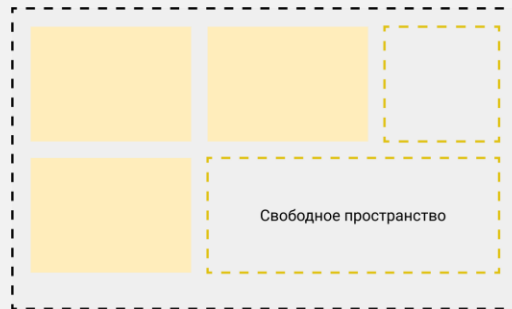
# FLEX-WRAP: WRAP И ФИКСИРОВАННЫЙ БАЗОВЫЙ РАЗМЕР

---



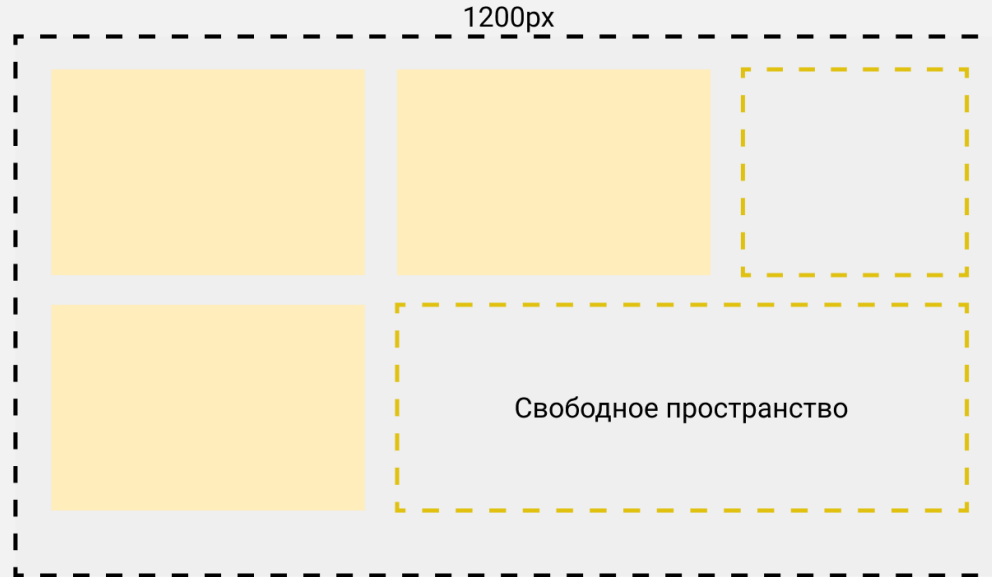
# АДАПТИВНОСТЬ «ОТ КОНТЕЙНЕРА»

---



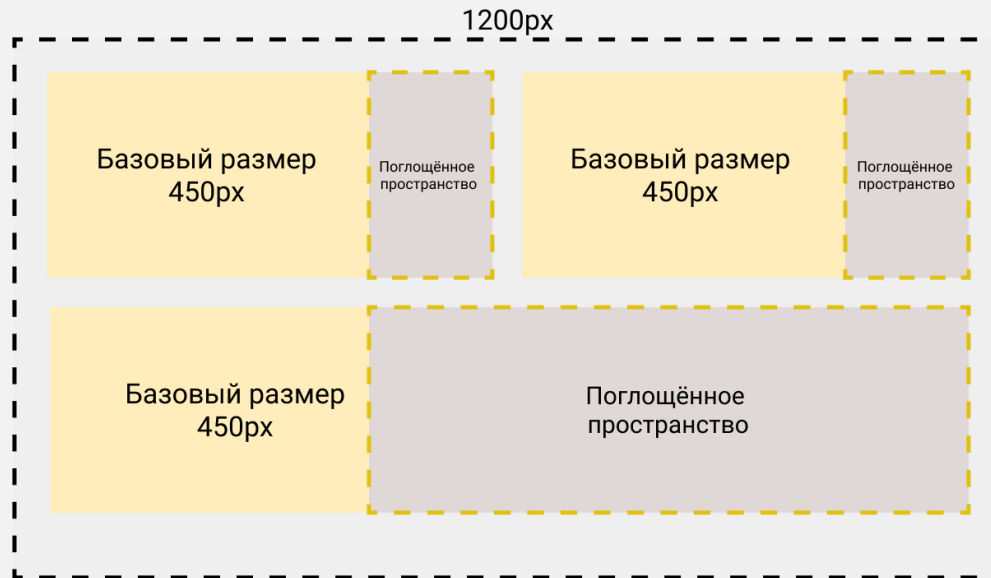
# FLEX-WRAP: WRAP, FLEX-GROW: 0, ФИКСИРОВАННЫЙ БАЗОВЫЙ РАЗМЕР

