



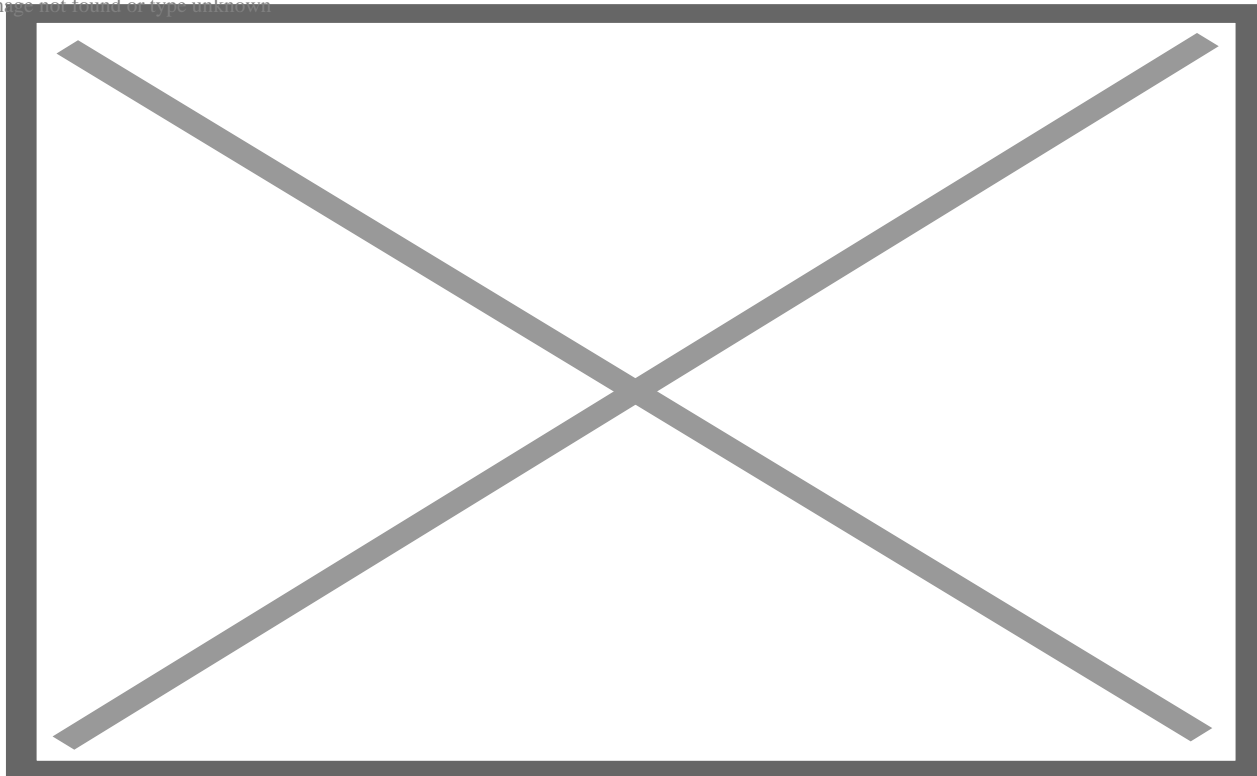
## Как создавать и запускать сценарии PowerShell: Пошаговое руководство

### Описание

Работа со сценариями PowerShell может быть захватывающей перспективой. В конце концов, он предоставляет вам чистый интерфейс для создания сценариев автоматизации повседневных задач. Если вы разработчик, то вы можете создавать полноценные сценарии автоматизации – PowerShell отлично справляется с этой задачей, особенно если вы используете его для создания, тестирования и развертывания решений в средах CI/CD. В этой статье мы подробнее рассмотрим создание и запуск сценариев PowerShell. Здесь мы рассмотрим пошаговое руководство по выполнению этих действий с помощью VS Code, блокнота и интегрированной среды сценариев. Давайте начнем.

### Что такое PowerShell?

Image not found or type unknown



PowerShell – это кроссплатформенный инструмент автоматизации и настройки. Он предлагает полный пакет для администраторов, разработчиков и других пользователей для использования командлетов (небольших и легких команд) для управления различными задачами.

