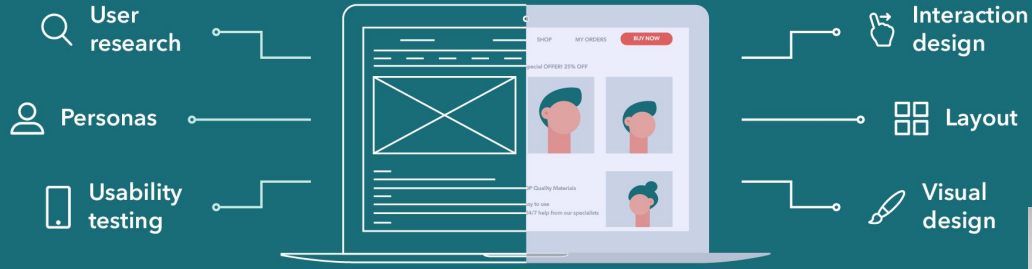


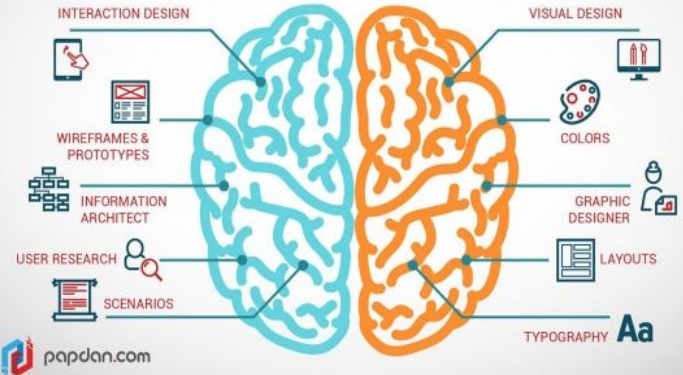
Тестирование UI (/ux)

Сперва поговорим о UI и UX

UX UI



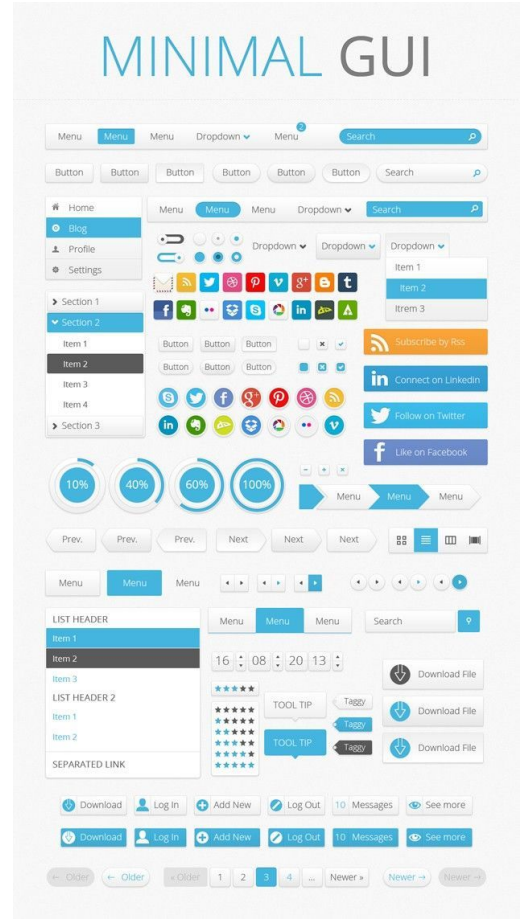
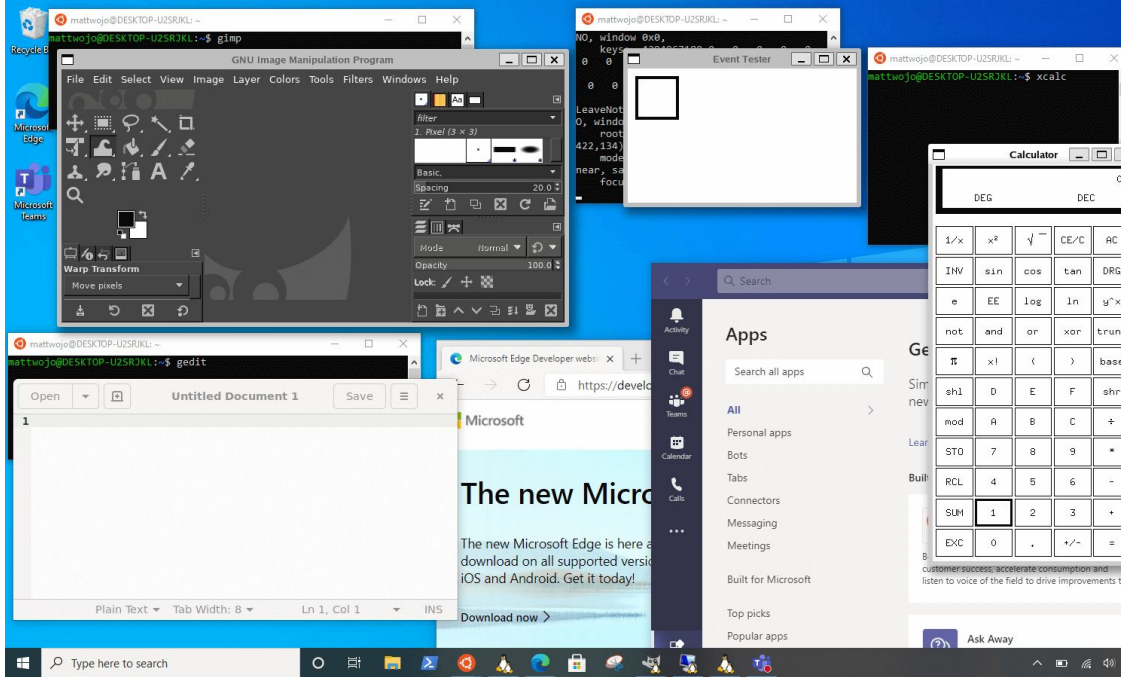
UX & UI