---



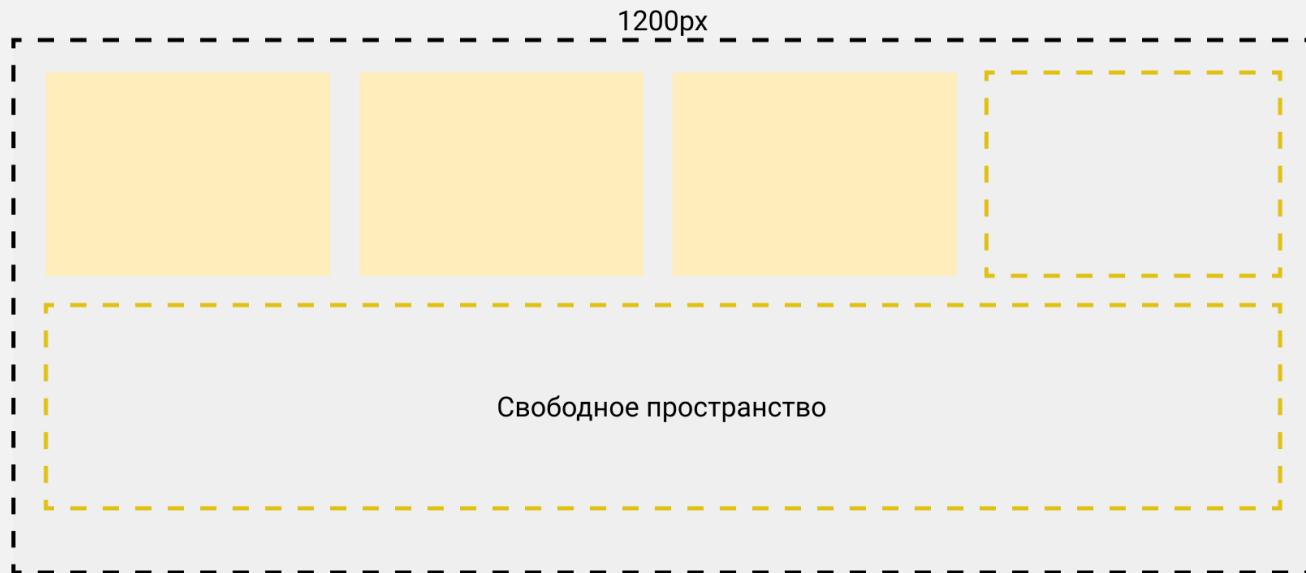
# FLEX-WRAP: WRAP, FLEX-GROW: 1, ФИКСИРОВАННЫЙ БАЗОВЫЙ РАЗМЕР