В рамках пакета вы получаете следующее:

- Оболочку командной строки
- Соответствующий язык сценариев
- фреймворк для обработки команд

В отличие от Command Prompt, он доступен на всех основных платформах, включая Windows, Linux и macOS. Технически PowerShell построен поверх фреймворка .NET. Благодаря его подходу, он поддерживает объекты и структурированные данные, такие как JSON, CSV и XML. Разработчики также могут использовать REST API с PowerShell. К сожалению, базовая Командная строка не поддерживает ничего из этого, поскольку она является текстовой. Все сценарии PowerShell записываются в расширении .ps1.

При написании сценариев в PowerShell необходимо знать три ключевых понятия:

- **Cmdlet:** Это атомарный исполняемый сценарий, который предлагает предварительно настроенную функциональность. Например, вы можете использовать Copy-Item для копирования, Get-Help для получения справки или Write-Host для вывода чего-либо на экран.
- **Псевдонимы:** Псевдонимы обеспечивают простой способ ссылки на команды. Это обеспечивает более быстрый доступ. Однако рекомендуется использовать их с подсказками для интерактивных целей, а не при написании сценариев PowerShell.
- **Функции:** Функция – это набор операторов PowerShell. Пользователи могут вызвать функцию, обратившись к ней.

## PowerShell и создание сценариев

Сценарии – это важный метод автоматизации различных аспектов вашей работы. Будь вы программист, администратор или фрилансер, сценарии могут автоматизировать рутинные рабочие процессы.

В конце концов, у сценариев есть множество преимуществ, включая:

- Экономия драгоценного времени благодаря автоматизации.
- Переносимые знания.
- Позволяют сохранять единообразие в разных системах.
- Повышает вашу квалификацию и расширяет возможности трудоустройства.

Вы можете использовать PowerShell, Bash Scripting, Ruby или Python. В целом, PowerShell – отличный инструмент для написания сценариев. Однако Windows не позволяет запускать сценарии из-за настроек безопасности. Это может привести к ошибкам типа “Запуск сценариев отключен в этой системе”. Чтобы обойти эту проблему, вам нужно установить правильную политику выполнения. (Об этом рассказывается далее в статье).

## Почему вы должны использовать PowerShell для выполнения сценариев?

Одна из главных причин, по которой PowerShell является отличным выбором для написания сценариев, – это его способность работать на разных платформах. Она поддерживает Windows, Linux и MacOS, чего не делает стандартная командная

---

строка Windows. Вы можете использовать PowerShell для создания простых заданий с помощью простых конструкторов задач. Например, вы можете использовать PowerShell для перезагрузки машины, периодической очистки кэша или даже аутентификации сеансов. Кроме того, сценарии PowerShell в основном используются для автоматизации управления системой. Однако вы можете использовать его как разработчик для тестирования, сборки и развертывания сборок через среду CI/CD.

## Как создаются сценарии PowerShell?

Для создания сценариев PowerShell вам понадобится текстовый редактор. По большей части вы можете обойтись обычным текстовым редактором, но с современным редактором текста/кода весь процесс будет намного проще. Поэтому мы начнем с демонстрации того, как создавать сценарии PowerShell с помощью Visual Studio Code (также известного как VS Code). Мы также рассмотрим, как создавать сценарии PowerShell в Блокноте и других удобных для создания сценариев интегрированных средах, таких как PowerShell Integrated Scripting Environment. (PowerShell ISE).