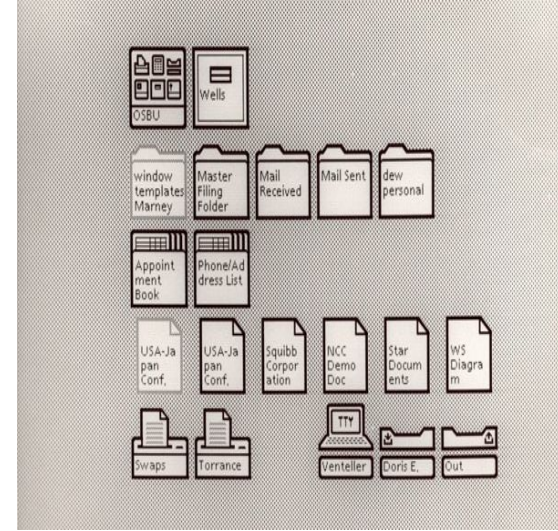
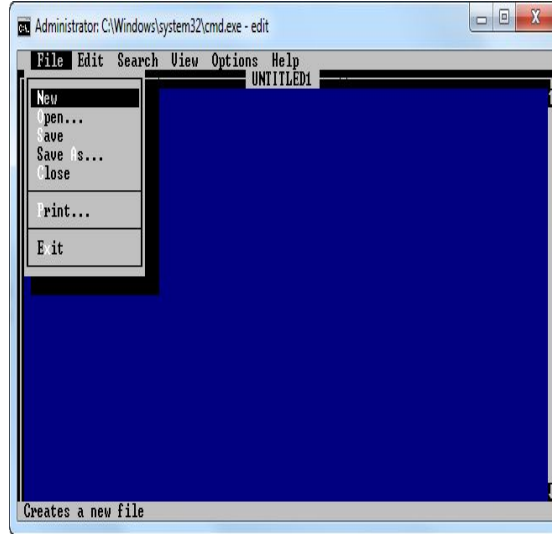
Какие же критерии у UX

- эффективность (efficiency)
- правильность (accuracy)
- активизация в памяти (recall)
- эмоциональная реакция (emotional response)

