Разработка ПРИЛОЖЕНИЙ

Занятие #1. Основы React.

Компоненты, JSX, git, JS

ПЛАН

01 Git

02 Компоненты

03 Практика

04 JS снова

Кто препод?

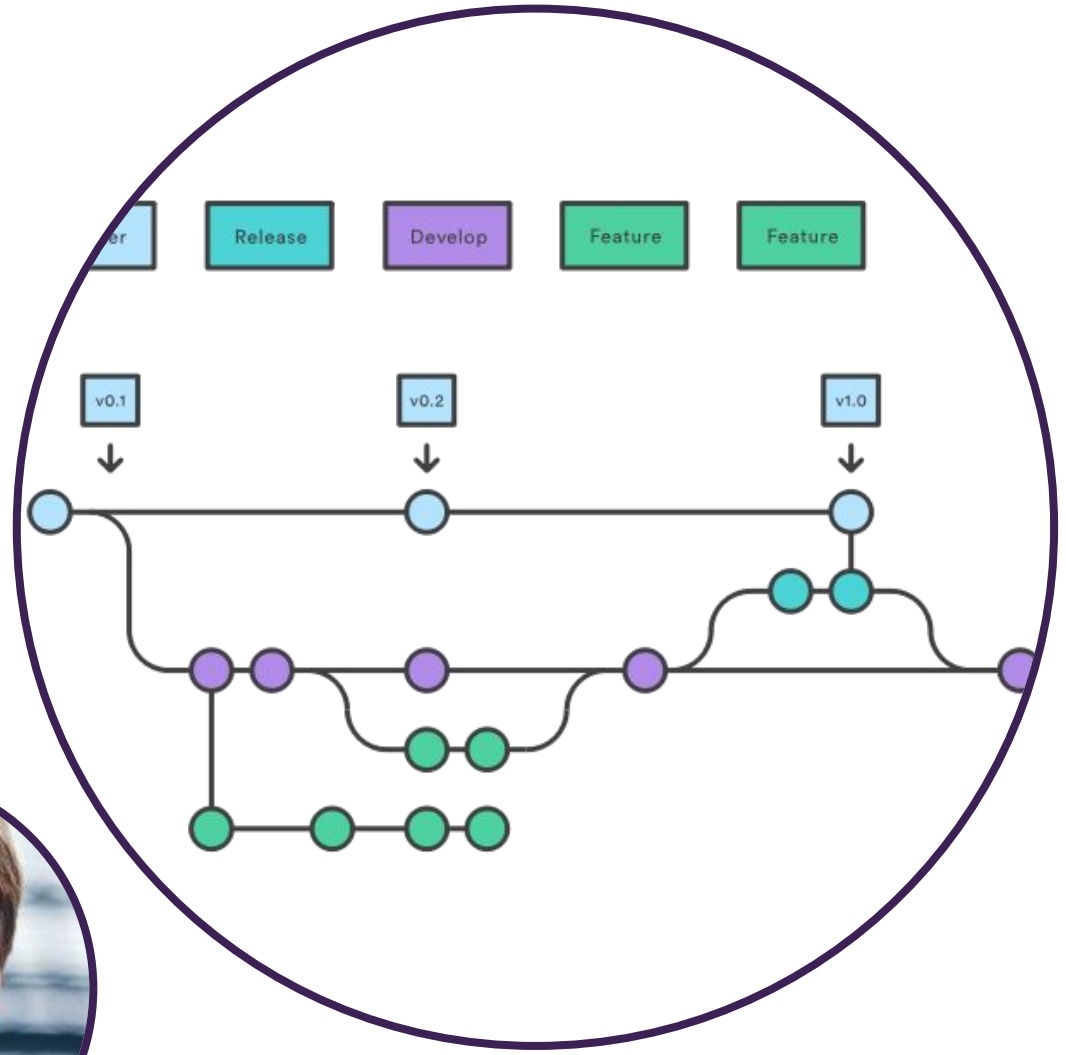
Данил Давлетов

- СГПИ КубГУ 2012, “Информатика”
- СПбПУ ИППТ ВШТП 2022,
  “Technology leadership and Entrepreneurship”
- ОАО "Зарубежэнергопроект", Иваново, DBA
- ФГБУК “музей-заповедник “Фанагория”, инженер-программист
- Проекты: “I bet I can”, “Булка с МАК”, “Парк проектов”

