

# Шрифты и текст

---

# Единицы измерения

---

Пиксель px – это самая базовая, абсолютная и окончательная единица измерения.

Существуют также «производные» от пикселя единицы измерения: mm, cm, pt и pc.

Вот их значения:

1mm (мм) = 3.8px

1cm (см) = 38px

1pt (типографский пункт) =  $4/3$  px

1pc (типографская пика) = 16px

Так как браузер пересчитывает эти значения в пиксели, то смысла в их употреблении нет.

# Относительно шрифта: em

---

1em – текущий размер шрифта. Размеры в em – относительные, они определяются по текущему контексту.

```
1 <div style="font-size:24px">  
2   Страусы  
3   <div style="font-size:24px">Живут также в Африке</div>  
4 </div>
```

---

Страусы  
Живут также в Африке

---

# Относительно шрифта: em

---

1em – текущий размер шрифта. Размеры в em – относительные, они определяются по текущему контексту.

```
1 <div style="font-size:24px">
2   Страусы
3   <div style="font-size:24px">Живут также в Африке</div>
4 </div>
```

Страусы  
Живут также в Африке

```
1 <div style="font-size:1.5em">
2   Страусы
3   <div style="font-size:1.5em">Живут также в Африке</div>
4 </div>
```

Страусы  
Живут также в Африке

# Проценты %

---

Проценты %, как и em – относительные единицы. Как правило, процент будет от значения свойства родителя с тем же названием, но не всегда.

```
1 <div style="font-size:150%">
2   Страусы
3   <div style="font-size:150%">Живут также в Африке</div>
4 </div>
```

Страусы

Живут также в Африке

---

# Единица rem: смесь px и em

---

Итак, мы рассмотрели:

- px – абсолютные, чёткие, понятные, не зависящие ни от чего.
- em – относительно размера шрифта.
- % – относительно такого же свойства родителя

(а может и не родителя, а может и не такого же – см. примеры выше).

```
1 <ul>
2 <li>Собака
3   <ul>
4     <li>бывает
5       <ul>
6         <li>кусачей
7           <ul>
8             <li>только
9               <ul>
10                <li>от жизни
11                  <ul>
12                    <li>собачей</li>
13                  </ul>
14                </li>
15              </ul>
16            </li>
17          </ul>
18        </li>
19      </ul>
20    </li>
21  </ul>
22 </li>
23 </ul>
```

- Собака
  - бывает
    - кусачей
      - только
        - от жизни
          - собачей

# Единица rem: смесь px и em

---

Теперь уменьшим размер шрифта до 0.8em, вот что получится:

```
1 <style>
2   li {
3     font-size: 0.8em;
4   }
5 </style>
6
```

- Собака
  - бывает
    - кусачей
    - палач
    - отжиги
    - собачей



# Единица rem: смесь px и em

---

Единица rem задаёт размер относительно размера шрифта элемента <html>.

```
html {  
  font-size: 14px;  
}  
li {  
  font-size: 0.8rem;  
}
```

- Собака
  - бывает
    - кусачей
      - только
        - от жизни
          - собачей

# Шрифты

---

Шрифт — графический рисунок начертаний букв и знаков, составляющих единую стилистическую и композиционную систему, набор символов определенного размера и рисунка.



# Векторные шрифты

---

Представляют собой набор описаний символов с помощью математических формул.



Как и любое векторное изображение, векторные шрифты можно масштабировать без потери качества.

# Типографика

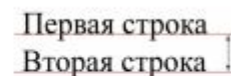
---

A diagram showing a horizontal line (the baseline) passing through the bottom of the letters 'H' and 'g'. The 'H' is a simple block letter, while the 'g' has a descender that goes below the baseline.

Базовая линия

A diagram showing two letters, 'x' and 'H', positioned above a horizontal baseline. A vertical double-headed arrow is drawn to the left of the letters, indicating the height from the baseline to the top of the 'x' and 'H'.

Высота знаков

A diagram showing two lines of text. The top line is labeled 'Первая строка' and the bottom line is labeled 'Вторая строка'. A vertical double-headed arrow is drawn between the two lines, indicating the distance between them.