---



# FLEX-WRAP: WRAP, FLEX-GROW: 0, ФИКСИРОВАННЫЙ БАЗОВЫЙ РАЗМЕР

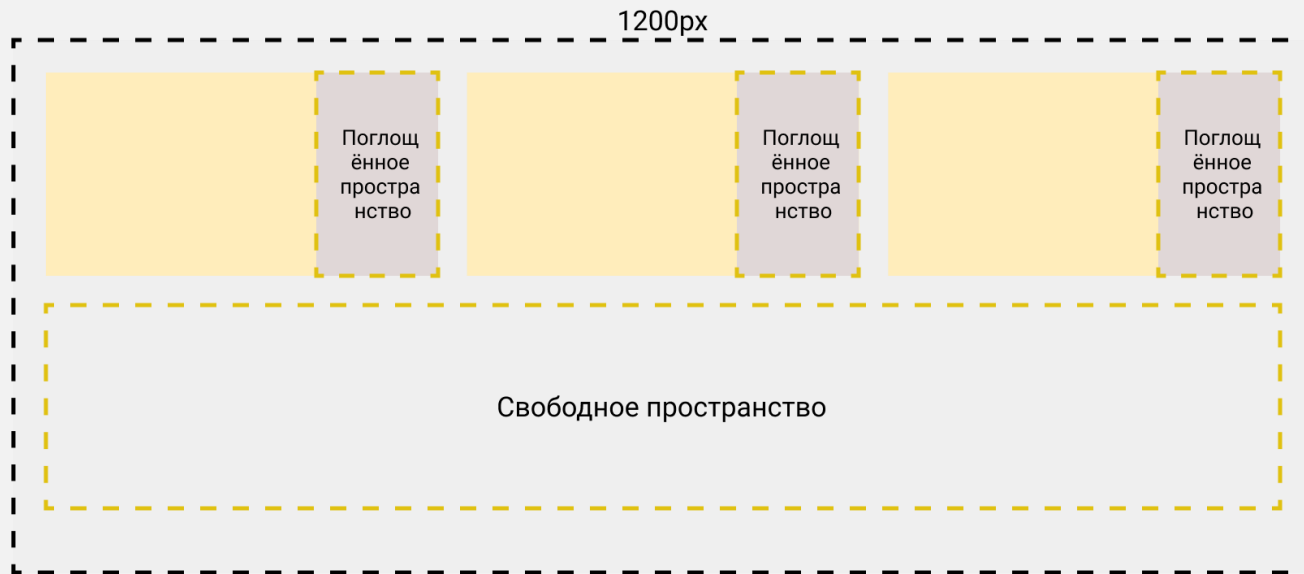
---





# FLEX-WRAP: WRAP, FLEX-GROW: 1, ФИКСИРОВАННЫЙ БАЗОВЫЙ РАЗМЕР

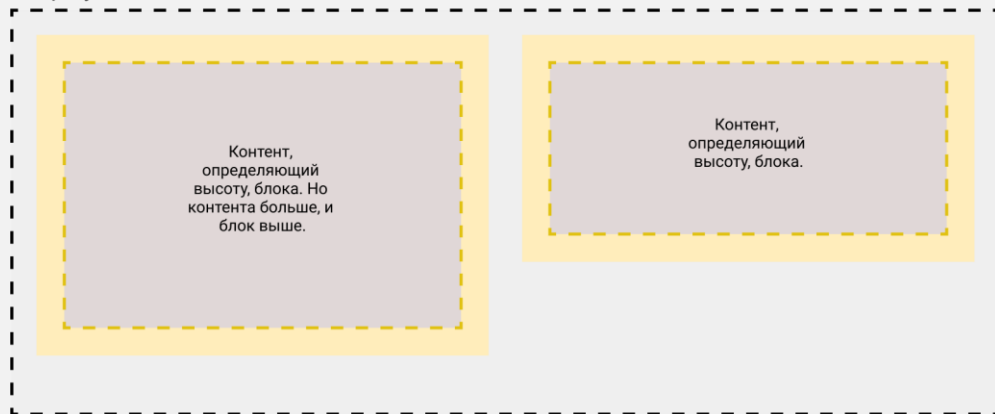
---



# ВЛОЖЕННЫЕ ФЛЕКСЫ

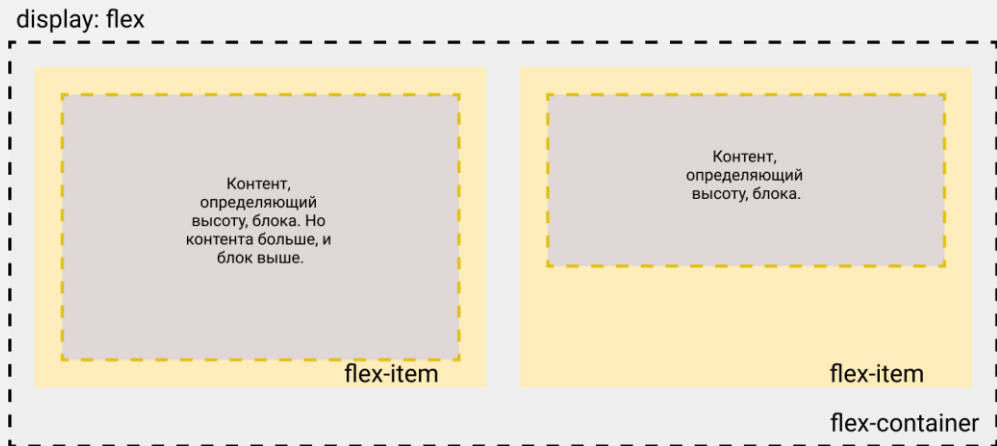
---

display: flex



# ВЛОЖЕННЫЕ ФЛЕКСЫ

---



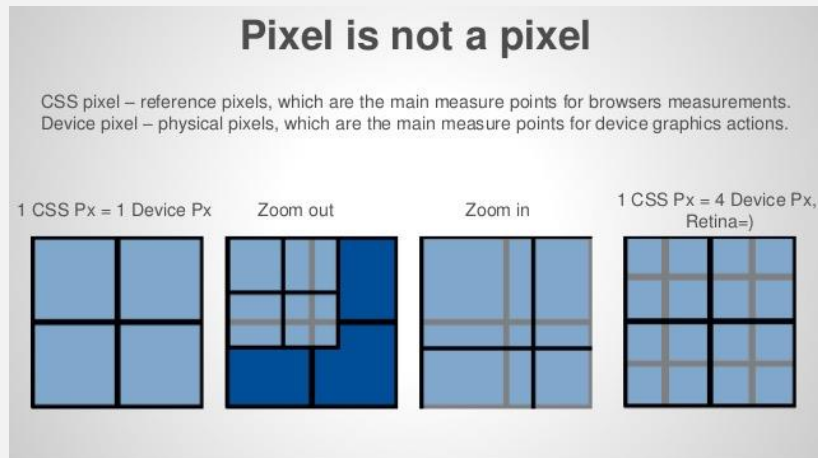


# **АДАПТИВНАЯ ГРАФИКА**

# АДАПТИВНАЯ ГРАФИКА

## ЭКРАННЫЕ ПИКСЕЛИ И CSS- ПИКСЕЛИ

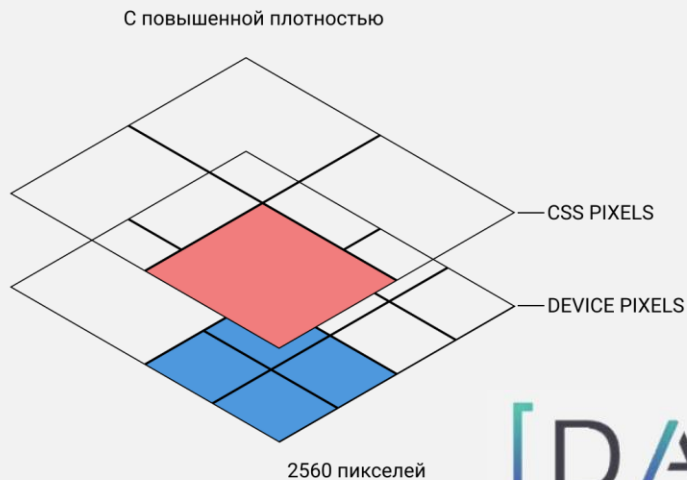
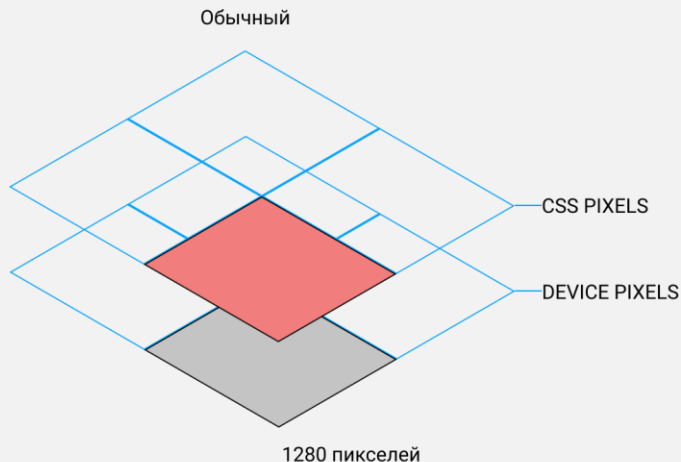
- Физические пиксели экрана и CSS-пиксели разные сущности, их размеры редко совпадают.
- Это может происходить при зуме или на современных экранах.



# АДАПТИВНАЯ ГРАФИКА

## ЭКРАНЫ С ПОВЫШЕННОЙ ПЛОТНОСТЬЮ ПИКСЕЛЕЙ

Количество физических пикселей больше, чем у обычного экрана, а сами пиксели меньше.



# АДАПТИВНАЯ ГРАФИКА

---

**Ретина** - это экраны с повышенной плотностью пикселей.

**Физические пиксели (device pixel или physical pixel)** — привычные нам пиксели: самые маленькие элементы любого дисплея, каждый из которых имеет свой цвет и яркость.

**Плотность экрана (Screen density)** — это количество физических пикселей дисплея. Обычно измеряется в пикселях-на-дюйм (PPI: pixels per inch). Apple, разработав Retina-экраны с двойной плотностью пикселей, утверждает, что человеческий глаз не способен различить бо'льшую плотность.

# АДАПТИВНАЯ ГРАФИКА

---