01 Git

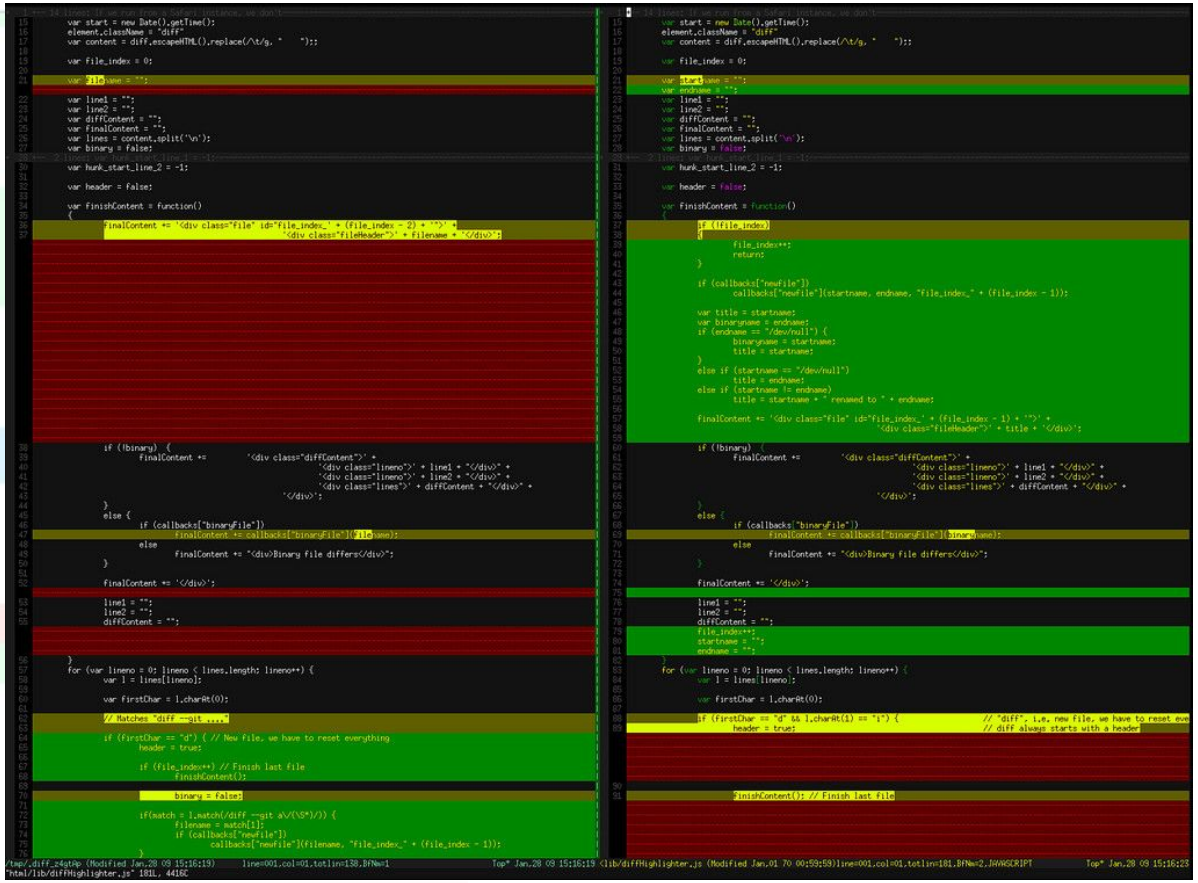
git. зачем.

- Хранение версий
- Децентрализация
- Совместная разработка
- Стандарт разработки
- Жизненный цикл проекта



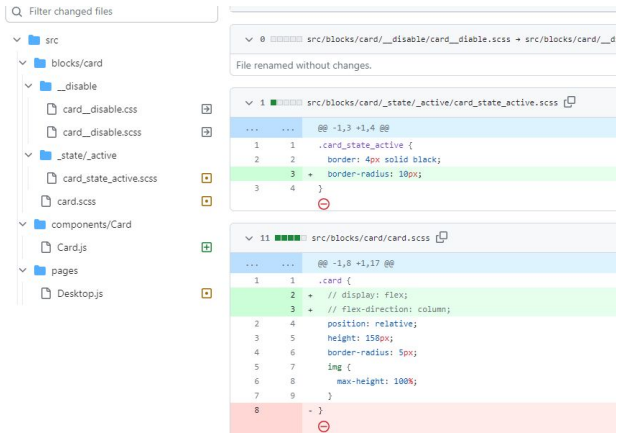
git diff

```
 4   5   import Header from "../components/Header/Header";
 5   6   import Button from "../components/Button/Button";
 6   7   import Main from "../components/Main/Main";
 7   8   import Spoiler from "../components/Spoiler/Spoiler";
     9 + import Card from "../components/Card/Card";
    10 + import Footer from "../components/Footer/Footer";
    11 + import CardSlider from "../components/CardSlider/CardSlider";
 8  12
 9  13   export default function () {
10  14     const navigate = useNavigate();
           @@ -19,15 +23,21 @@ export default function () {
19  23     console.log(currentUser);
20  24
21  25     return (
22      -    <>
    26 +    <Content>
23  27       <Header>
24  28         <Button>Пригласить пользователя</Button>
25  29       </Header>
26  30       <Main type="table">
27  31         {/* TODO: разобраться с button_func_random */}
28      -        <Button type="circle" func="random"></Button>
    32 +        <Button type="circle" func="random"></Button>'
    33 +        <Card />
29  34       </Main>
30  35       <Spoiler />
31      -    </>
    36 +      <Footer>
    37 +        <CardSlider />
```