Интерлиньяж

- **Базовая линия** — воображаемая прямая линия, проходящая по нижнему краю прямых знаков без учёта свисаний и нижних выносных элементов.
- **Высота прописных знаков** — расстояние от базовой линии до верхней линии прописных, то есть высота прописных букв без учета свисаний.
- **Высота строчных знаков** — расстояние от базовой линии до верхней линии строчных, то есть высота строчных букв без свисаний и выносных элементов.
- **Интерлиньяж (межстрочный интервал)** — расстояние между базовыми линиями строк.

# Кегельная площадка

---

**Кегельная площадка** — верхняя прямоугольная часть ножки литеры, на которой расположено выпуклое (печатающее) изображение знака. В цифровом шрифте кегельная площадка важна при проектировании шрифта и представляет собой прямоугольник, в который вписывается изображение знака.



# Размер шрифта

---

Размер шрифта определяется как высота от базовой линии до верхней границы кегельной площадки. Размер шрифта элемента определяется с помощью свойства `font-size`. Размер шрифта можно установить с помощью **ключевых слов** или **единиц измерения длины**.

# Ключевые слова

---

Первый набор — ключевые слова, задающие абсолютные значения:

xx-small, x-small, small, medium, large, x-large, xx-large

<b>CSS</b>	xx-small	x-small	small	medium	large	x-large	xx-large	
<b>HTML</b>	1		2	3	4	5	6	7

По сути такие ключевые слова являются прямыми наследниками `<font size = "X">`  
`</font>` с небольшим сдвигом.



# Ключевые слова

---

Второй набор ключевых слов задают относительные значения: `smaller` и `larger`. Они изменяют размер шрифта относительно размера шрифта родителя примерно на 20%.

```
1  mark {  
2      font-size: larger;  
3  }  
4  
5  strong {  
6      font-size: smaller;  
7  }
```

# Единицы измерения длины

---

- Пиксель

```
1  p {  
2    font-size: 10px;  
3  }
```

- em, %

```
1  div {  
2    font-size: .6em; /* то же, что и 60% */  
3    color: #0b0;  
4  }
```

- rem

```
1  html {  
2    font-size: 16px;  
3  }  
4  
5  h2 {  
6    font-size: 60px;  
7  }  
8  
9  div {  
10   font-size: 2rem;  
11 }
```

# Высота строки

---

Высота строки устанавливается свойством `line-height` двумя типами значений: единицами измерения длины или множителем.

```
1  body { font-size: 20px; }
2
3  .line-height-mult { line-height: 1.5; }
4  .line-height-em   { line-height: 1.5em; }
5  .line-height-pixel { line-height: 30px; }
```

```
1  <div class="line-height-mult">
2    <!-- Высота строки множителем 1.5 x 20 = 30 -->
3    <p>...</p>
4  </div>
5
6  <div class="line-height-em">
7    <!-- Высота строки в em 1.5 x 20 = 30 -->
8    <p>...</p>
9  </div>
10
11 <div class="line-height-pixel">
12   <!-- Высота в пикселях 30 = 30 -->
13   <p>...</p>
14 </div>
```

Множитель использует размер шрифта элемента, к которому применено свойство, а не родительского элемента, как в случае с `em`.

# Расстояние между буквами

---

Для изменения расстояния между буквами предусмотрено свойство `letter-spacing` со значением в любой единице измерения длины кроме процентов.

```
1   .zero {
2       letter-spacing: normal;
3   }
4
5   .first {
6       letter-spacing: -2.5px;
7   }
8
9   .second {
10      letter-spacing: 1em;
11  }
```

# Расстояние между словами

---

Аналогичным свойством является свойство `word-spacing`, которое устанавливает расстояние между словами. Значение может быть в любой единице измерения длины кроме процентов.

```
1  .zero {
2      word-spacing: normal;
3  }
4
5  .first {
6      word-spacing: -10px;
7  }
8
9  .second {
10     word-spacing: 1em;
11 }
```

# Начертание

---

**Капитель** — начертание, в котором строчные знаки выглядят как уменьшенные прописные. Устанавливается с помощью свойства `font-variant`, у которого есть два значения: `normal` и `small-caps`.

```
1 .h3 {  
2     font-variant: normal;  
3 }
```

```
1 .h3 {  
2     font-variant:  
3     ↪ small-caps;  
4 }
```