**CSS-пиксели (CSS pixels)** — абстрактная величина, используемая браузерами для точного отображения контента на страницах, вне зависимости от экрана (DIPs: device-independent pixels).

Пример: `<div height="200" width="300"></div>`



# АДАПТИВНАЯ ГРАФИКА

---

## ПРОБЛЕМЫ ОТОБРАЖЕНИЯ ГРАФИКИ НА ОБЫЧНЫХ И RETINA ДИСПЛЕЯХ

Графику для ретина-экранов надо специальным образом подготавливать, чтобы она не выглядела «замыленной».



# ВЕКТОРНАЯ ГРАФИКА

---

Векторный формат изображения на основе XML для 2D-графики, который поддерживает интерактивность и анимацию. Разрабатывается с 1999 года. Задуман и реализован для того, чтобы описать с помощью языка программирования графические элементы сайта.

**Акроним:** Scalable Vector Graphics.

**Тип изображения:** векторный.

**Тип сжатия:** обычно не сжимается, но может быть сжат без потерь.

# ВЕКТОРНАЯ ГРАФИКА ПЛЮСЫ

---

- ✓ Отзывчивость.
- ✓ SVG — это код, легко редактируется руками.
- ✓ Работает в браузере, можно прямо в HTML, включает стили и скрипты.
- ✓ Меньший размер по сравнению с растровыми аналогами и жмётся на сервере gzip.

# ВЕКТОРНАЯ ГРАФИКА МИНУСЫ

---

- ✗ Не поддерживается в IE  $\leq 8$ , может работать с ограничениями в IE 9-11.
- ✗ Подходит только для простой графики.
- ✗ SVG — это технология, а в любой технологии есть баги.