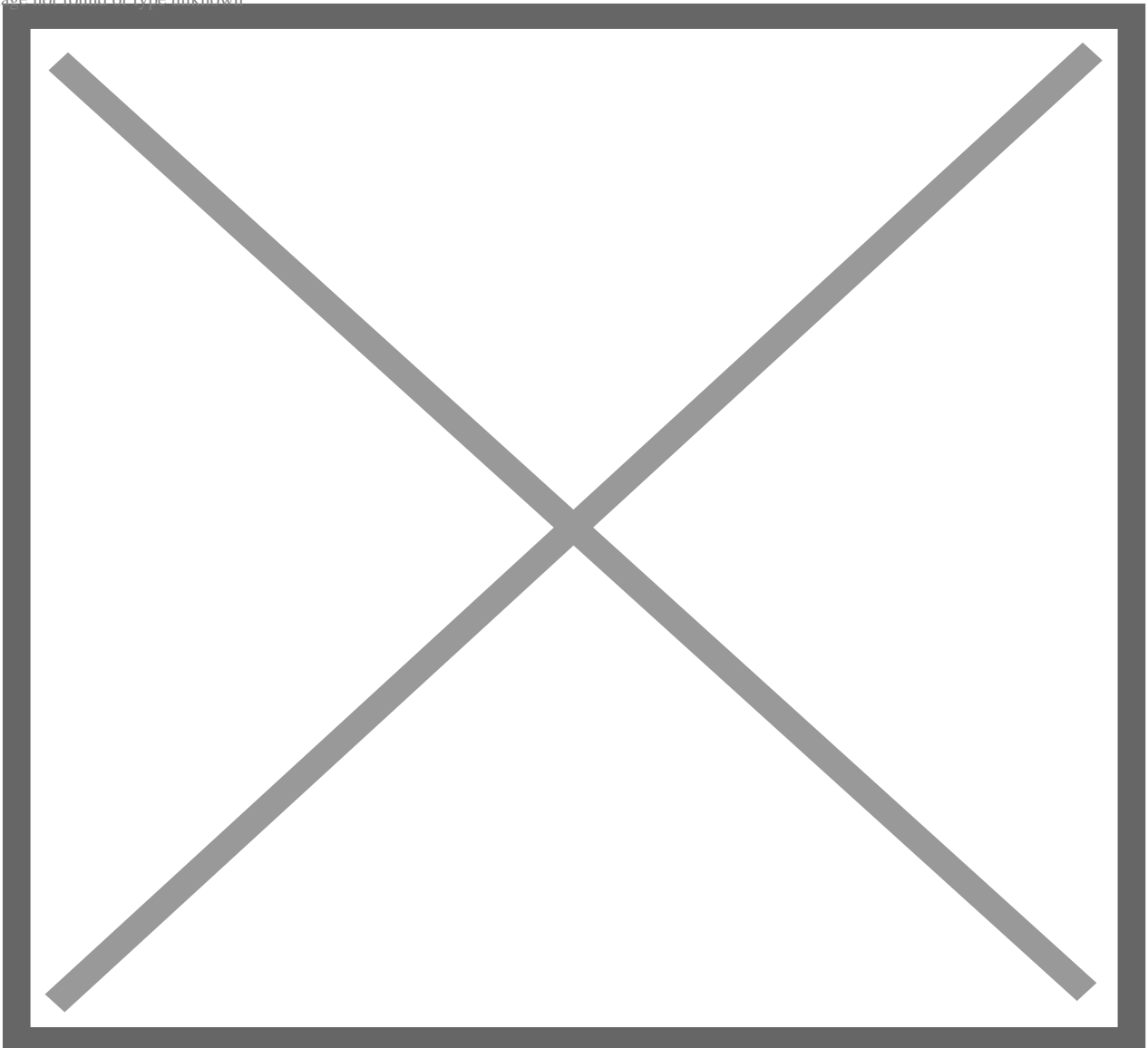
### С VS Code

VS Code – это прочный и надежный бесплатный кроссплатформенный редактор кода. Его разработкой занимается компания Microsoft, что дает вам дополнительное преимущество в виде хорошей поддержки PowerShell. Вы получаете чистый интерфейс из коробки и можете использовать его практически с любым языком программирования. Поскольку VS Code – это настоящий редактор кода, он поддерживает расширения. И для улучшения работы со сценариями PowerShell мы будем использовать расширение PowerShell.

### Установка Visual Studio Code

Но прежде чем приступить к работе, необходимо установить VS Code. Для этого перейдите на страницу загрузки Visual Studio Code. Выберите правильную версию VS Code в зависимости от вашей операционной системы. Поскольку мы используем Windows 11, мы выбрали версию для Windows.

