Разработка ПРИЛОЖЕНИЙ

Занятие #2.

ПЛАН

01 Повтор

02 Списки и пропсы

03 Хуки

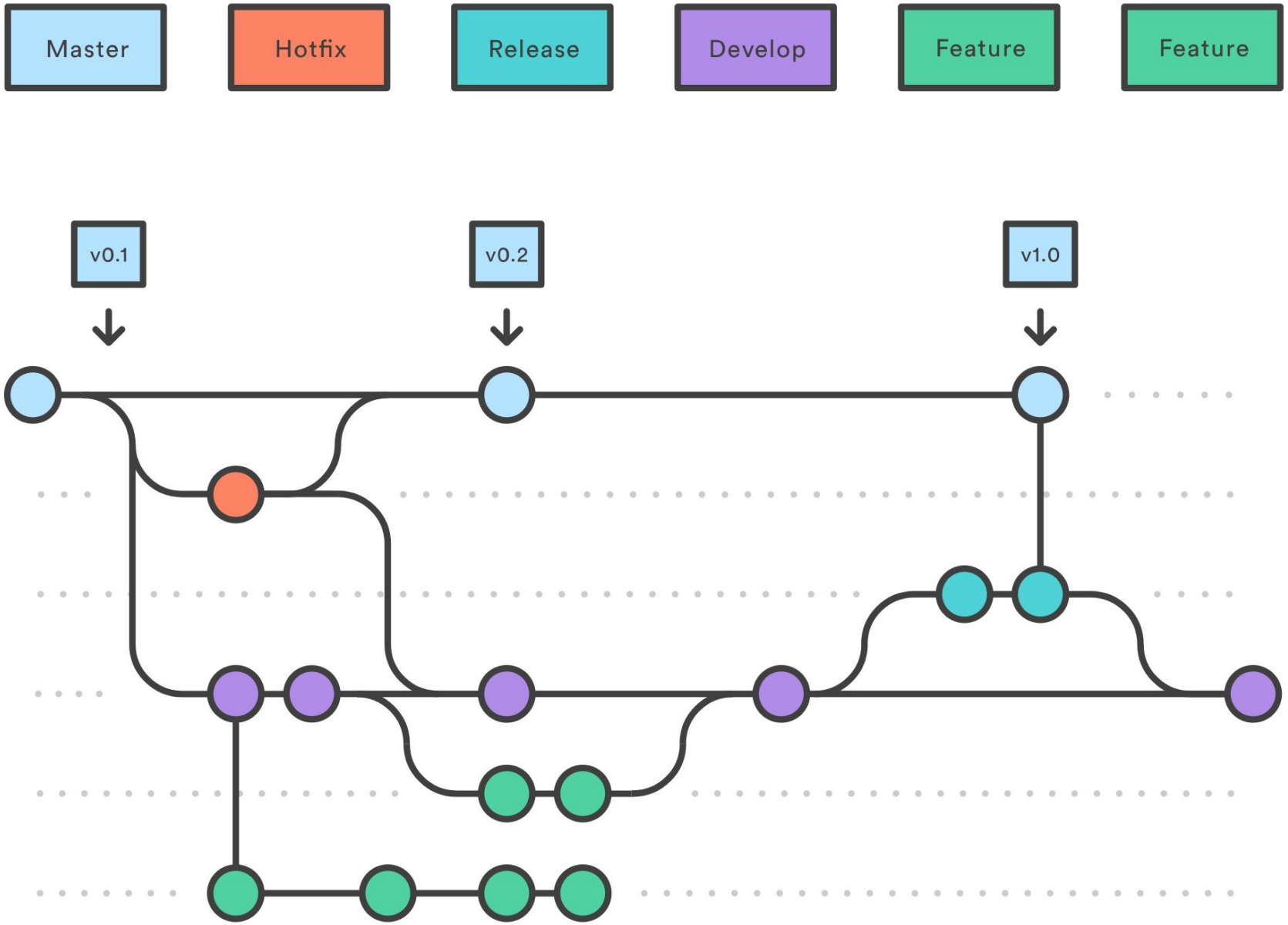
Прогресс

- ★ create-react-app, основы, npm
- ★ компоненты, jsx
- ★ git

- ❑ Состояния, переменные, props
- ❑ Стилизация
- ❑ useState(), useEffect(), useRef()
- ❑ Маршруты, React Router
- ❑ Redux, useContext()
- ❑ Firebase и API
- ❑ Тест, деплой
- ❑ БЭМ, цикл проекта, figma
- ❑ PWA
- ❑ Улучшения, библиотеки
- ❑ Отладка
- ❑ React + TS