# Характеристики рисунка: наклон, насыщенность и ПЛОТНОСТЬ

---

- Свойство font-style

```
1 font-style:  
  ↳ normal; /*  
  ↳ a */  
2 font-style:  
  ↳ oblique; /*  
  ↳ б */  
3 font-style:  
  ↳ italic; /*  
  ↳ в */
```



п п п

а б в

а — прямые; б — наклонные;  
в — курсивные

# Характеристики рисунка: наклон, насыщенность и ПЛОТНОСТЬ

---

- Свойство font-weight

```
1  font-weight:  
   → normal /*  
   → обычный (a)  
   → */  
2  bold /*  
   → жирный  
   → (в) */  
3  light /*  
   → тонкий  
   → */
```



а — светлые; б — полужирные;  
в — жирные



# Характеристики рисунка: наклон, насыщенность и ПЛОТНОСТЬ

---

Также можно установить значение числом от 100 до 900 с шагом 100:

```
1 font-weight: 100 /* thin */
2 font-weight: 200 /* extra light */
3 font-weight: 300 /* light */
4 font-weight: 400 /* normal */
5 font-weight: 500 /* medium */
6 font-weight: 600 /* semi-bold (6) */
7 font-weight: 700 /* bold */
8 font-weight: 800 /* extra bold */
9 font-weight: 900 /* black */
```

# Характеристики рисунка: наклон, насыщенность и ПЛОТНОСТЬ

---

Также можно указать относительное значение.

```
1   font-weight: lighter /* тоньше шрифта родителя на ~300 */  
2   font-weight: bolder /* жирнее шрифта родителя на ~300 */
```

# Характеристики рисунка: наклон, насыщенность и ПЛОТНОСТЬ

---

- Свойство font-stretch

```
1 font-stretch:  
  ↳ ultra-condensed;  
2 font-stretch:  
  ↳ extra-condensed;  
3 font-stretch:  
  ↳ condensed;  
4 font-stretch:  
  ↳ semi-condens  
5 font-stretch:   П   П   П   а — узкие; б — нормальные;  
  ↳ normal;      ↳   ↳   ↳   в — широкие  
6 font-stretch:  
  ↳ semi-expanded;  
7 font-stretch:  
  ↳ expanded;  
8 font-stretch:  
  ↳ extra-expanded;  
9 font-stretch:  
  ↳ ultra-expanded;
```

# Семейства шрифтов

---

- `serif` — шрифты с засечками (антиква)
- `sans-serif` — шрифты без засечек (гротески)
- `cursive` — рукописные шрифты (каллиграфические)
- `monospace` — моноширинные шрифты (непропорциональные)
- `fantasy` — декоративные шрифты (акцидентные)

# Семейства шрифтов

---

Существует ряд шрифтов, которые есть практически в каждой ОС, таким образом их использование считается наиболее безопасным. Самые известные из них:

- Без засечек: Arial
- С засечками: Times New Roman
- Моноширинный: Courier New

# Семейства шрифтов

---

Чтобы использовать нужный шрифт, нужно указать свойство `font-family` с названием нужного шрифта:

```
1  body {  
2      font-family: 'Arial';  
3  }
```

Можно писать название шрифта без кавычек:

```
1  body {  
2      font-family: Arial;  
3  }
```

Также можно писать несколько шрифтов через запятую:

```
1  body {  
2      font-family: Arial, Helvetica, Calibri;  
3  }
```

# Семейства шрифтов

---

Чтобы указать браузеру, что нужно установить шрифт по умолчанию нужного семейства, можно в качестве значений использовать:

```
1 font-family: serif; /* с засечками */
2 font-family: sans-serif; /* без засечек */
3 font-family: cursive; /* рукописные */
4 font-family: monospace; /* моноширинные */
5 font-family: fantasy; /* декоративные */
```

# Нестандартные шрифты

---

Всегда есть возможность подключить свой шрифт на страницу с помощью at правила `@font-face`.

```
1  @font-face {  
2      font-family: myCustomFont;  
3      src: url(myCustomFont.woff);  
4  }
```

После этого можно написать:

```
1  body {  
2      font-family: myCustomFont, sans-serif;  
3  }
```