Image not found or type unknown

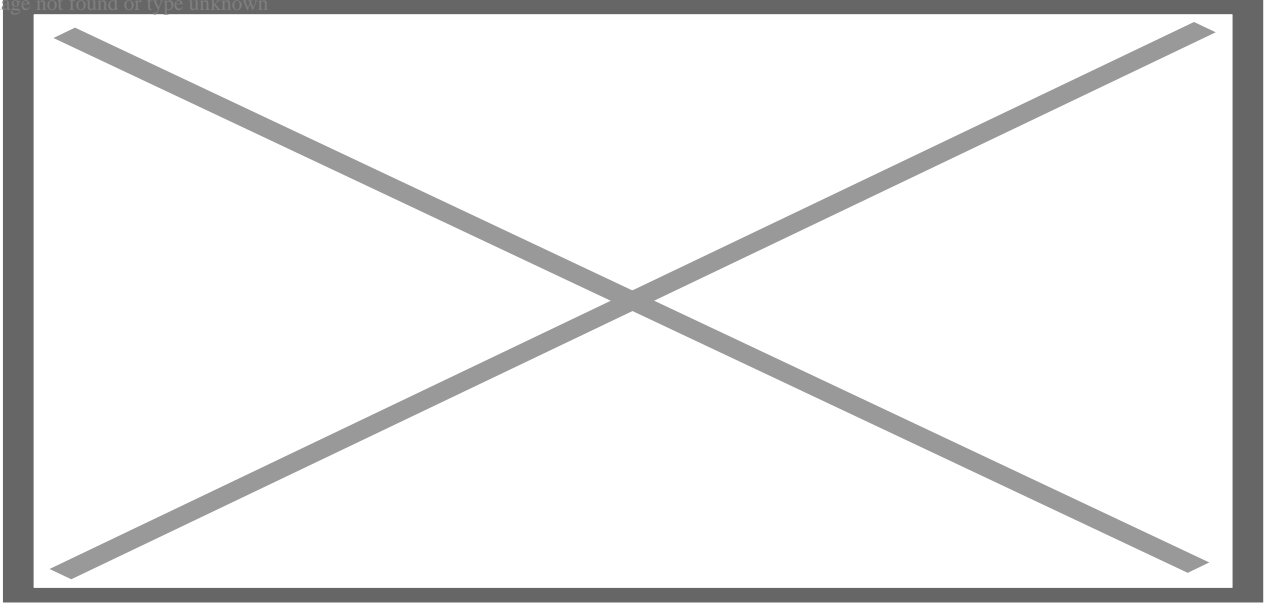


После загрузки дважды щелкните программу установки и следуйте указаниям мастера, чтобы установить ее на свой компьютер.