Виды

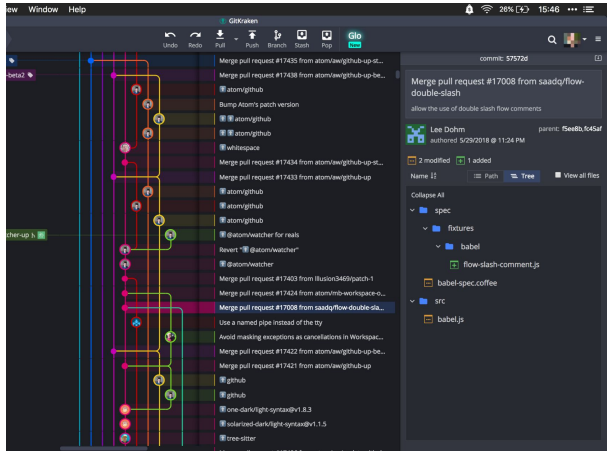
Консоль



Web

GitHub

GitLab

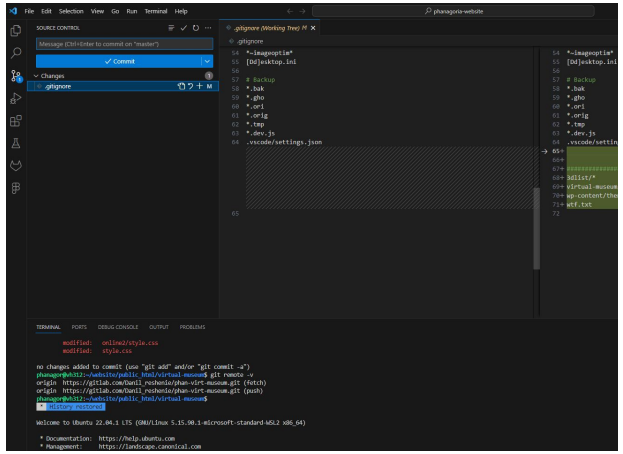


GUI

GitKraken

SourceTree

GitHub Client



плагины

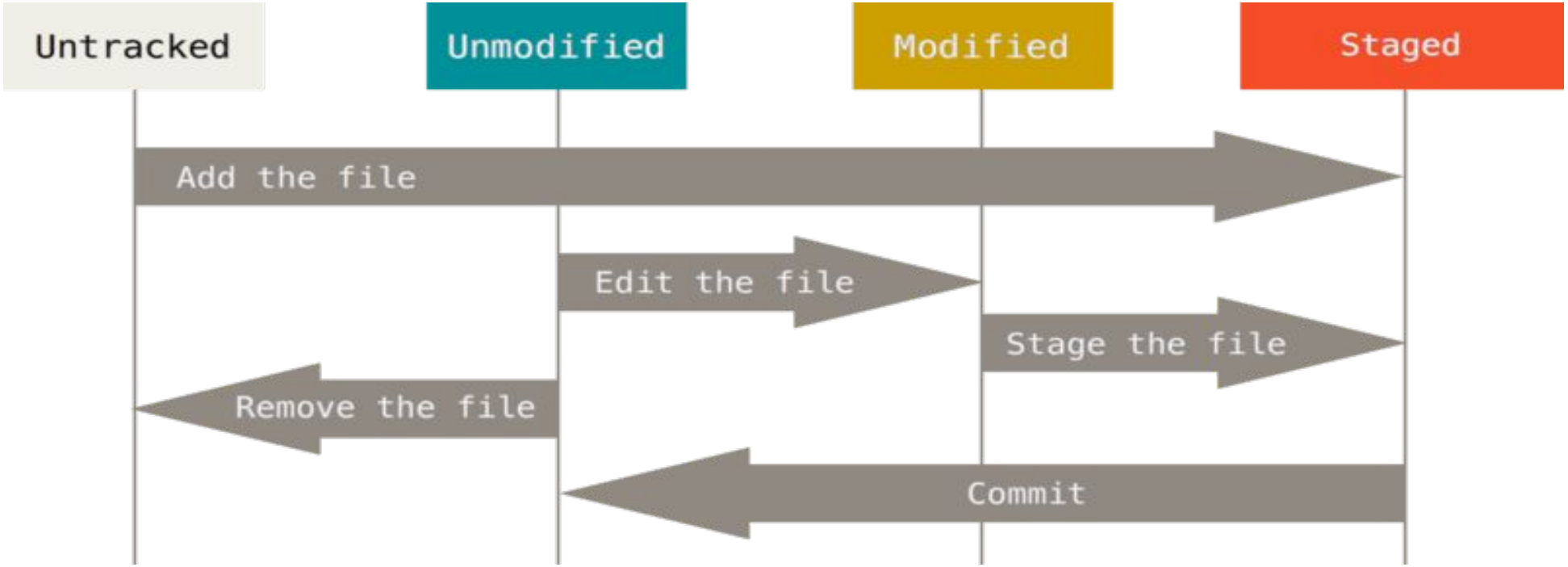
VS Code

Sublime Merge

Основные команды

- **git init** - создание репозитория
- **git clone <адрес>** - клонирование репозитория
- **git config [<options>]** - настройка общих параметров
- **git status** - просмотр состояние файлов
