

# Добрый день, друзья

Рад приветствовать на первой лекции курса

«Введение в контроль версий»



# Ильнар Шафигуллин



- Методолог программы «Разработчик»
- Преподаватель мехмата Казанского Университета
- Стаж преподавания 10 лет
- Кандидат физико-математических наук



Введение в контроль версий

# Что это и для чего может быть нужно?



Введение в контроль версий

# Что это и для чего может быть нужно?

Возможность **хранить**  
различные версии проекта

Возможность **возвращаться**  
к различным версиям проекта





# Например

Сайт V 1.0



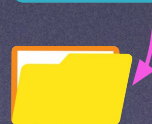
Работающая версия  
сайта на 30 декабря

Сайт V 1.2



Работающая версия  
сайта на 10 января

Сайт V 2.0



Работающая версия  
сайта на 10 мая



Введение в контроль версий

**Это и есть контроль версий!**



Введение в контроль версий

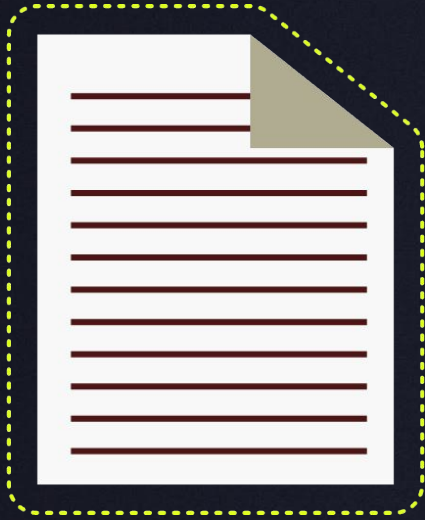
# Это как сохранение в играх!





Введение в контроль версий

## Еще пример... посложнее



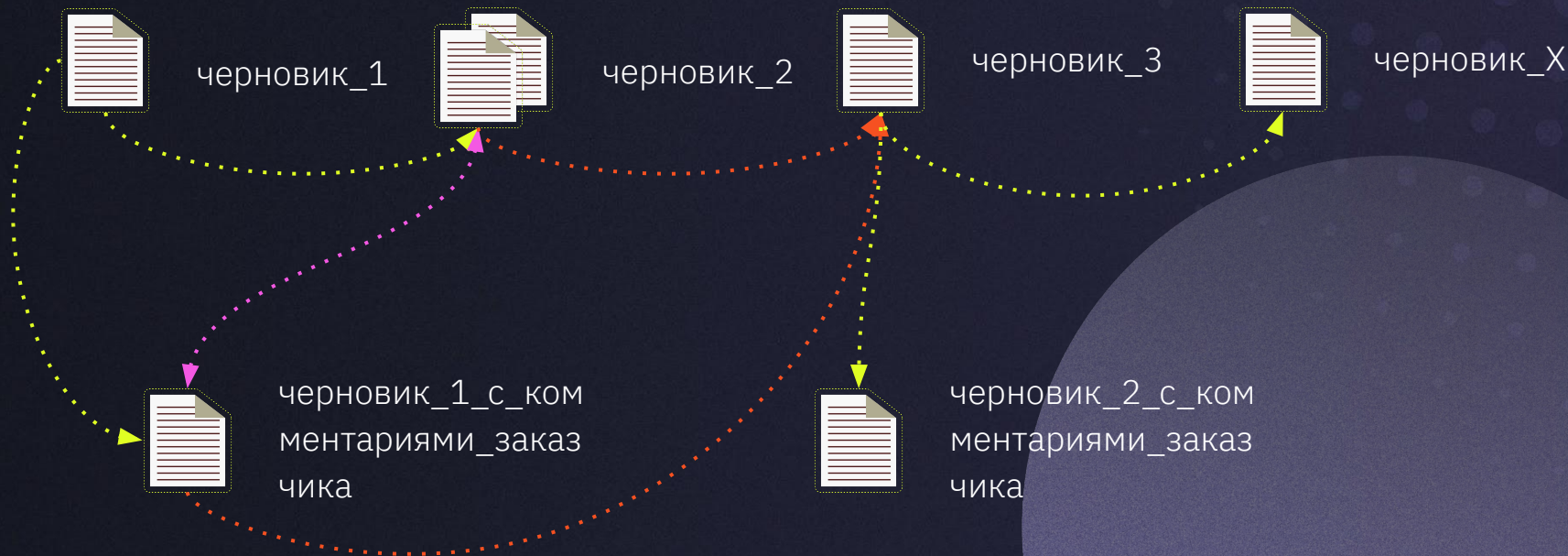
В проекте участвует 2 человека

Главное – эта работа должна  
занимать больше нескольких дней





## Пример посложнее...



Введение в контроль версий

# Печаль...

- Много проблем
- Неудобно
- Долго



А если в команде будет  
**БОЛЬШЕ** людей?