### **Установка расширения PowerShell**

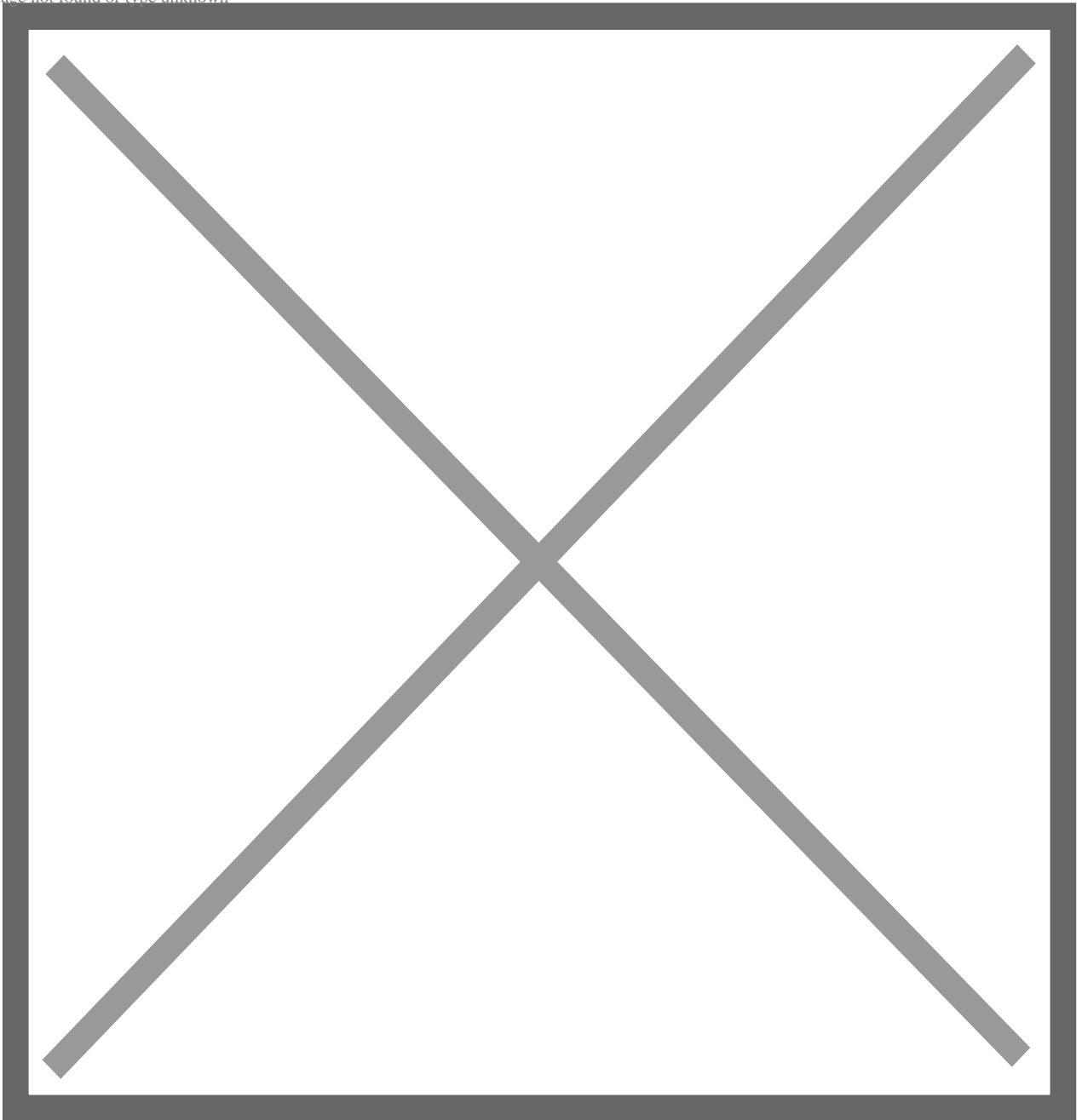
Расширение PowerShell можно установить двумя способами. Во-первых, создайте новый файл с расширением .ps1. VSCode достаточно умен, чтобы понять, что вы работаете с PowerShell, и автоматически рекомендует вам установить “расширение PowerShell” от Microsoft.