- **git add** - добавление в отслеживаемые файлы
- **git diff [<options>]** - просмотр изменений
- **git commit** - зафиксировать изменения
- **git commit --amend** - перезаписать последний коммит
- **git rebase -** изменение коммитов в истории
- **git log** - просмотр истории коммитов (в текущей ветке)

Жизненный цикл

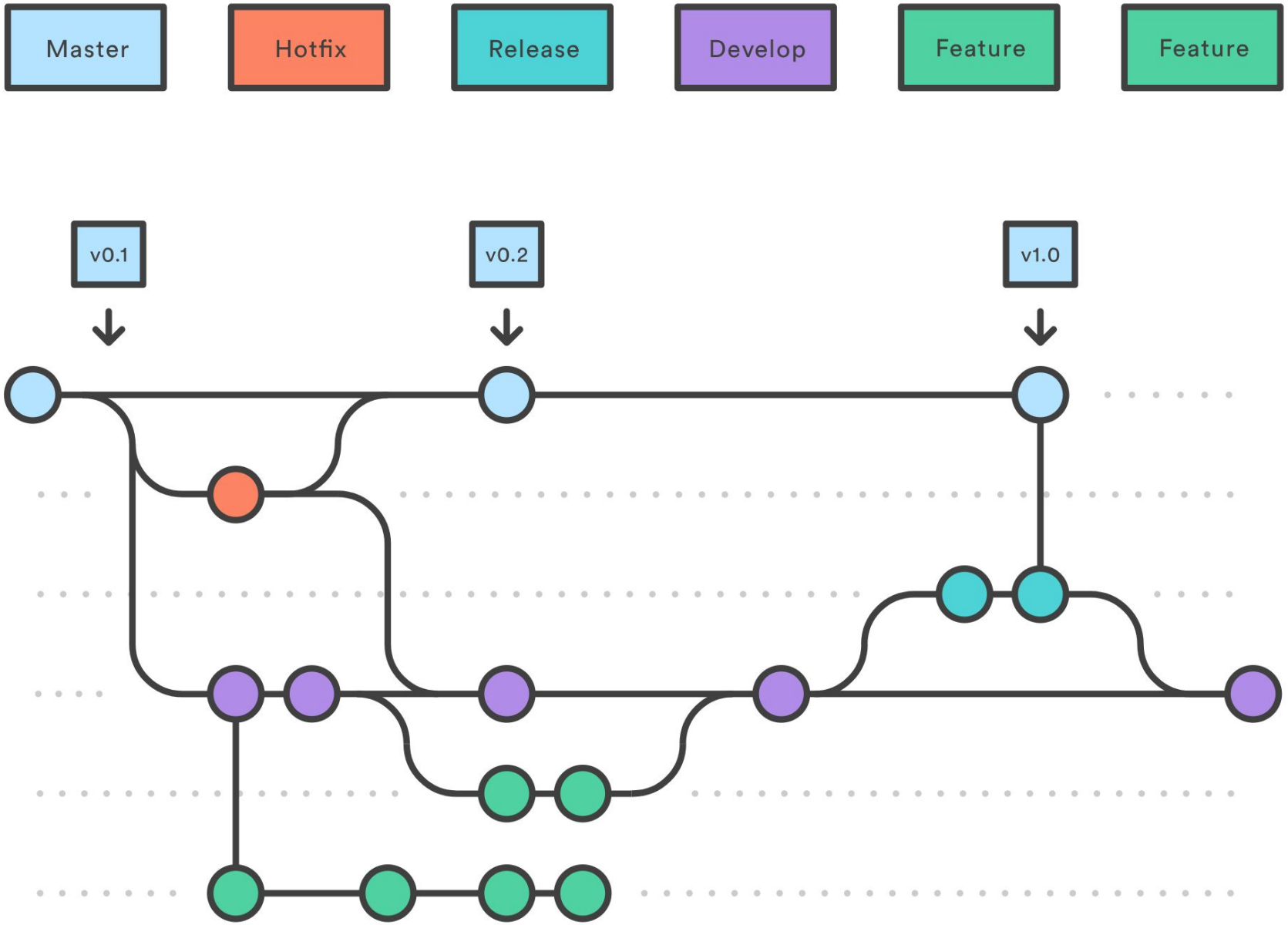


- git status
- git add
- git commit -m “message”