# ВАРИАНТЫ ИСПОЛЬЗОВАНИЯ (РЕАЛИЗАЦИИ)

---

## ПОДКЛЮЧАЕМ SVG В <IMG>

Подходит для контентных изображений

// Работают встроенные стили, но скрипты нет

```

```

# ВАРИАНТЫ ИСПОЛЬЗОВАНИЯ (РЕАЛИЗАЦИИ)

---

## ПОДКЛЮЧАЕМ SVG В CSS

Подходит для оформительских изображений

// Работают встроенные стили, но скрипты нет

```
.background {  
  width: 200px;  
  background-repeat: no-repeat;  
  background-image: url('picture.svg');  
  background-size: 200px auto;  
}
```

# ВАРИАНТЫ ИСПОЛЬЗОВАНИЯ (РЕАЛИЗАЦИИ)

---

## SVG + OBJECT/EMBED/IFRAME

- Подходит для интерактивных SVG.
- Работают встроенные стили и скрипты.
- Можно управлять из JavaScript

# ВАРИАНТЫ ИСПОЛЬЗОВАНИЯ (РЕАЛИЗАЦИИ)

---

## ПОДКЛЮЧАЕМ SVG В IFRAME

- Неверно с идеологической точки зрения.
- Iframe не подстраивается под размер содержимого.

```
<iframe src="picture.svg" width="400" height="400"></ iframe>
```



# ВАРИАНТЫ ИСПОЛЬЗОВАНИЯ (РЕАЛИЗАЦИИ)

---

## ПОДКЛЮЧАЕМ SVG В OBJECT И EMBED

- Позволяют встраивать не-HTML содержимое в HTML документ.

```
<object data="picture.svg" type="image/svg+xml" width="200" height="200"></object>
```

```
<embed src="picture.svg" width="200" height="200">
```

# ВАРИАНТЫ ИСПОЛЬЗОВАНИЯ (РЕАЛИЗАЦИИ)

---

## ПОДКЛЮЧАЕМ SVG В OBJECT И EMBED

- Позволяют встраивать не-HTML содержимое в HTML документ.

```
<object data="picture.svg" type="image/svg+xml" width="200" height="200"></object>
```

```
<embed src="picture.svg" width="200" height="200">
```

# ВАРИАНТЫ ИСПОЛЬЗОВАНИЯ (РЕАЛИЗАЦИИ)

---

## ИНЛАЙН SVG В HTML

- Полный контроль над SVG, так как это часть DOM.
- «Замусоривает» разметку (решается шаблонизацией).
- Дубликаты утяжеляют страницу (решается с помощью спрайта и gzip на сервере).

```
<html>
<body>
  <svg style="fill: red;">
    <circle r="30" cx="75" cy="75" stroke-width="2" stroke="yellow"></circle>
    ...
    <circle r="30" cx="75" cy="75" stroke-width="2" stroke="red"></circle>
    ...
    <circle r="30" cx="75" cy="75" stroke-width="2" stroke="blue"></circle>
  </svg>
</body>
</html>
```

# ВАРИАНТЫ ИСПОЛЬЗОВАНИЯ (РЕАЛИЗАЦИИ)

---

## ИНЛАЙН SVG В <IMG>

- Минимизирует количество запросов к серверу.
- «Замусоривает» разметку (решается шаблонизацией).
- Дубликаты утяжеляют страницу (решается gzip на сервере).

```
<html>
  <body>
    <img src='data:image/svg+xml;utf8,<svg>...</svg>'>
  </body>
</html>
```

# ВАРИАНТЫ ИСПОЛЬЗОВАНИЯ (РЕАЛИЗАЦИИ)

---

## ИНЛАЙН SVG В CSS

- Кэшируется вместе со стилевым файлом — стили и картинки в одном месте.
- Ухудшает читаемость кода.

```
.background {  
  background-image: url('data:image/svg+xml;utf8,<svg>...</svg>');  
}
```

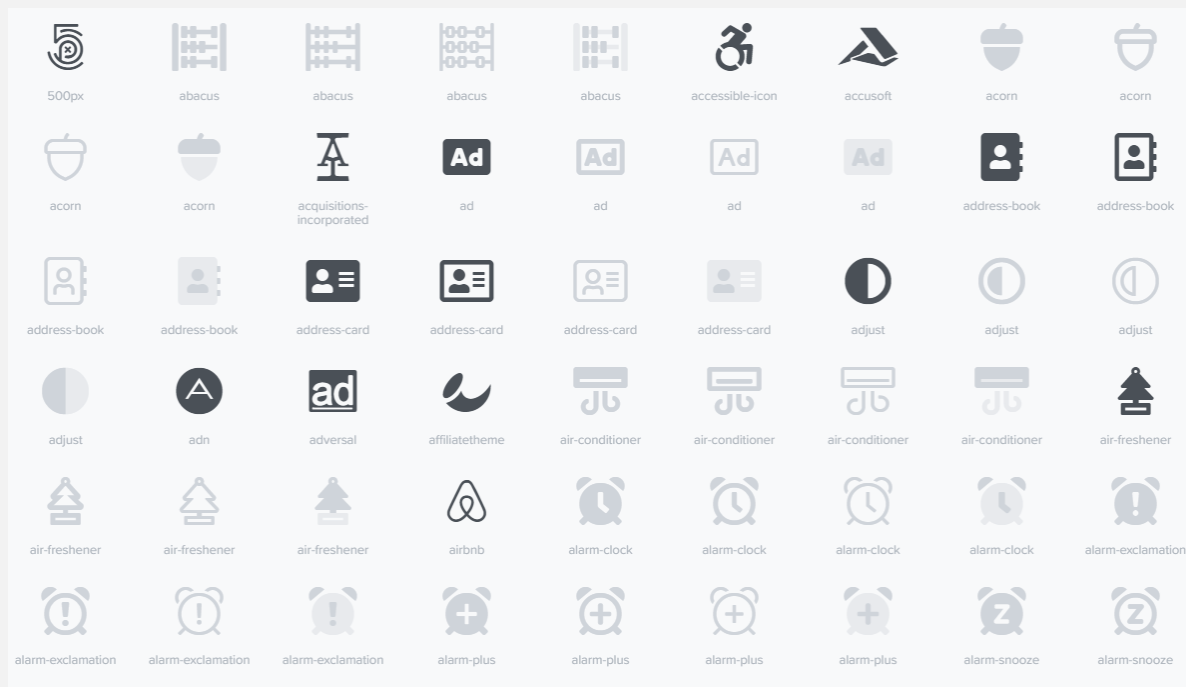
SVG-код должен быть закодирован <http://yoksel.github.io/url-encoder/>.