Image not found or type unknown



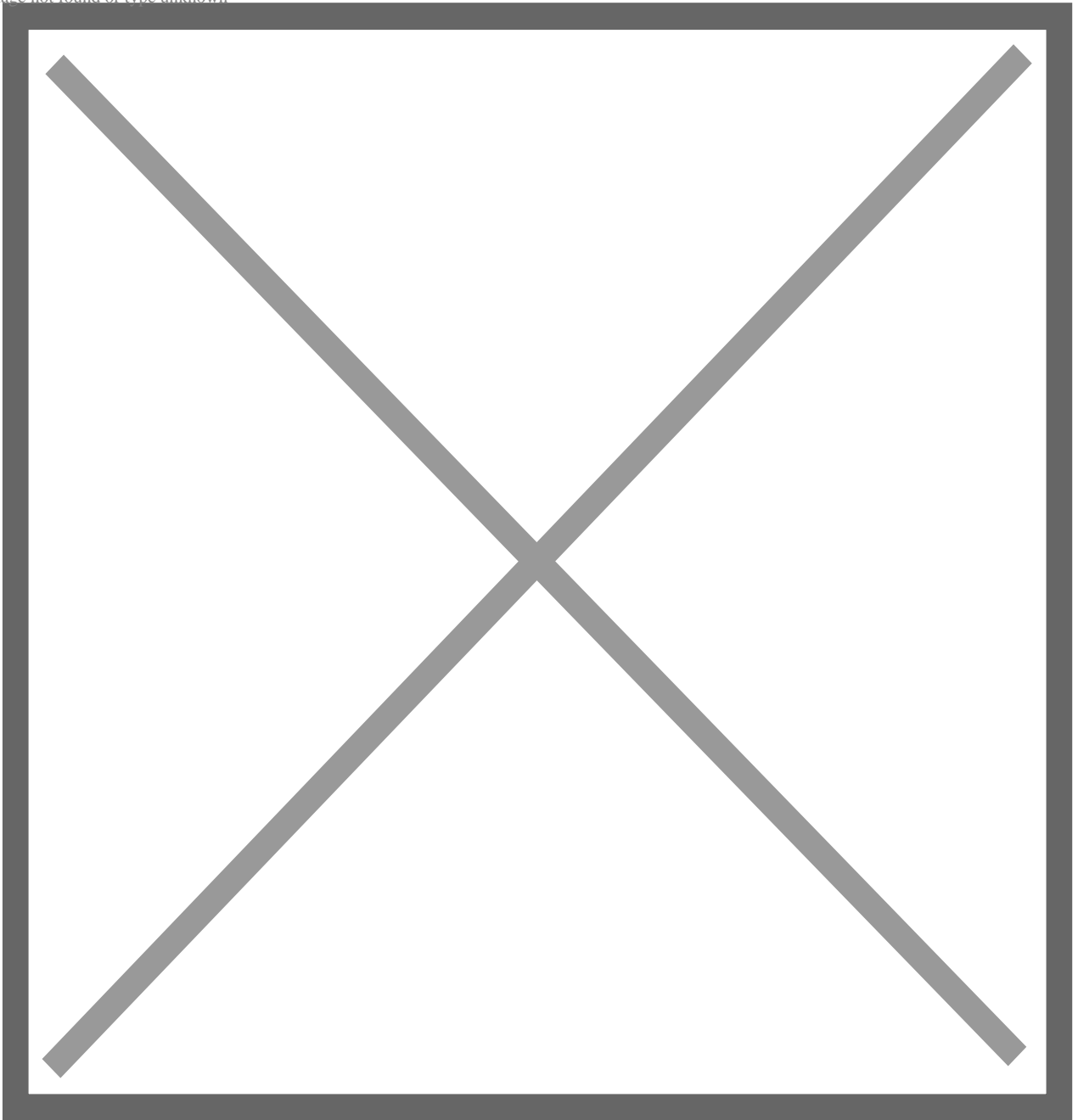
Вы можете нажать “Установить”, чтобы установить его. В противном случае нажмите “Показать рекомендацию”, чтобы узнать о нем больше.

Image not found or type unknown



Если по какой-то причине Visual Studio Code не дал вам никаких рекомендаций, вы можете вручную найти плагин. Нажмите на символ четырех квадратов (с одним вылетевшим квадратом) или нажмите `Ctrl + Shift + X`, чтобы открыть окно расширения. Теперь найдите “PowerShell от Microsoft” и нажмите на первый результат. Оттуда перейдите к пункту “Установить”.

Image not found or type unknown



Наконец, он спросит вас, хотите ли вы доверять авторам файлов в рабочей области. Нажмите “Trust Workspace & Install”, чтобы завершить установку.