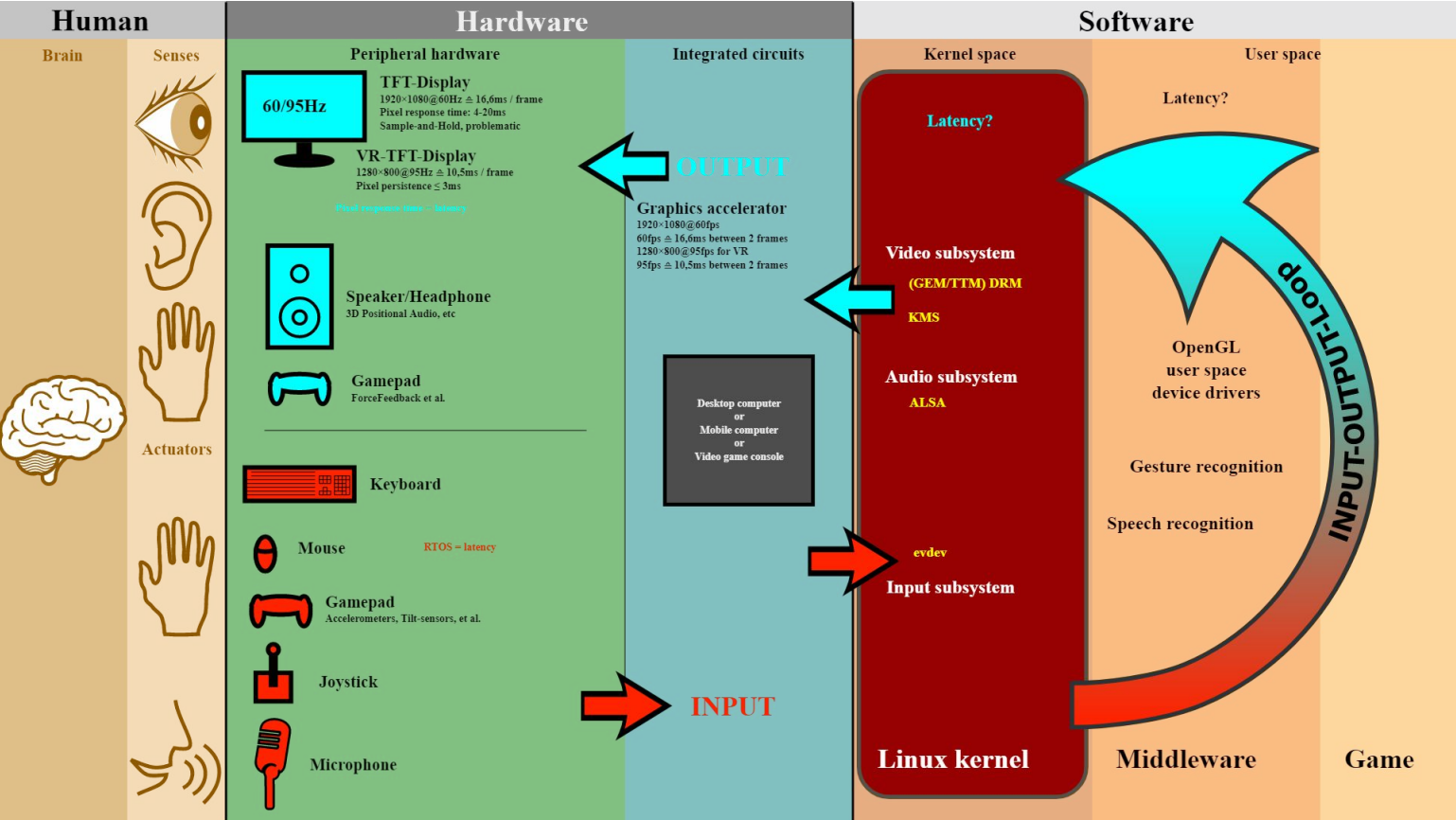
Теперь поговорим о GUI



Сначала были консоли

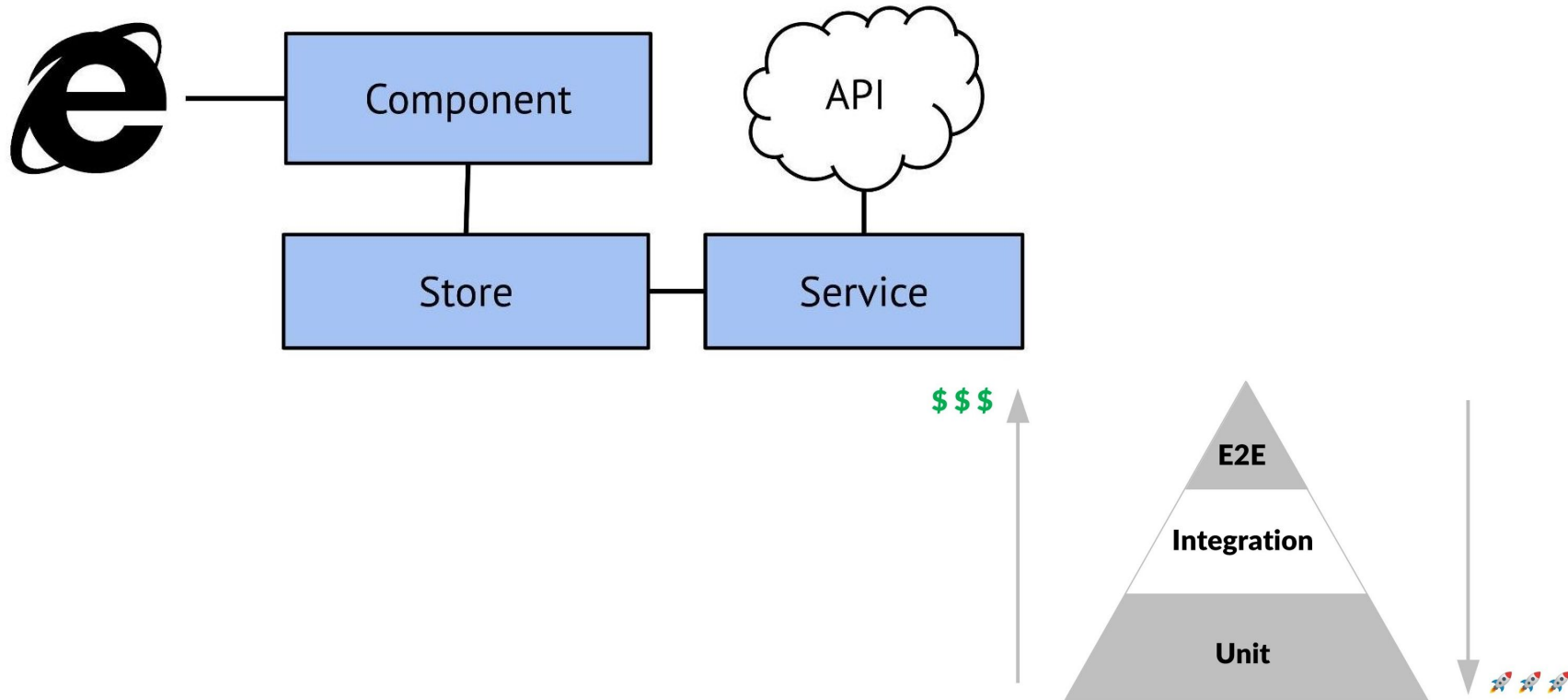


GUI сейчас



WebGui

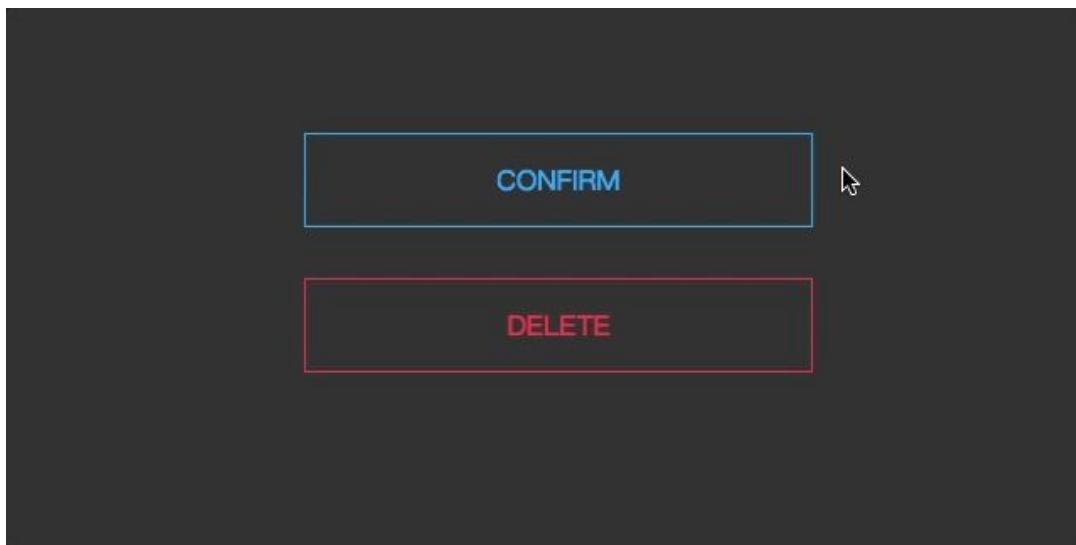
Модули в архитектуре



Функциональные тесты

- Настраиваем окружение
- Рендерим компонент
- Делаем какие-то действия в DOM
- Ждём ререндера
- Проверяем что окружение было вызвано так, как мы ожидали (в нашем случае проверяем вызов API).
- Проверяем, что в DOM-дереве находится контент, который мы ожидали увидеть.

Функциональное или не функциональное??



- Элементы страницы расположены [как на макете](#) на всех устройствах.
- Сайт одинаково выглядит и работает [во всех нужных браузерах](#).
- Кнопки нажимаются и после этого что-то происходит, слайдеры крутятся, гамбургеры раскрываются. (эта часть прежде всего мануальна но можно и автоматизировать)
- Все [JavaScript-скрипты](#) работают корректно. (ну тут на самом деле работа за unit tests)
- Отображается правильный контент. (очень легко сформулировать но сложно проверить)
- Отдаются нужные заголовки. (да это тоже желательно проверить если есть документация)
- [Фавиконка](#) установлена. (ага иногда об этом забывается)
- Текст отображается не кракозябрами (в 2020 такое редко, но бывает).
- Курсор интерактивный на интерактивных элементах и обычный на обычных. (опять же смотрим по ТЗ)
- С локализацией всё в порядке (русская, английская версия).
- Страница не разъезжается, если включить блокировщик рекламы. (опционально)

И отдельно для Веб-форм

- Обязательные поля подписаны.
- Если данные должны быть записаны в базу, проверяем это.
- Выводятся понятные сообщения об ошибках заполнения.
- Проверяем экранирование символов в формах на уровне клиента и сервера.
- Приходят подтверждающие письма (если так задумано).

