

# Урок №6

## А. Регулярные выражения

### ДЕТАЛИ:

- Знакомство с регулярными выражениями
- Модуль re
- GitHub



# Git

Git - это консольная утилита, для отслеживания и ведения истории изменения файлов, в вашем проекте. Чаще всего его используют для кода, но можно и для других файлов. Например, для картинок - полезно для дизайнеров.

С помощью Git-а вы можете откатить свой проект до более старой версии, сравнивать, анализировать или сливать свои изменения в репозиторий.

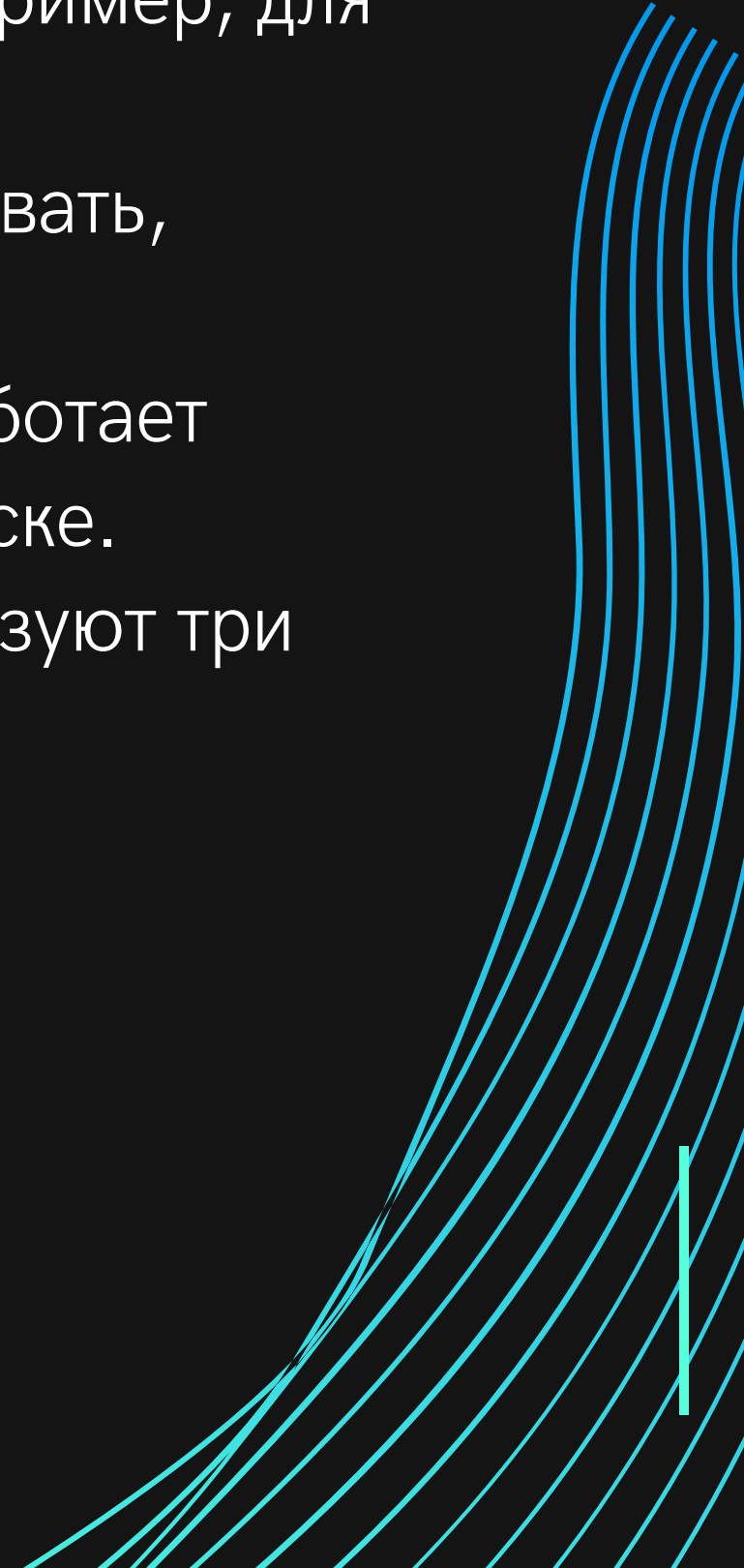
Репозиторием называют хранилище вашего кода и историю его изменений. Git работает локально и все ваши репозитории хранятся в определенных папках на жестком диске.