Работа с удаленным репозиторием

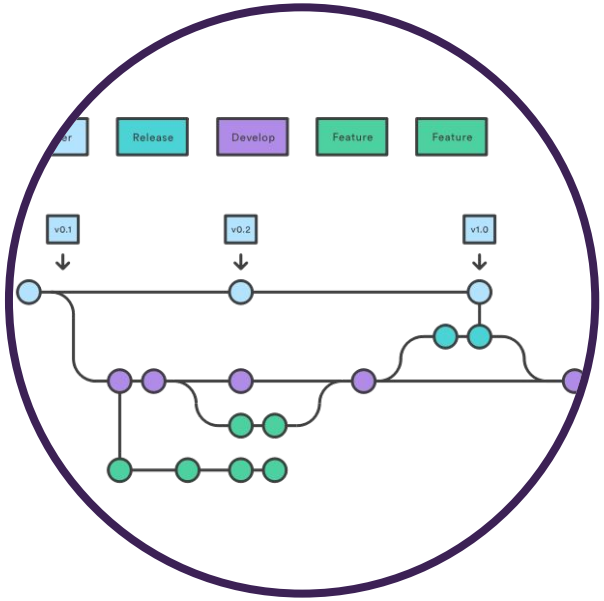
- **git clone <адрес>** - клонирование репозитория
- **git pull -** подтянуть изменения
- **git push -** отправить изменения
- **git remote** - работа с записями репозиториев
  - git remote -v (просмотр)
  - git remote add (добавление)

Работа с ветками



Работа с ветками

- **git branch <branch_name>** - создать ветку
- **git checkout <branch_name> -** переключение веток *
  - с ключом “**-b**”: создает ветку и переключает на неё
- **git merge <branch_name>** - слияние веток



*переключиться на коммит: **git checkout [<commit>]**

Игнорируемые файлы, файл .gitignore

.gitignore

```
# Logs
logs
*.log
npm-debug.log*
yarn-debug.log*
yarn-error.log*
firebase-debug.log*
firebase-debug.*.log*

# Firebase cache
.firebase/

# Firebase config

# Uncomment this if you'd like others to create their own Firebase project.
# For a team working on the same Firebase project(s), it is recommended to leave
# it commented so all members can deploy to the same project(s) in .firebaserc.
# .firebaserc

# Runtime data
pids
*.pid
```

- не храним бинарные файлы

Практика*

```
git config --global user.name "Rick Sanchez"
git config --global user.email "rick_s@yahoo.com"
```