Небольшой чек-лист

- **Типы данных.** Убедитесь, что для определенных типов данных, таких как валюта и даты, можно вводить только допустимые значения.
- **Ширина поля.** Если определенное текстовое поле предназначено для определенного количества символов, укажите в пользовательском интерфейсе, что введенные данные не должны превышать границу по количеству символов. (Например, поле, которое позволяет использовать 50 символов в базе данных приложения, не должно позволять пользователям вводить более 50 символов в интерфейсе).
- **Элементы навигации.** Убедитесь, что все кнопки навигации на странице работают и перенаправляют пользователей на нужную страницу или экран.

- **Индикаторы прогресса.** Иногда приложению нужно время, чтобы выполнить порученную работу, в таких случаях используйте индикатор прогресса, он поможет понять, что работа все еще выполняется.
- **Подсказки ввода.** В выпадающем меню с сотнями элементов при вводе первой буквы должны остаться только те элементы, которые начинаются с этой буквы, так вы уберете пользователей от просмотра длинной портянки значений.
- **Скролл таблиц.** Когда данные из таблицы перетекают на следующую страницу, функция прокрутки должна позволять пользователям прокручивать данные, но не трогать все заголовки.
- **Ведение журнала ошибок.** Когда в системе возникает фатальная ошибка, убедитесь, что приложение записывает сведения об ошибке в специальный файл журнала для последующего просмотра.
- **Пункты меню и режим.** Убедитесь, что приложение отображает только те пункты меню, которые доступны в определенном режиме.

- **Комбинации клавиш.** Проверьте комбинации клавиш, правильно ли они работают, независимо от браузера, платформы или устройства.
- **Кнопки подтверждения действий.** Убедитесь, что пользовательский интерфейс имеет работающую кнопку подтверждения каждый раз, когда пользователь хочет сохранить или удалить элемент.

Так ну теперь к инструментам

нагрузочные тесты



снимок тесты



TDD дымовые тесты



скриншот-тесты



функциональные тесты

ATDD



BDD



снимок тесты



браузерные тесты



приемочные тесты

Начнем с классики

```
nk version="1.1" fill-rule="evenodd" fill="none" stroke="none" stroke-linecap="square" stroke-miterlimit="10" overflow="hidden" preserveAspectRatio="none" style="position: absolute; top: 0px; left: 0px; display: none; width="1568px" height="801px" viewBox="0 0 597408 305181"></svg>
<svg xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink" version="1.1" fill-rule="evenodd" fill="none" stroke="none" stroke-linecap="square" stroke-miterlimit="10" overflow="hidden" preserveAspectRatio="none" style="position: absolute; top: 0px; left: 0px; display: none; width="1568px" height="801px" viewBox="0 0 597408 305181"></svg>
<svg xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink" version="1.1" fill-rule="evenodd" fill="none" stroke="none" stroke-linecap="square" stroke-miterlimit="10" overflow="hidden" preserveAspectRatio="none" style="position: absolute; top: 0px; left: 0px; width="565px" height="801px" viewBox="0 0 215265 305181">
  <rect fill="#####" fill-opacity="0" width="100%" height="100%"></rect>
  <g transform="translate(6858 96012) scale(0.553125)">
    <defs cursor="default"></defs>
    <g id="editor-ge311826f4e_0_8">
      <path fill="###" fill-opacity="1" d="M 0 0 L 365760 365760 205740 205740 Z"></path>
      <g id="editor-ge311826f4e_0_8-bg"></g>
      <g id="editor-ge311826f4e_0_9">
        <g></g>
        <g></g>
        <g></g>
      </g>
      <path fill="#000" fill-opacity="0" stroke="#000" stroke-opacity="0" stroke-width="6888.135593220934" stroke-linecap="round" stroke-linejoin="round" stroke-miterlimit="10" pointer-events="visiblePainted" d="M 12468 17801 L 353292 17801 353292 40709 12468 40709 Z"></path> == $0
    </g>
    <g></g>
  </g>
  <g id="editor-ge311826f4e_0_10"></g>
</svg>
</div>
<div class="punch-animation-preview-pane" style="display: none; background: rgb(238, 238, 238);"></div>
<div class="sketchy-auto-fit-contextual-menu-container docs-ui-unprintable" style="direction: ltr; left: 110px; top: 137px; display: none;"></div>
<div class="docs-ui-toast" role="complementary" tabindex="0" aria-describedby=":50.d-u-t-c" style="display: none;"></div>
<div class="docs-explore-widget docs-ui-unprintable docs-explore-widget-active docs-explore-widget-open" style="margin-right: 17px; margin-left: 17px; margin-bottom: 8px;" id="punch-explore-widget"></div>
</div>
... pace.cursor-default div#pages svg g g#editor-ge311826f4e_0_8 g#editor-ge311826f4e_0_9 g g path ...
```