# Нестандартные шрифты

---

Если есть вероятность, что какой-либо шрифт уже установлен на пользовательский компьютер, можно указать его через `local()`:

```
1  @font-face {
2      font-family: myCustomFont;
3      src: local('Native Custom Font')
4          url(myCustomFont.woff);
5  }
```

# Нестандартные шрифты

---

Подключенный шрифт жирного начертания и, например, курсивного начертания – два разных файла, и подключать их нужно отдельно.

```
1  @font-face {
2      font-family: Sooo Good Font Bold; /* жирный */
3      src: url(soGoodFontBold.woff);
4  }
5
6  @font-face {
7      font-family: Sooo Good Font Italic; /* курсивный */
8      src: url(soGoodFontItalic.woff);
9  }
```

# Нестандартные шрифты

---

В таком случае приходится запоминать название шрифта вместе с начертанием.

```
1  header {
2      font-family: Sooo Good Font Bold, sans-serif; /* жирный */
3  }
4
5  footer {
6      font-family: Sooo Good Font Italic, sans-serif; /* курсивный
7      → */
8  }
```

# Нестандартные шрифты

---

Но можно привязывать определенный файл к определенному начертанию с помощью свойств `font-weight` и `font-style`, указанных в `@font-face`:

```
1  @font-face {
2      font-family: Sooo Good Font;
3      src: url(soGoodFont.woff);
4      font-style: normal; /* обычный */
5  }
6
7  @font-face {
8      font-family: Sooo Good Font;
9      src: url(soGoodFontItalic.woff);
10     font-style: italic; /* курсивный */
11 }
```

```
1  body {
2      font-family: Sooo Good Font; /* обычный */
3  }
4
5  footer {
6      /* font-family унаследуется от body */
7      font-style: italic; /* курсивный */
8  }
```

# Свойство font

---

Свойство `font` — универсальное свойство, которое позволяет контролировать каждое свойство, связанное с шрифтами.

```
1   font: variant style weight size / line-height family;

1   body {
2     font: 14px;           /* не будет работать */
3     font: italic Arial;  /* тоже не будет работать */
4
5     font: 14px Arial, sans-serif;
6     font: 14px / 1.2 Arial, sans-serif;
7     font: italic bold 14px Times New Roman;
8     font: small-caps italic 700 14px / 20px Arial, sans-serif;
9   }
```

# Текст

---

Выравнивание текста осуществляется с помощью свойства `text-align`:

```
1  .uno   { text-align: left;   } /* По левому краю */
2  .dos   { text-align: start;  } /* По краю начала текста */
3  .tres  { text-align: right;  } /* По правому краю */
4  .quatro { text-align: end;   } /* По краю, противоположенному
   ↳ началу текста */
5  .cinco { text-align: center; } /* По центру */
6  .seis  { text-align: justify; } /* По всей ширине */
```

Свойство `text-decoration`:

- `overline` (надчеркивание)
- `line-through` (зачеркивание)
- `underline` (подчеркивание)

# Текст

---

Свойство text-transform:

- capitalize
- lowercase
- uppercase

```
1  .eins {  
2      text-indent: 5%;  
3  }  
4  
5  .zwei {  
6      text-indent: -1em;  
7  }
```

# Цвет текста

---

Цвет текста задается с помощью свойства `color`

```
1  h1 {  
2      color: #f00;  
3  }
```

- Свойство `color` устанавливает так называемый «цвет переднего плана» (в противовес `background-color` – цвет фона), поэтому он имеет влияние на многие свойства

```
1  p {  
2      color: #0f0;           /* зеленый цвет текста */  
3      border: 5px solid;     /* рамка будет зеленой */  
4      box-shadow: 0 0 50px;  /* и тень элемента */  
5      text-shadow: 4px 4px 10px; /* и тень текста */  
6  }
```

- Свойства, которые не наследуют цвет от `color` автоматически, могут сделать это при помощи ключевого слова `currentColor`.