Так же ваши репозитории можно хранить и в интернете. Обычно для этого используют три сервиса:

- GitHub
- Bitbucket
- GitLab



git





GitHub — это сервис для совместной разработки и хостинга проектов. С помощью GitHub над кодом проекта может работать неограниченное количество программистов из любых точек мира. В GitHub есть система контроля (управления) версий Git: сервис позволяет просматривать и контролировать любые изменения кода любым разработчиком и возвращаться к состоянию до изменений.

В целом GitHub — это социальная сеть для разработчиков, в которой можно найти проекты с открытым кодом от других разработчиков, практиковаться в написании кода и хранить свое портфолио.



# Регулярные выражения

Регулярные выражения (Regex) – это строки, задающие шаблон для поиска определенных фрагментов в тексте. Помимо поиска, с помощью специальных Regex-шаблонов можно манипулировать текстовыми фрагментами – удалять и изменять подстроки частично или полностью.



## Правило поиска для точки

. — любой символ

\. — точка



# Диапазон допустимых значений

**[нл]** — только «н» и «л»

**[а-я]** — все русские буквы в нижнем регистре от «а» до «я» (кроме «ё»)

**[А-Я]** — все заглавные русские буквы **[А-Яа-**

**яЁё]** — все русские буквы

**[a-z]** — латиница мелким шрифтом

**[a-zA-Z]** — все английские буквы

**[0-9]** — любая цифра

**[В-Ю]** — буквы от «В» до «Ю» (да, диапазон это не только от А до Я)

**[А-ГО-Р]** — буквы от «А» до «Г» и от «О» до «Р»

Единственный допустимый разделитель — это дефис

Например: [1-3] - верно, [1-31] - неверно



# Исключение и дополнение допустимых значений

^ внутри [] означает исключение:

[**^0-9**] — любой символ, кроме цифр

[**^ёЁ**] — любой символ, кроме буквы «ё»

[**^а-в8**] — любой символ, кроме букв «а», «б», «в» и цифры 8

[**а-я5,.**] — любой символ от а до я цифра 5 или запятая

# Экранирование спец символов

Вот полный перечень этих символов:

`[ ] \ ^ $ . | ? * + ( )`

Чтобы использовать специальный символ как обычный, добавьте к нему обратную косую черту: `\`.

Это называется «экранирование символа»



# Перечисление вариантов

Чтобы указать допустимые значения для:

- Одного символа — используем `[]`
- Нескольких символов или целого слова — используем `|`



Круглые скобки обозначают группу символов.

Зачем нужны скобки?

Показать, где начинается и заканчивается группа.

Иначе вертикальная черта применится ко всем символам

**A(n|л)я** или **A(нн|л)я**

# Метасимволы

Символ	Эквивалент	Пояснение
\d	[0-9]	Цифровой символ
\D	[^0-9]	Нецифровой символ
\s	[ \f\n\r\t\v]	Пробельный символ
\S	[^ \f\n\r\t\v]	Непробельный символ
\w	[:word:]	Буквенный или цифровой символ или знак подчёркивания
\W	[^[:word:]]	Любой символ, кроме буквенного или цифрового символа или знака подчёркивания
.		Вообще любой символ

# Квантификаторы

Квантификатор	Число повторений
?	Ноль или одно
*	Ноль или более
+	Один или более
{n}	Ровно n раз
{m,n}	От m до n включительно
{m,}	Не менее m

Regex: data{2}  
Найдет: dataa  
Не найдет: datadata

Regex: (data){2}  
Найдет: datadata  
Не найдет: dataa

Не забывайте — квантификатор применяется к последнему символу или группе символов!



# Метасимволы, обозначающие позицию строки

Символ	Значение
\b	граница слова
\B	Не граница слова
^	начало текста (строки)
\$	конец текста (строки)

# ДЗ №6

1. Создать класс под названием `data`, в класс добавить атрибуты `full_name`, `email`, `file_name`, `color`.

Добавить геттеры и сеттеры для всех атрибутов

2. Затем считать из файла `MOCK_DATA.txt`, в котором 1000 строк с данными (Имя и Фамилия, почта, название файла с расширением и код цвета)

3. Из каждой считанной строки создать объект класса `data` и добавить его в список.

4. Затем через цикл пройтись по каждому объекту и записать в 4 разных файла все типы информации. (1й файл: Имена с фамилиями, 2й файл: почта, 3й файл: названия файлов с расширением, 4й файл: коды цветов)