Styles Computed Layout Event Listeners

Filter :hov .cls +

Pseudo ::scrollbar-button element

```
::-webkit-scrollbar-button 639357884-e-s_ltr.css:2 button {
  height: 0;
  width: 0;
}
```

Pseudo ::scrollbar-corner element

```
::-webkit-scrollbar-corner 639357884-e-s_ltr.css:2 corner {
  background: transparent;
}
```

Pseudo ::scrollbar-thumb element

```
.docs-gm ::-webkit-scrollbar-thumb 639357884-e-s_ltr.css:2 scrollbar-thumb {
  border-style: solid;
  border-color: transparent;
  border-width: 4px;
  background-color: #dadce0;
  border-radius: 8px;
  box-shadow: none;
}
::-webkit-scrollbar-thumb 639357884-e-s_ltr.css:2 thumb {
  background-color: rgba(0,0,0,.2);
  background-clip: padding-box;
  border: solid transparent;
  border-width: 1px 1px 1px 6px;
  min-height: 28px;
  padding: 100px 0 0;
  box-shadow:
    inset 1px 1px 0 rgb(0 0 / 10%),
    inset 0 -1px 0 rgb(0 0 / 7%);
}
```

Pseudo ::scrollbar-track element

```
.docs-gm ::-webkit-scrollbar-track 639357884-e-s_ltr.css:2 scrollbar-track {
  box-shadow: none;
  margin: 0 4px;
}
::-webkit-scrollbar-track 639357884-e-s_ltr.css:2 track {
  background-clip: padding-box;
  border: solid transparent;
  border-width: 0 0 4px;
}
```