# Цвет текста

---

СВОЙСТВО text-shadow:

```
1 text-shadow: offset-x offset-y blur-radius color
```

```
1 h1 {  
2   text-shadow: 4px 4px;  
3 }
```

Текст для примера тени

```
1 h1 {  
2   text-shadow: 4px 4px #f00;  
3 }
```

Текст для примера тени

```
1 h1 {  
2   text-shadow: 4px 4px 10px #f00;  
3 }
```

Текст для примера тени

```
1 h1 {  
2   text-shadow: 4px 4px 10px #f00,  
3                -4px -4px 10px #0f0;  
4 }
```

Текст для примера тени

# Многоколоночность

---

```
1 <div>
2     В европейских языках чтение текста происходит слева направо,
3     в то время как есть языки, где текст читается справа
4     налево. При смешении в одном документе разных
5     по написанию символов в системе юникод,
6     их направление определяется браузером из характеристик
7     и содержимого текста.
8 </div>
```

Нужное количество колонок можно указать с помощью `column-count`:

```
1   div {
2     column-count: 2;
3   }
```

Расстояние между колонками можно задать, используя `column-gap`:

```
1   div {
2     column-count: 2;
3     column-gap: 300px;
4   }
```

# Многоколоночность

---

С помощью `column-rule` можно добавить разделитель между колонками, задав ширину, тип и размер разделителя:

```
1   div {  
2     column-count: 2;  
3     column-rule: 2px solid #0f0;  
4   }
```

С помощью `column-width` можно указать ширину колонок:

```
1   div {  
2     column-width: 200px;  
3     column-rule: 2px solid #fff;  
4   }
```

# Переполнение текста

---

Пусть дана следующая таблица.

Name	Surname	Points
Cloud	Strife	13
John	Cena	37
Akakiy	Stanislav	666
Sokolovskiy	Ruslan	2

В данном случае переполнение контента следует скрыть с помощью `overflow: hidden`.

Name	Surname	Points
Cloud	Strife	13
John	Cena	37
Akakiy	Stanislav	666
Sokolovskiy	Ruslan	2

# Переполнение текста

---

Свойство `text-overflow` имеет два значения: `clip` (значение по умолчанию) и `ellipsis`.

Name	Surname	Points
Cloud	Strife	77
Vladimir	Putin	147
Akakiy	Stanisl...	666
Sokolo...	Ruslan	2

# Переносы внутри слов

---

Чтобы использовать автопереносы, нужно указать свойство `lang` для параграфа, а также установить автопереносы с помощью свойства `hyphens`.

```
1 <p lang="ru">
2     Далеко-далеко за словесными горами в стране
3     гласных и согласных букв живут рыбные тексты.
4     Вдали от всех живут они в буквенных домах на берегу
5     Семантика большого языкового океана.
6 </p>
```

```
1 p {
2     hyphens: auto;
3 }
```

# Мягкие переносы

Переносы без дефиса устанавливаются с помощью `<wbr>`:

```
1 <p>
2   Да<wbr>ле<wbr>ко-да<wbr>ле<wbr>ко за
   → сло<wbr>вес<wbr>ны<wbr>ми
3   го<wbr>ра<wbr>ми в стра<wbr>не глас<wbr>ных и со<wbr>
   → глас<wbr>ных букв
4   жи<wbr>вут рыб<wbr>ные тексты. Вда<wbr>ли от всех жи<wbr>вут
   → они в
5   бук<wbr> вен<wbr>ных до<wbr>мах на бе<wbr>ре<wbr>гу
   → Се<wbr>ман<wbr>ти<wbr>ка
6   боль<wbr>шо<wbr>го язы<wbr>ко<wbr>во<wbr>го оке<wbr>ана.
7 </p>
```

Если слово разрывать можно в любом месте, то это можно указать, используя свойство `word-break` со значе

```
1 .normal {
2   word-break: normal;
3 }
4
5 .breakall {
6   word-break: break-all;
7 }
```



# Мягкие переносы

---

Существуют ситуации, когда слово или словосочетание не нужно разрывать ни при каких обстоятельствах.