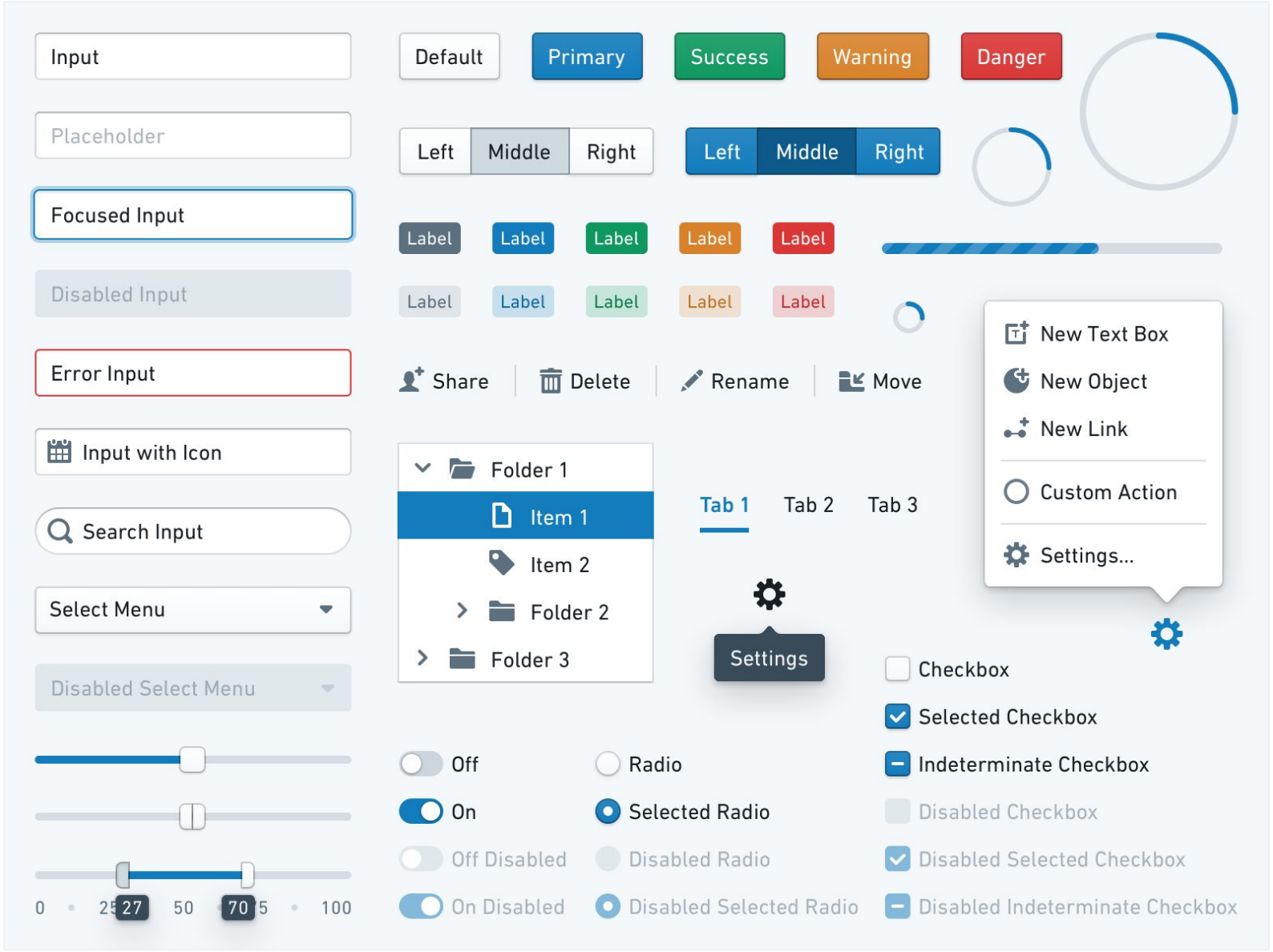
Клонировать презентацию из репозитория

Ссылки, материалы

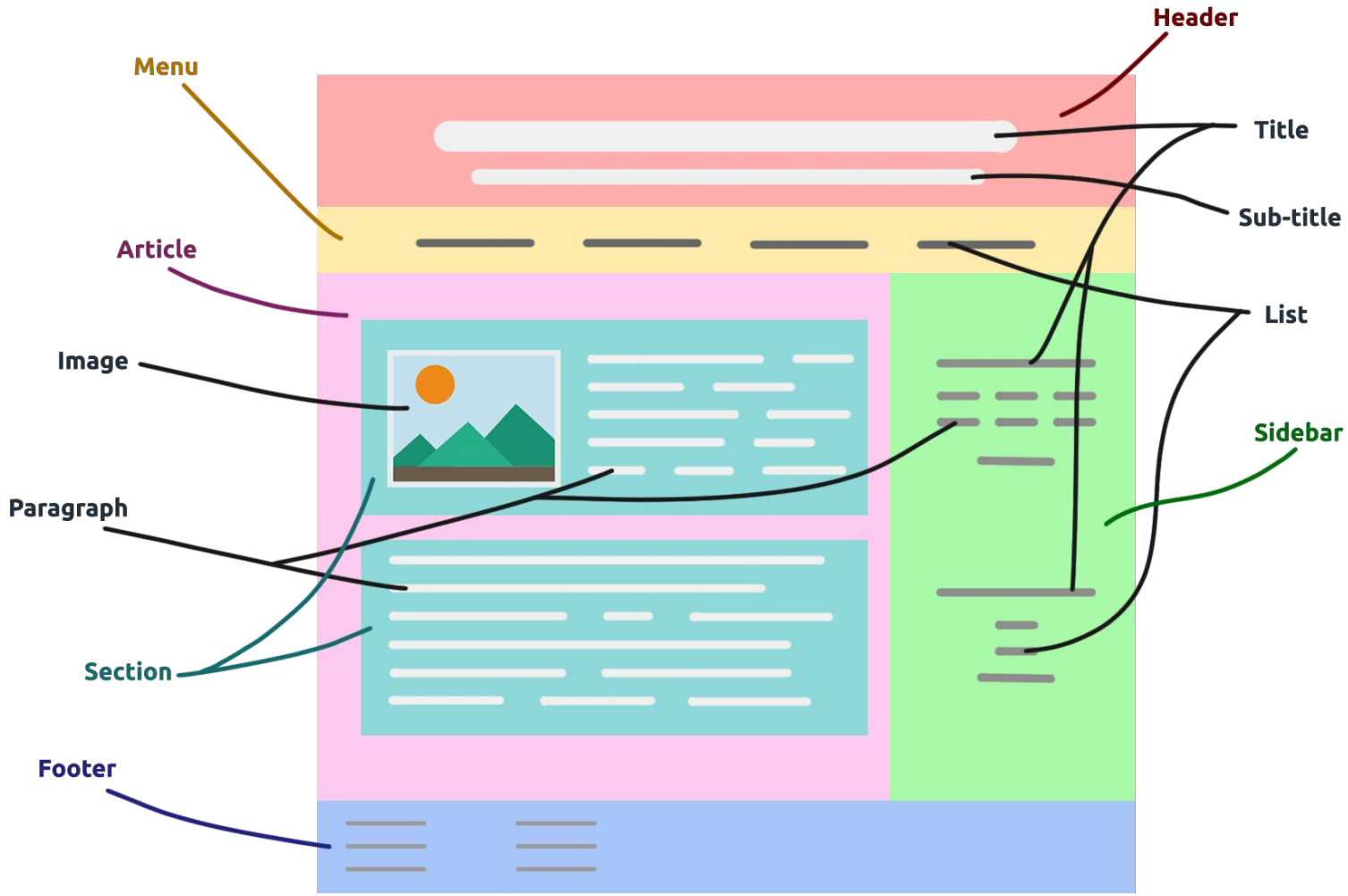
- https://git-scm.com/
- https://habr.com/ru/companies/ruvds/articles/599929/
- https://proglib.io/p/git-for-half-an-hour