margin - - - -

border - - - -

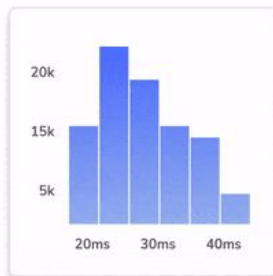
padding - - - -

auto x auto

Скриншот тесты

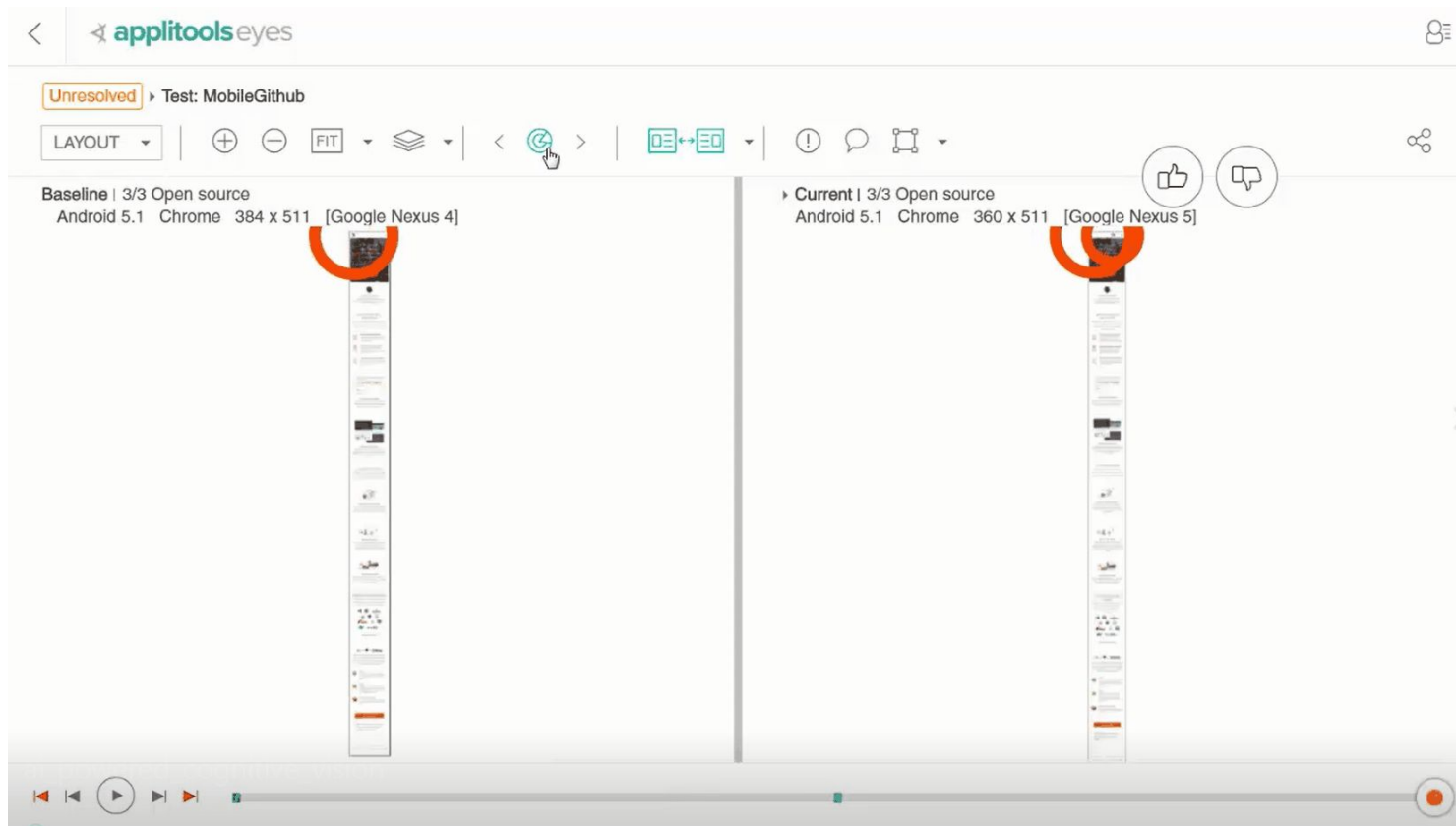


Storybook

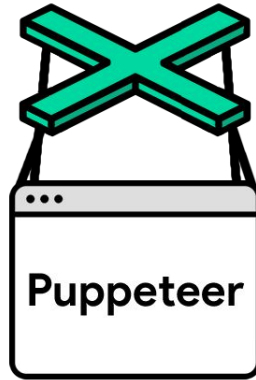


Capture image of each component

Еще вариант



Ну и наконец тестирование UI с помощью кода



cypress.io

Немного о селекторах

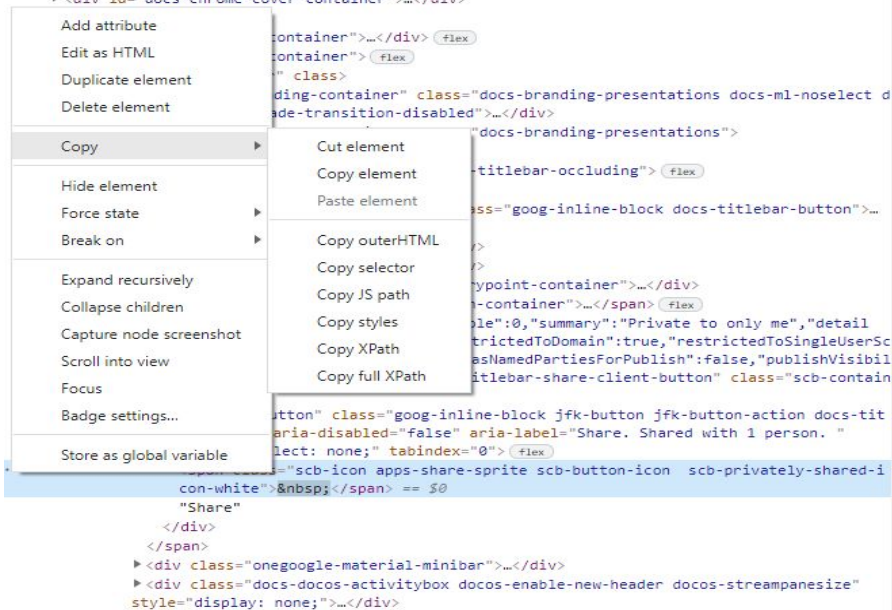
Для примера возьмём следующий XHTML документ.

```
1 <html>
2 <body>
3   <div>Первый слой
4     <span>блок текста в первом слое</span>
5   </div>
6   <div>Второй слой</div>
7   <div>Третий слой
8     <span class="text">первый блок в третьем слое</span>
9     <span class="text">второй блок в третьем слое</span>
10    <span>третий блок в третьем слое</span>
11  </div>
12  <span>четвёртый слой</span>
13  <img />
14 </body>
15 </html>
```