- 1 Тут нужно что-то написать, чтобы текст сдвинул `mark` в неудобное  
→ положение.
- 2 Правило `<mark>hyphens: auto;</mark>` использует примерно никто,  
3 ибо пока оно не работает нормально

Перенос нужно запретить

```
1  mark {  
2    white-space: nowrap;  
3  }
```

```
1 <div>
2   We ache to be transparent
3     We run from the "open" arms;
4       The facade of something greater than ourselves
5         ↳ <span>(здесь просто дополнительный текст, чтобы
6           ↳ не помещался в строку)</span>
7           And we're left to coexist with infestation
8
9     Our history is cursed
10       Through the past, present, and future
11       If they're created in his image
12       Then his image is disgusting...
13 And even he can't wipe you clean
</div>
```

We ache to be transparent We run from the "open" arms; The facade of something greater than ourselves (здесь просто дополнительный текст, чтобы не помещался в строку) And we're left to coexist with infestation Our history is cursed Through the past, present, and future If they're created in his image Then his image is disgusting... And even he can't wipe you clean

EXAMPLE

## Если же указать

```
1  p {  
2    white-space: pre;  
3  }
```

We ache to be transparent  
We run from the "open" arms;  
The facade of something greater than ourselves (здесь просто дополнительно)  
And we're left to coexist with infestation

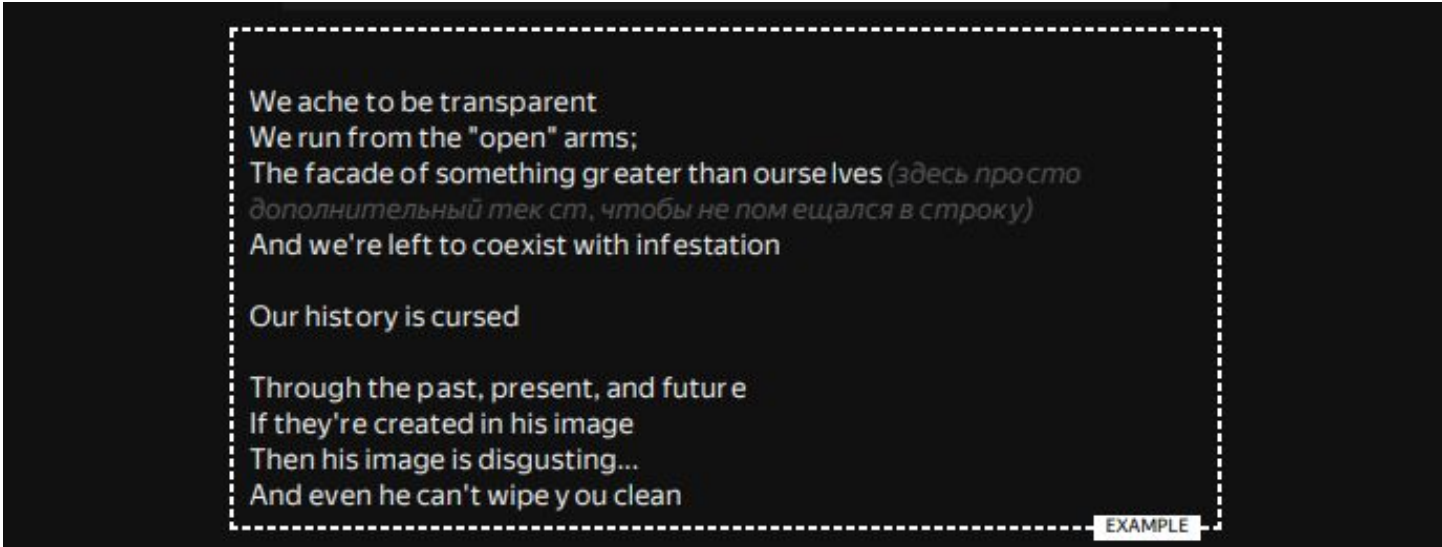
Our history is cursed

Through the past, present, and future  
If they're created in his image  
Then his image is disgusting...  
And even he can't wipe you clean

EXAMPLE

Если установить значение pre-line,

```
1  p {  
2    white-space: pre-line;  
3  }
```



We ache to be transparent  
We run from the "open" arms;  
The facade of something gr eater than ourse lves (*здесь просто  
дополнительный тек ст, чтобы не пом ещался в строку*)  
And we're left to coexist with infestation