Введение в контроль версий

git



## Линус Торвальдс

Финно-американский программист.  
Создатель операционной системы  
Linux, создатель git



Введение в контроль версий

**git**

**Это просто программа, которая берет  
на себя вопросы контроля версий  
над проектом**



Введение в контроль версий

**git**

Сохраняет в памяти не файлы целиком,  
а **разницу** между файлами



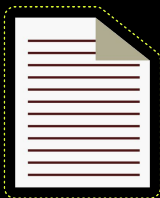
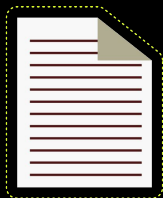


Введение в контроль версий

# Например

Добавим в файл с текстом еще 2 слова и сохраним

Старый принцип



git



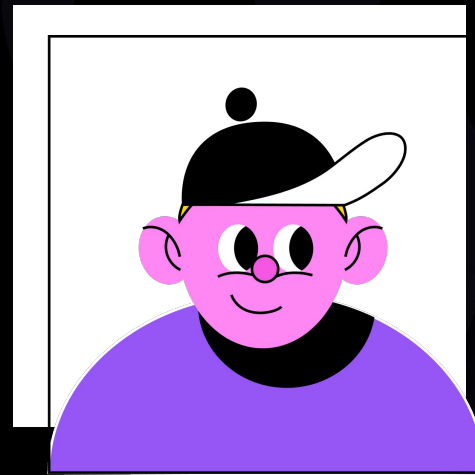
2 слова



Введение в контроль версий

# Что нужно

- Сохранять разные версии
- Перемещаться между ними



Введение в контроль версий

# Для работы потребуется

- Установить на ПК
- Настроить

Все **ссылки** можно найти **под лекцией**,  
а также к этому вопросу вернемся  
на семинарах

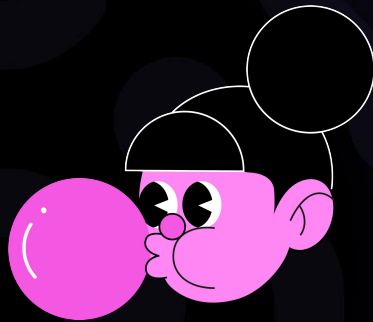




Введение в контроль версий

# Наша задача

Научиться работать  
с контролем версий



Введение в контроль версий

# Как будем решать?

- Как программисты
- Работая с текстами



Введение в контроль версий

## Кстати

**Какой сложностью будет добавление программирования в процесс обучения работе с контролем версий и почему?**

**Напишите в комментариях**



Введение в контроль версий

# Помним!

Исходный код программы – это отформатированный по каким-то требованиям **ТЕКСТОВЫЙ** файл





Введение в контроль версий

# А теперь посмотрим на практике



# Синтаксис языка Markdown.

## Запомнить!

- Жирный текст — **\*\***
- Курсивный текст — *\**
- Зачеркнутый текст — ~~~~~
- Выделяют заголовки — **#** в начале строки
- Показать уровень заголовка — подчеркивание знаками **=** или **-**
- Нумерованные Списки — обозначаются обычными цифрами **1, 2, 3**
- Ненумерованные Списки — обозначаются **знаками \*** в начале строки
- Вложенные Списки — выполняем отступы



Введение в контроль версий

# git init. **Запомнить!**

ПРОБЕЛ ОБЯЗАТЕЛЬНО



```
PS C:\Users\oleg-\OneDrive\Документы\testgit> git init
Initialized empty Git repository in C:/Users/oleg-/OneDrive/Документы/testgit/.git/
```



Введение в контроль версий

# git add. **Запомнить!**

```
PS C:\Users\oleg-\OneDrive\Документы\Lecton1Prepare> git add first.txt
PS C:\Users\oleg-\OneDrive\Документы\Lecton1Prepare> █
```

← Расширение  
файла

ПРОБЕЛ

ПРОБЕЛ

Название файла.  
Расширение  
обязательно



Введение в контроль версий

# git commit. **Запомнить!**

```
PS C:\Users\oleg-\OneDrive\Документы\Lecton1Prepare> git commit -a -m "First.txt changed"
[master ea7c313] First.txt changed
1 file changed, 2 insertions(+), 1 deletion(-)
```

Кавычки обязательны  
с двух сторон

ПРОБЕЛ

ПРОБЕЛ

Флаг сообщения

Текст сообщения





## Введение в контроль версий

# git diff. Запомнить!

```
PS C:\Users\oleg-\OneDrive\Документы\Lecton1Prepare> git diff
diff --git a/Readme.md b/Readme.md
index 5274727..c1f0625 100644
--- a/Readme.md
+++ b/Readme.md
@@ -6,4 +6,8 @@
```

ПРОБЕЛ  
ОБЯЗАТЕЛЬНО

# Ещё их можно обозначать вот так

```
-## а можно и так
\ No newline at end of file
+## а можно и так
+
```

Измененная строка

```
+добавим список
+* Первый элемент
```

Добавленная строка

```
:...skipping...
diff --git a/Readme.md b/Readme.md
index 5274727..c1f0625 100644
--- a/Readme.md
+++ b/Readme.md
@@ -6,4 +6,8 @@
```

Статистика изменений



## Введение в контроль версий

# git log. **Запомнить!**

```
PS C:\Users\oleg-\OneDrive\Документы\Lecton1Prepare> git log  
commit 6f39027914e34f40f432a63aba6364938640ecc9 (HEAD -> master)  
Author: oleg golenishev <oleg-golenishev63@yandex.ru>  
Date: Thu Sep 30 13:01:28 2021 +0300
```

added list

```
commit 3432a6393ccb49ccd7434b04adc9e0ee1f607068  
Author: oleg golenishev <oleg-golenishev63@yandex.ru>  
Date: Thu Sep 30 12:56:56 2021 +0300
```

changed readme

```
commit 95b35e1f5ac4f9c971abed2813212e7d9165726b  
Author: oleg golenishev <oleg-golenishev63@yandex.ru>  
Date: Thu Sep 30 12:31:04 2021 +0300
```

first commit

ПРОБЕЛ  
ОБЯЗАТЕЛЬНО

Номер  
коммита

Сообщение к коммиту



**Всем спасибо!**