01 Повтор

Работа с ветками



Работа с удаленным репозиторием

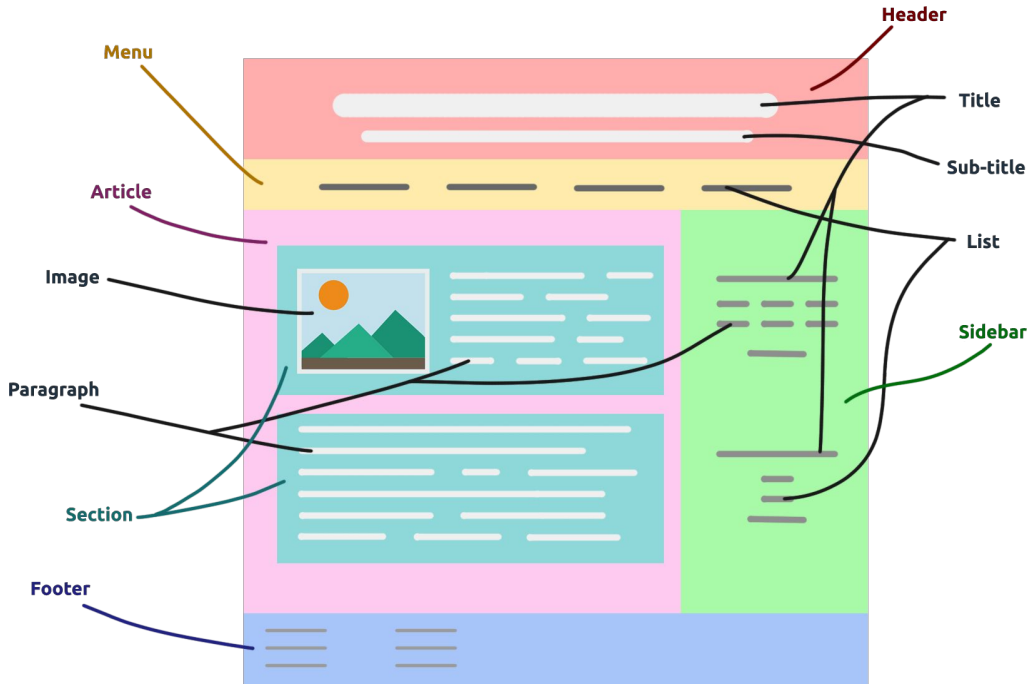
- **git clone <адрес>** - клонирование репозитория
- **git pull -** подтянуть изменения
- **git push -** отправить изменения
- **git remote** - работа с записями репозиториев
  - git remote -v (просмотр)
  - git remote add (добавление)

Вопросы

- Как инициализировать репозиторий?
- Как клонировать репозиторий?
- Как сделать форк?
- Как сделать коммит?
- Как отправить коммиты в удаленный репозиторий?
- Как исключить файлы из отслеживания?

Компонент

- Любой UI элемент в React
- Функция на JavaScript
- возвращает часть UI
- использует JSX разметку
- Большие и маленькие
- Вложенные и нет
- Сторонние и собственные



Типовой компонент

Button.js

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

использование

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

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

- скобки в return
- с большой буквы

JSX правила

- “HTML in JS”
- Единственный корневой элемент
- JS в {фигурных скобках}
- Все теги должны быть закрыты
- camelCase (верблюжий регистр)

Конвертер html в JSX:
https://transform.tools/html-to-jsx

02 Списки, props

Демонстрация

- множественный экспорт
- вложенность и контент, props
- условный рендеринг
- списки: filter(), map(), key
- переменные и рендеринг

03 Хуки

Вспомним JS

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

События JS

onchange	An HTML element has been changed
onclick	The user clicks an HTML element
onmouseover	The user moves the mouse over an HTML element
onmouseout	The user moves the mouse away from an HTML element
onkeydown	The user pushes a keyboard key
onload	The browser has finished loading the page

справочник

Хуки

- useState() - управление состоянием
- useEffect() - синхронизация (зависимости)
- useRef() - сослаться на значение, которое не нужно для рендеринга

ДЗ

- **Передавать разные фото котов в компонент Cats из массива** (проще)
- **Почитать про useEffect и починить новых котов.** (сложнее)