02 Компоненты

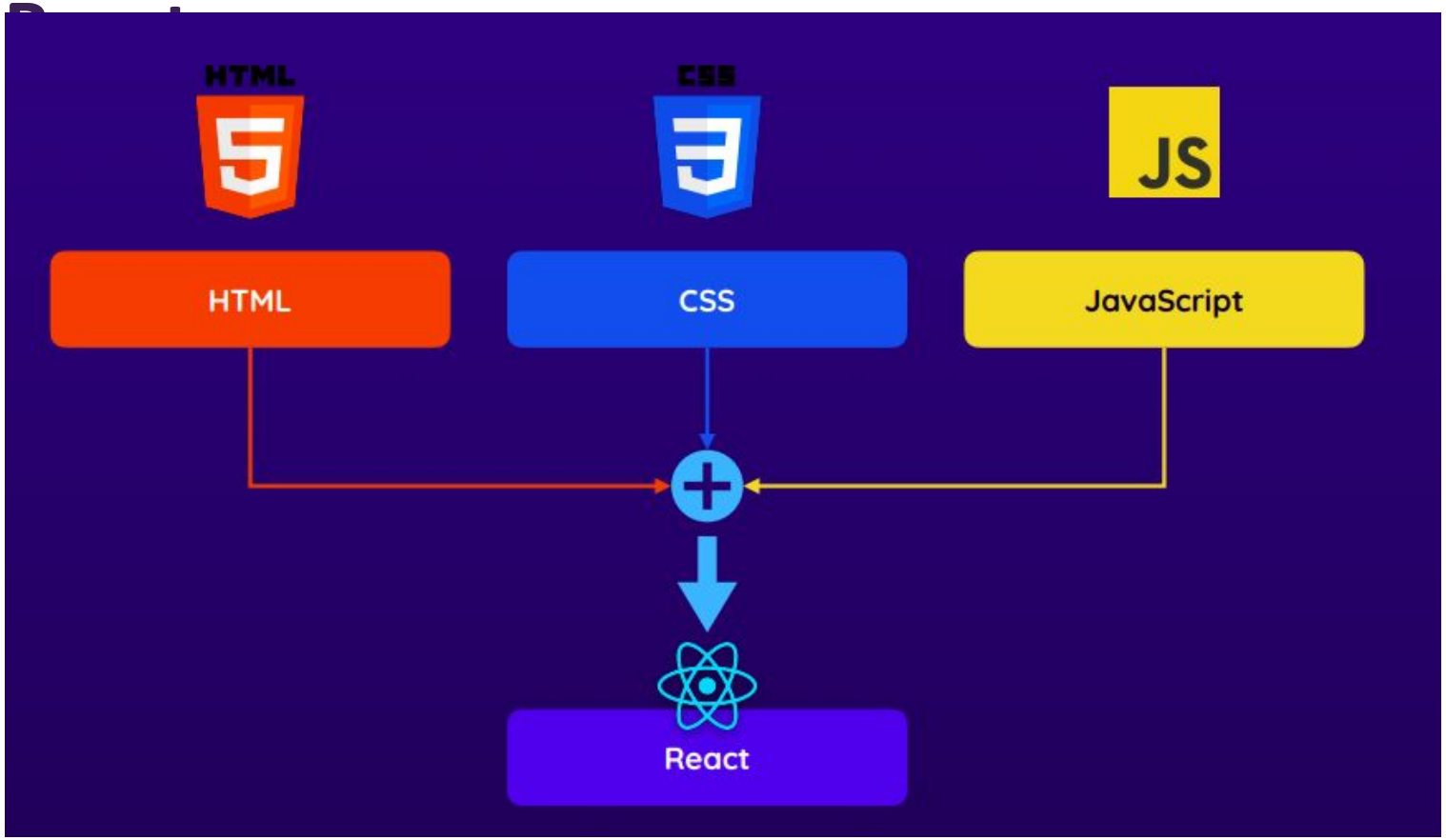
UI - компоненты



Компоненты


Header
Menu
Title
Sub-title
Article
List
Image
Sidebar
Paragraph
Section
Footer

Компоненты в

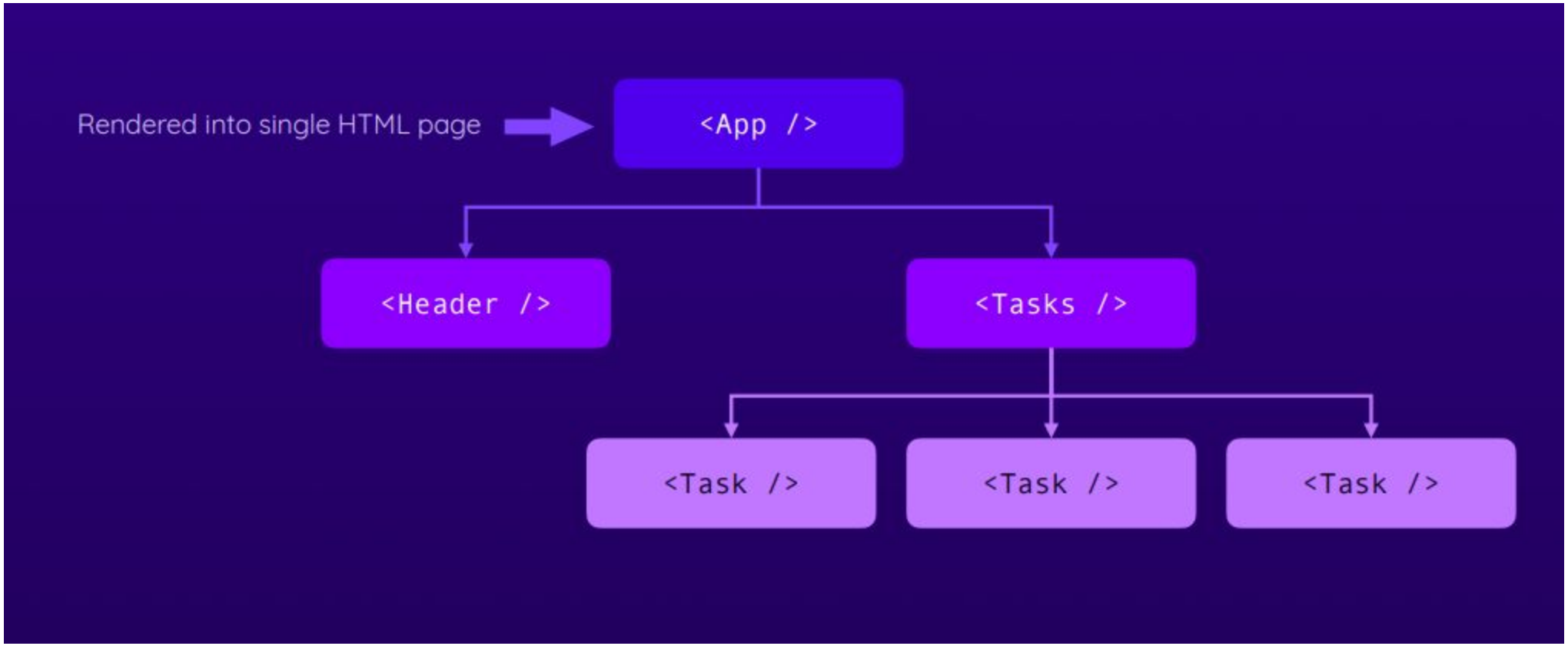


Зачем?

- Переиспользовани
  е
- Разделение

- HTML + CSS + JS = React component (JSX)
- Декларативный подход

Компоненты в React



Типовой компонент

Button.js

```
const Button = function () {
  return <a class="button">Я кнопка</a>;
};
export default Button;
```

App.js

```
import Button from "./components/Button";

function App() {
  return (<Button></Button>);
}
```

Создание проекта

Новый проект, локально:

- npx create-react-app my-app < реакт проект + git репозиторий

Существующий проект:

- git clone < клонировать репозиторий
- npm install < установить пакеты локально

Web песочница

- http://react.new < в браузере

Демонстрация

- Как работает React
- JSX
- Компоненты

Установка пакетов

- Google it
- npm install <название пакета>

mui.com/material-ui/

https://mui.com/material-ui/getting-started/installation/

03 Практика

Что делать

- *Создать новый проект (повтор)
- Клонировать проект из репозитория (форк)
- Создать компонент <Card />
- Добавить сторонние компоненты
- Отправить в свой репозиторий

04 Вспомним JS

Вспомним JS

- ES6
- <script src=””></script> <script></script>
- import, export, type=”module”
- let, const
- arrow func
- arr.map()
- destructing (array, obj) [,]
- Spread,Rest