XPath-путь `/html/body/*<span[@class]` будет соответствовать в нём двум элементам исходного документа :

```
<span class="text">первый блок в третьем слое</span> и
<span class="text">второй блок в третьем слое</span>.
```

```
<html data-cast-api-enabled="true" lang="en">
  <head>...</head>
  <body dir="ltr" role="application" class="docs-gm" itemscope itemtype="http://schema.org/CreativeWork/PresentationObject">
    <iframe tabindex="-1" aria-hidden="true" style="position: absolute; width: 9em; height: 9em; top: -99em;">...</iframe>
    <iframe tabindex="-1" aria-hidden="true" style="position: absolute; width: 9em; height: 9em; top: -99em;">...</iframe>
    <span id="sketchy-wrapped-iframe" style="position: absolute; top: -5000px; white-space: nowrap; overflow: hidden">...</span>
    <script nonce>...</script>
    <noscript>...</noscript>
    <meta itemprop="name" content="Untitled presentation">
    <meta itemprop="faviconUrl" content="https://ssl.gstatic.com/docs/presentations/images/favicon.ico">
    <meta itemprop="url" content="https://docs.google.com/presentation/u/1/d/1V8jNTXr29uc-zbbbhAXzUvqPM-TDU5YhR8_ea9PLuI4/edit?usp=drive_web&ouid=104553772326030521448&usp=embed_googleplus">
    <meta itemprop="embedURL" content="https://docs.google.com/presentation/u/1/d/1V8jNTXr29uc-zbbbhAXzUvqPM-TDU5YhR8_ea9PLuI4/preview?usp=embed_googleplus">
    <div id="docs-chrome" tabindex="0" role="banner" aria-label="Menu bar" class="docs-material companion-enabled">
      <div id="docs-butterbar-container" class="docs-butterbar-container" aria-live="assertive" aria-atomic="true"></div>
      <div id="docs-chrome-cover-container">...</div>
```



Puppeteer

```
const puppeteer = require('puppeteer');
const iPhone = puppeteer.devices['iPhone 6'];
(async () => {
  const browser = await puppeteer.launch();
  const page = await browser.newPage();
  //выбираем тип устройства с которым хотим взаимодействовать
  //await page.emulate(iPhone);
  await page.setViewport({width: 1366, height: 768});
  await page.setUserAgent('Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/78.0.3904.108 Safari/537.36');
  //заходим на нужную страницу
  await page.goto('http://localhost:3000/');
  await page.waitForSelector("#root > div > div.component-search-input > div > input"); //ждем когда загрузится элемент

  //симулируем ввод клавиатурой
  await page.focus("#root > div > div.component-search-input > div > input")
  await page.keyboard.type('Smile')

  //делаем скриншот
  await page.screenshot({ path: 'example3.png' });

  await browser.close();
})();
```

C:\Users\mikhail.yabinin\Downloads\emoji-search-master

File Edit View Window Help

emoji-search-master

Support Docs Log In

Tests Runs Settings

Welcome to Cypress!

We've created some sample test files that demonstrate key Cypress concepts to help you get started.
[How to write your first test](#) | No thanks, delete example files

Press Ctrl + F to search... [+ New Spec File](#)

INTEGRATION TESTS [COLLAPSE ALL](#) | [EXPAND ALL](#) [Run 20 integration specs](#)

- 1-getting-started
 - todo.spec.js
- 2-advanced-examples
 - actions.spec.js
 - aliasing.spec.js
 - assertions.spec.js
 - connectors.spec.js
 - cookies.spec.js
 - cypress_api.spec.js
 - files.spec.js
 - local_storage.spec.js
 - location.spec.js
 - misc.spec.js
 - navigation.spec.js
 - network_requests.spec.js
 - querying.spec.js
 - spies_stubs_clocks.spec.js
 - traversal.spec.js
 - utilities.spec.js
 - viewport.spec.js
 - waiting.spec.js
 - window.spec.js

Version 8.0.0 [Changelog](#)

Selenium

The screenshot displays the Selenium IDE interface for a test suite named "Potikal V Smile". The test is running on the URL "http://localhost:3000". The test suite contains four steps:

Step	Command	Target	Value
3	click	css=input	
4	type	css=input	Smile
5	click	css= .component-emoji-result-row:nth-child(2)	
6	click	css=html	

The log at the bottom of the interface shows the following entries:

- 1. open on / OK 13:22:11
- 2. setWindowSize on 914x1060 OK 13:22:11
- 3. click on css=input OK 13:22:11
- 4. type on css=input with value Smile OK 13:22:11
- 5. click on css= .component-emoji-result-row:nth-child(2) OK 13:22:12
- 6. click on css=html OK 13:22:12

'Potikal V Smile' completed successfully 13:22:12

Некоторые дополнительные утилиты

1. Расширения для браузера позволяющие сделать скриншот всей страницы.
2. Иметь ADBlock для того чтобы понимать как он реагирует на тестируемый сайт
3. Наличие расширения для VPN (ну это просто реально полезно вообще везде)
4. Наличие какого либо ПО для удобной и гибкой записи экрана (идеально было бы с возможностью быстрого комментирования) тут к сожалению реально крутых инструментов не подскажу (но я юзал для этих целей ПО Скриншотер, но он далеко не самый удобный)