# **Разширения для Chrome**

**uBlock Origin** блокировщик рекламы

**Adblock Unlimted** блокирощик рекламы

**AdBlocker Ultimate** блокировщик рекламы

**Rememberry** для перевода англиского текста на страницах

# **chrome://flags**

Smooth Scrolling -> Enabled

GPU rasterizetion -> Enabled

Zero-copy rasterizer -> Enabled

Parallel downloading -> Enabled

# **Настройки VS Code Снаружи**

Верхняя Линия -> Command Center -> **---** отключает отображение поиска по центру

# **Настройки VS Code Внутри**

Files: Auto Save -> after Delay включает автосохранение файлов

Editor -> Parameter Hints -> Enabled -> **+++**  включает подсказки при наведении на функции

Workbench -> Status bar -> Visible -> **+++** включает нижнюю панель

Editor -> Format on Save-> **+++** влючает форматирование файла при сохранении

Ctrl + Shift + P -> View: Toggle Minimap -> **---** убираем боковой поиск по коду слева

Editor -> Hover : Delay -> 1500 устанавливает задержку для подсказок в мл. секундах

Terminal -> Integrated -> Cursor Style -> line устаналивает виду курсора в терминале

Explorer -> Compact Folders -> **---** отключаем компактное отображение вложеных пустых папок

# **Разширение VS Code**

**Code Runner** для выполнения любого кода , результт кода показываеться в консоли ->

**Python** для работы с кодом python

**Pylance** подсказки , автодоплнение кода python

**Python Debugger**  отладчик для python

**Pylint** подсказки, автодополнение кода python

**Black Formatter** форматировщик кода python

**Auto Rename Tag** автоматическое закрытые тега html

**Material Icon Theme** новые иконки

**Css Peek** переход на указаный селектор в css файл при нажанити Ctrl + Наведение

# **Настройки Для Разширений VS Code**

Code-runner -> Clear Previous Output -> **+++** отключает отображение предидущего вивода консоли

Code-runner -> Show Execution Message **-> ---** отключает отображение лишеного текста в консоли

Code-runner -> Save all file before Run -> **+++** включает сохранение кода перед выполнением

Python -> Analysis Indexing -> **+++**

Python -> Analysis -> Enable Pytest Support -> **+++**

Python -> Analysis -> Import Format -> absolute

Python -> Analysis -> Type Checking Mode -> off отключает дополнительные предостерижения в коде

Python -> Analysis -> Diagnostic Mode -> workspace

Python -> Analysis -> Auto Search Paths -> **+++**

Python -> Auto Import Completions -> **---** отключает автоимпорт модулей

Python -> Analysis -> Complete Function Parens -> **---**

Explorer -> Auto Reveal -> False отключает автоматическое открытия папок при открытии файла из них

# **settings.json**

Python -> Analysis -> Package Index Depths -> { “name” : “django” , “depth” : 3 , “includeAllSymbols”: true } улучшаем автозаполнение при импортировании модулей в django

|  |
| --- |
| {              "name": "django",              "depth": 3,              "includeAllSymbols": true          }, |

|  |
| --- |
| "workbench.colorCustomizations": {          "editorError.foreground": "#00000000",          "editorWarning.foreground": "#00000000",          "editorInfo.foreground": "#00000000"      }, |

    },

# **Горячие Клавиши VS Code**

**Ctrl + S** сохранить файл

**Shift + Alt + F** форматирование кода

**Tab** допишет конец команды

**Ctrl + -** изменение маштаба редактора

**Ctrl + Shift + P** отрывает глобальный поиск

**Ctrl + B** открывает и закрывает боковую панель слева

**Ctrl + Shift + Space** подскажет подсказку к команде

**Shift + Alt + up/down** дублирование строки вверх или вниз

**Ctrl + L** виделить текущую строку

**Ctrl + Shift + K** удалить текущую строку

**Ctrl + J** открывает и закрывает панель с терминалом

**Ctrl + ?/** коментирует или розкоментирует выделенный участок кода

**Alt + up/down** перемещение вибраной строки вверх/вниз

**Ctrl + D**  виделяет весь указаный текст с начала и до конца

**Alt + mouse click** редактование сразу нескольких вибранных мишкою строк

**Ctrl + F2** вибирает все одинаковые названия в файле

**Alt + ~(тильда)** выполнение кода

**Ctrl + ~(тильда)** отрытие/закрытие терминала

**Ctrl + D** выход из образа docker

# **Django VS Code**

**Python: Select Interpreter ->** Интерпритатор python, который находиться в виртуальной среде проекта

**Run -> Add Configurations** автоматический запуск сервера

# **Настройка pgAdmin - PostgreSQL**

File -> Preferences -> Query Tool -> Editor -> Font Size изменение шрифта в редакторе ( 1.2 )

File -> Preferences -> Pinary Paths -> PostgreSQL 16 -> путь до програми и папки bin добавляет возможность делать backups базы данных

# **Сохранение Базы Данных в pgAdmin**

**Сохранение в формате кода SQL ==>**

Backup ->

my\_database.sql

Format = Plain ,

Query Options = Use Insert Command (вкл) ,

Table Options = Use Column Inserts (вкл)

**Сохранение в специальном формате бинарного файла бинарного файла ==>**

Backup ->

my\_database.backup

Format = Custom

# **Востановление Базы из Сохранений**

**Востановление Базы, если сохранение в формате .backup ==>**

Создаем Пустую Базу Данных -> Restore -> File Name my\_database.backup

**Востановление Базы, если сохранение в формате .sql ==>**

Создаем Пустую Базу Данных -> Создание Запроса -> Открыть файл my\_database.sql -> Выполнить