# ПОДКЛЮЧЕНИЕ SVG

---

- Подключение извне:
  - ✓ `<img>`: для контентных картинок;
  - ✓ фон CSS: для оформительских картинок;
- `object/embed/iframe`: для интерактивных SVG.
- Встраивание внутрь HTML:
  - ✓ инлайн `<svg>` в HTML: SVG — часть DOM, полный контроль из CSS и JS;
- инлайн в `<img>`: для минимизации запросов к серверу;
- инлайн в фон CSS: все картинки встроены в CSS и кешируются.

# СПРАЙТЫ



# ЧТО ТАКОЕ СПРАЙТЫ?

---

Изначально спрайтами называли **независимые объекты**, которые аппаратно **объединяются** с другими объектами.



# ЧТО ТАКОЕ СПРАЙТЫ?

---

У SVG есть несколько элементов-контейнеров – `<g>`, `<defs>` и `<symbol>`, лучшим для спрайтов является `<symbol>` т.к. он поддерживает все атрибуты как у `<svg>`.

Итак, нужно создать файл `icons.svg`, в нём будет общий контейнер `<svg>...</svg>`, далее в него помещаются изображения, с заменой `<svg>` на `<symbol>` оставляя все другие параметры и добавляется атрибут `id` для дальнейшего вывода.

# ЧТО ТАКОЕ СПРАЙТЫ?

```
▼<svg width="0" height="0" style="position:absolute"> == $0
  ▶<symbol viewBox="0 0 56 56" id="svg-product-abn" xmlns="http://www.w3.org/2000/svg">...</symbol>
  ▶<symbol id="svg-product-affiliate" viewBox="0 0 56 56" xmlns="http://www.w3.org/2000/svg">...</symbol>
  ▶<symbol viewBox="0 0 56 56" id="svg-product-assign-ip" xmlns="http://www.w3.org/2000/svg">...</symbol>
  ▶<symbol viewBox="0 0 56 56" id="svg-product-ato" xmlns="http://www.w3.org/2000/svg">...</symbol>
  ▶<symbol id="svg-product-backorder-domains" viewBox="0 0 56 56" xmlns="http://www.w3.org/2000/svg">...</symbol>
  ▶<symbol id="svg-product-bandwidth" viewBox="0 0 56 56" xmlns="http://www.w3.org/2000/svg">...</symbol>
  ▶<symbol id="svg-product-blackberry" viewBox="0 0 56 56" xmlns="http://www.w3.org/2000/svg">...</symbol>
  ▶<symbol id="svg-product-blog" viewBox="0 0 56 56" xmlns="http://www.w3.org/2000/svg">...</symbol>
  ▶<symbol id="svg-product-brand-removal" viewBox="0 0 56 56" xmlns="http://www.w3.org/2000/svg">...</symbol>
  ▶<symbol id="svg-product-business-directory" viewBox="0 0 56 56" xmlns="http://www.w3.org/2000/svg">...</symbol>
  ▶<symbol id="svg-product-business-registration" viewBox="0 0 56 56" xmlns="http://www.w3.org/2000/svg">...</symbol>
  ▶<symbol id="svg-product-catalog" viewBox="0 0 56 56" xmlns="http://www.w3.org/2000/svg">...</symbol>
  ▶<symbol id="svg-product-cloud-backup" viewBox="0 0 56 56" xmlns="http://www.w3.org/2000/svg">...</symbol>
  ▶<symbol viewBox="0 0 42 42" id="svg-product-cloud-network" xmlns="http://www.w3.org/2000/svg">...</symbol>
  ▶<symbol id="svg-product-connected" viewBox="0 0 56 56" xmlns="http://www.w3.org/2000/svg">...</symbol>
  ▶<symbol id="svg-product-contact" viewBox="0 0 56 56" xmlns="http://www.w3.org/2000/svg">...</symbol>
  ▶<symbol id="svg-product-contacts" viewBox="0 0 56 56" xmlns="http://www.w3.org/2000/svg">...</symbol>
```

# ЧТО ТАКОЕ СПРАЙТЫ?

---

```
<span class="svg-product-domain-register">  
  <svg><use xlink:href="#svg-product-domain-register"></use></svg>  
</span>
```

# ОФОРМЛЕНИЕ SVG

---

- ✓ fill, stroke и подобные атрибуты фигурам внутри SVG;
- тег <style> внутри SVG, стилизующий внутренние элементы символа;
- ✓ CSS извне SVG, стилизующий внутренние элементы символа;
- ✓ наследование значения атрибутов fill, stroke от <svg> с <use>;
- ✓ наследование цвета текста родительского контейнера через `currentColor`.