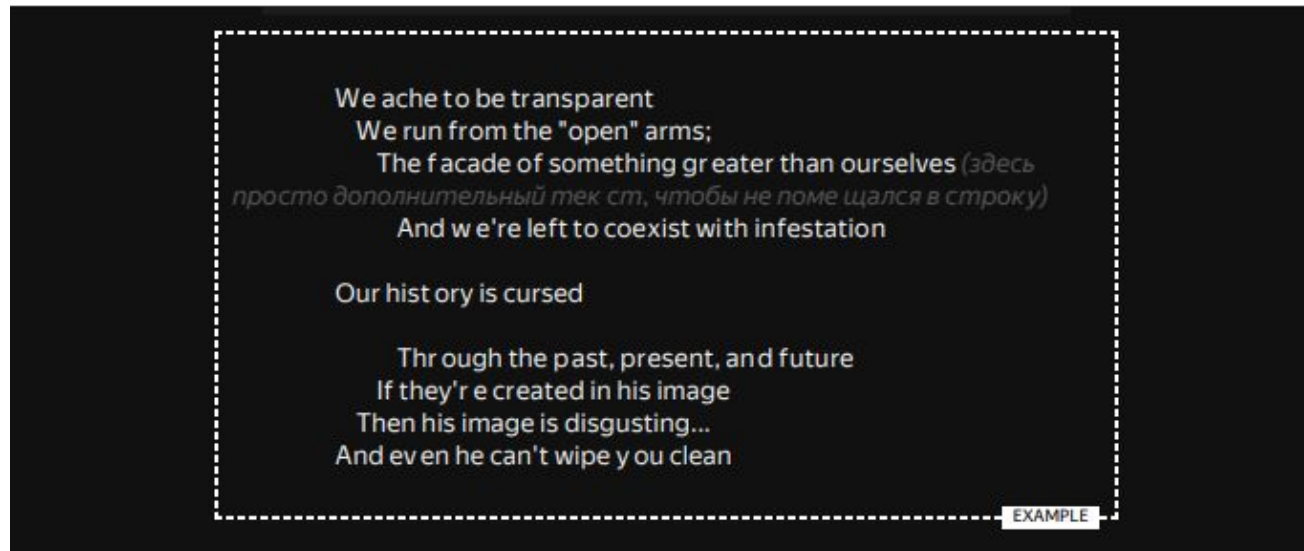
Our history is cursed

Through the past, present, and future  
If they're created in his image  
Then his image is disgusting...  
And even he can't wipe y ou clean

EXAMPLE

Если указать значение pre-wrap,

```
1  p {  
2    white-space: pre-wrap;  
3  }
```



Если установлено значение nowrap

```
1  p {  
2    white-space: nowrap;  
3  }
```

We ache to be transparent We run from the "open" arms; The facade of something greater

EXAMPLE

# Псевдоэлементы для работы с текстом

---

## Псевдоэлемент первой строки

```
1  p::first-line {  
2      font: 25px Courier New;  
3      letter-spacing: 6px;  
4  }
```

```
1  <p>  
2      Текст в первой строке будет выглядеть как китайско-русская  
3      ↳ инструкция  
4      купленного на таганчике барахла. И не спрашивайте откуда  
5      я это знаю и почему вообще показываю как этот шрифт  
6      ↳ имитировать.  
7  </p>
```

Текст в первой строке будет выглядеть как китайско-русская ,инструкция купленного на таганчике барахла. И не спрашивайте откуда я это знаю и почему вообще показываю как этот шрифт , имитировать.

# Псевдоэлементы для работы с текстом

---

## Псевдоэлемент первой буквы

```
1  p::first-letter {  
2      font: italic 50px Times New Roman;  
3      margin: 15px;  
4      padding: 10px;  
5      color: #f00;  
6      background: #0ff;  
7      border: 2px dashed #0f0;  
8  }
```

```
1  <p>  
2      Хочется сделать наиболее вырвиглазный вариант, отвратный прям  
3      ↳ максимально и бесповоротно  
4  </p>  
5  <p>  
6      А еще лучше - в два параграфа, чтобы вдвойне вырвиглазнее  
7  </p>
```



очется сделать наиболее вырвиглазный вариант, отвратный прям,максимально и бесповоротно

еще лучше – в два параграфа, чтобы вдвойне вырвиглазнее