# КАКИЕ SVG ВЫНОСИТЬ В СПРАЙТ?

---

- Всё, что не `<img>` и не фон.
- То есть все SVG, которые инлайнятся в HTML

# SVG АНИМАЦИЯ

---

- Есть два основных способа анимации SVG элемента:
- CSS анимация
- SMIL анимация, встроенная в SVG (на самом деле это SVG анимация, которая базируется на SMIL и расширяет его функционал)

# SVG ОПТИМИЗАЦИЯ

---



SVGO - <https://jakearchibald.github.io/svgomg/>

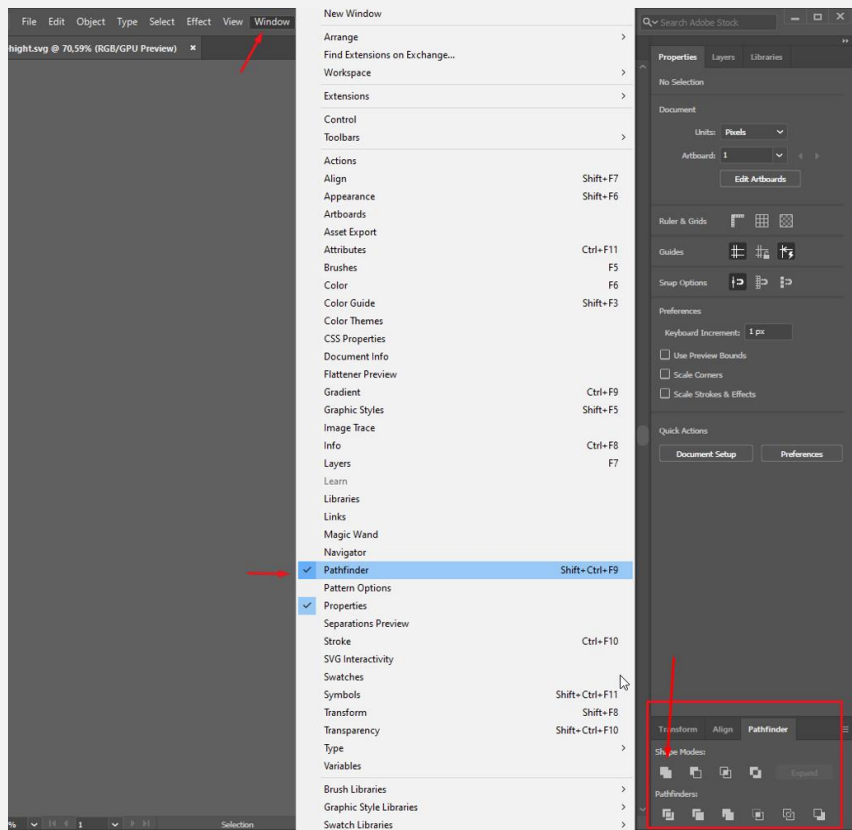
# ADOBE ILLUSTRATOR

---

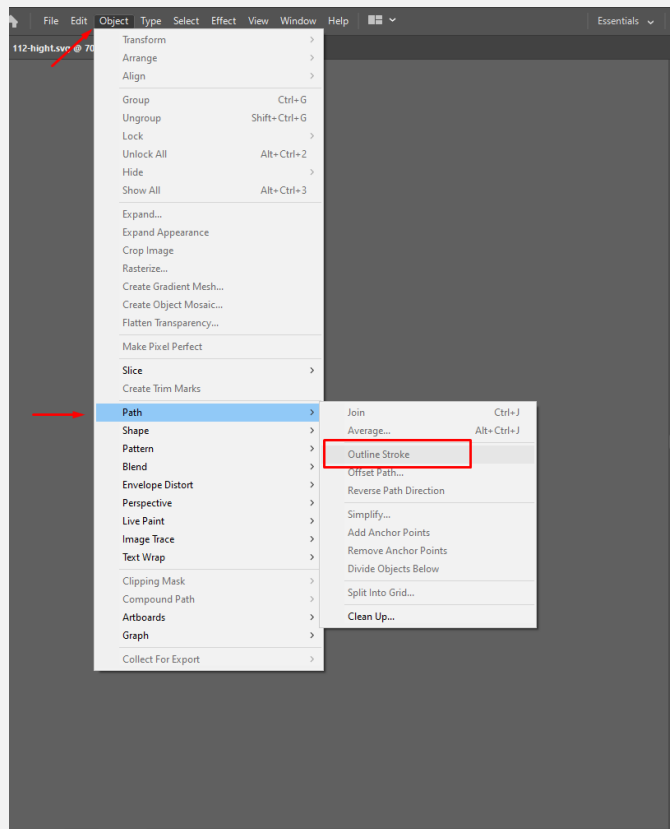




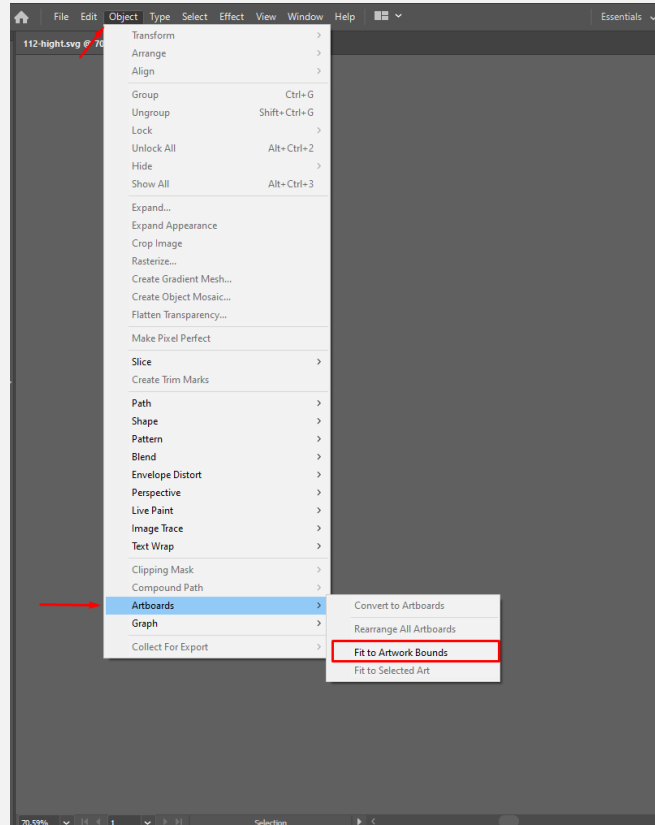
# ОБЪЕДИНЕНИЕ И ЧИСТКА ФОРМ



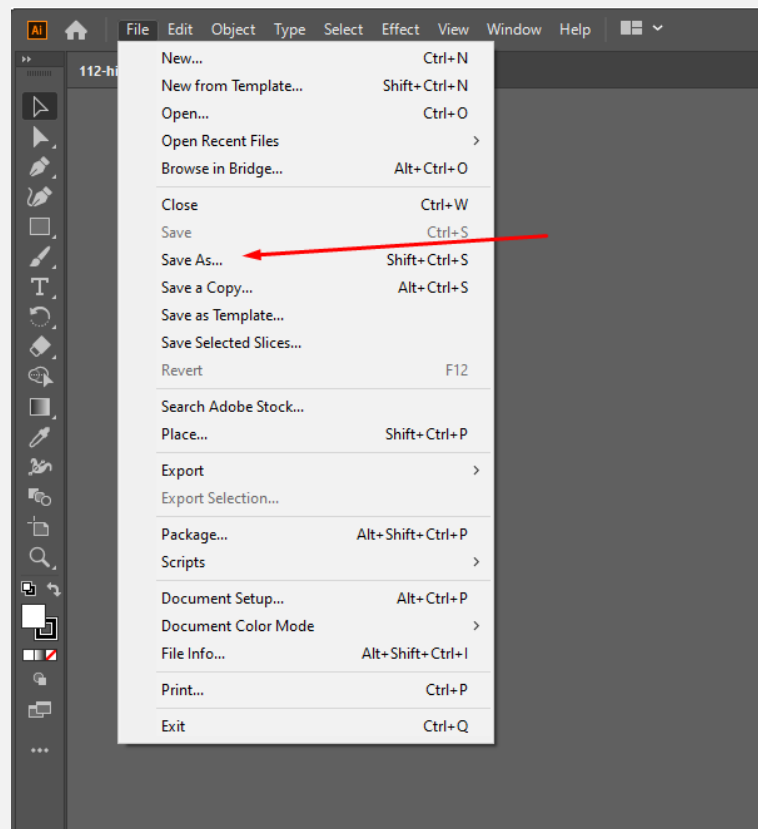
# УПРОЩЕНИЕ СЛОЖНЫХ ФОРМ



# PA3MEP



# СОХРАЩЕНИЕ SVG



# СОХРАЩЕНИЕ SVG

### SVG Options

SVG Profiles: SVG 1.1

Fonts

Type: SVG

Subsetting: None (Use System Fonts)

Options

Image Location: Link

☐ Preserve Illustrator Editing Capabilities

Advanced Options

CSS Properties: Style Elements

☐ Include Unused Graphic Styles

Decimal Places: 1 Encoding: Unicode (UTF-8)

☒ Output fewer <span> elements ☐ Include Slicing Data

☒ Use <textPath> element for Text on Path ☐ Include XMP

☒ Responsive

Description

Hold the cursor over a setting for additional information.

[Less Options](#) [SVG Code...](#) [OK](#) [Cancel](#)