### **Создание сценария PowerShell с помощью VS Code**

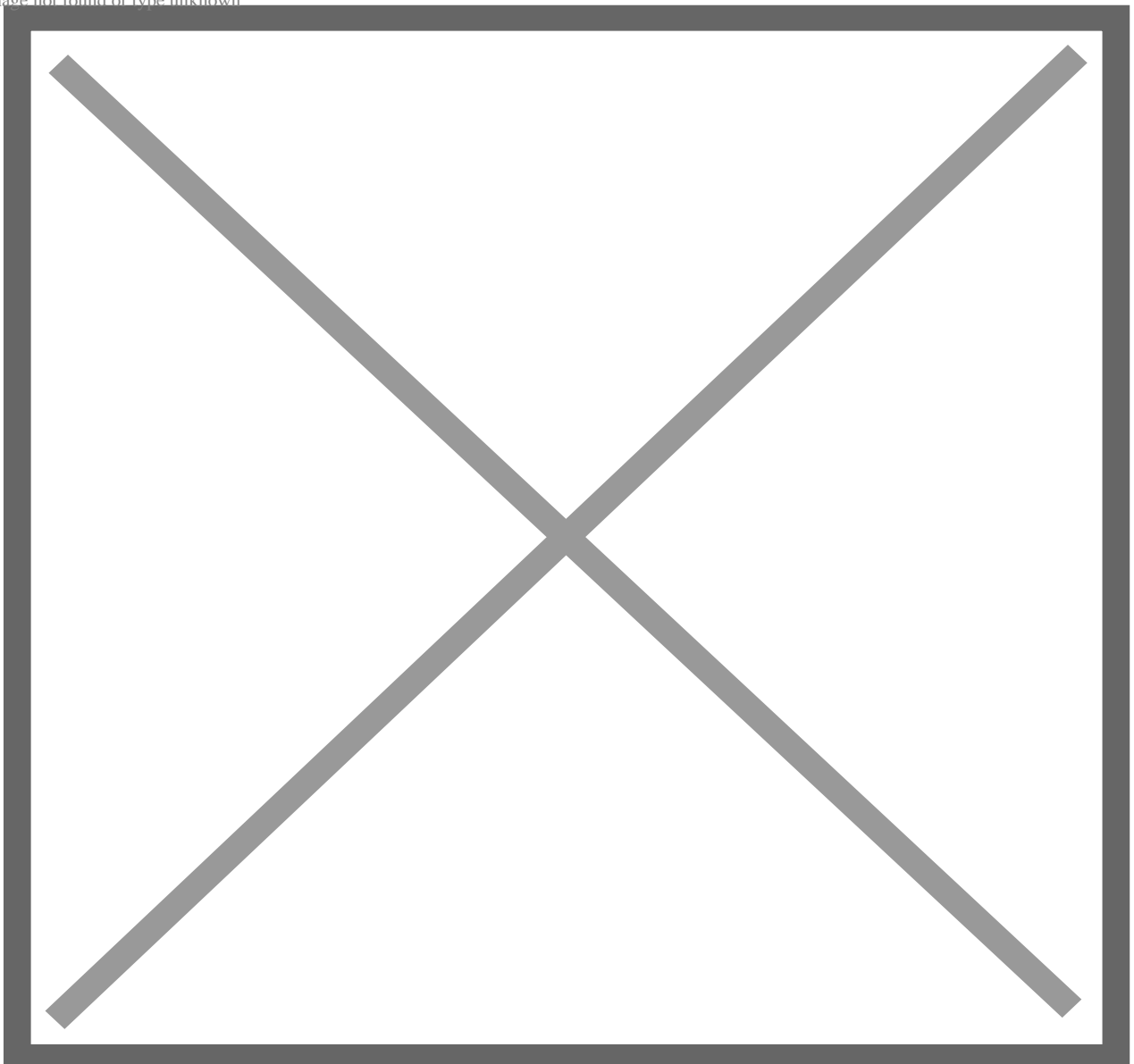
Теперь вы можете создать свой первый сценарий Powershell с установленным VS Code и его расширениями.



Если вы еще не создали новый файл, вы можете сделать это, выполнив следующие шаги.

1. Откройте VS Code
2. Перейдите в верхнее меню, нажмите Файл, а затем Новый текстовый файл.
3. Теперь сохраните файл, перейдя в меню Файл > Сохранить как.
4. Введите желаемое имя файла и не забудьте добавить расширение .ps1.
5. Наконец, нажмите на кнопку Сохранить файл.

Image not found or type unknown



Вот и все! Теперь вы можете создать свой сценарий соответствующим образом. Я добавил команду Write-Host со значением "This is a VS Code Script written for

Geekflare”. Мы увидим, как выполнить сценарий позже, так что продолжайте читать!

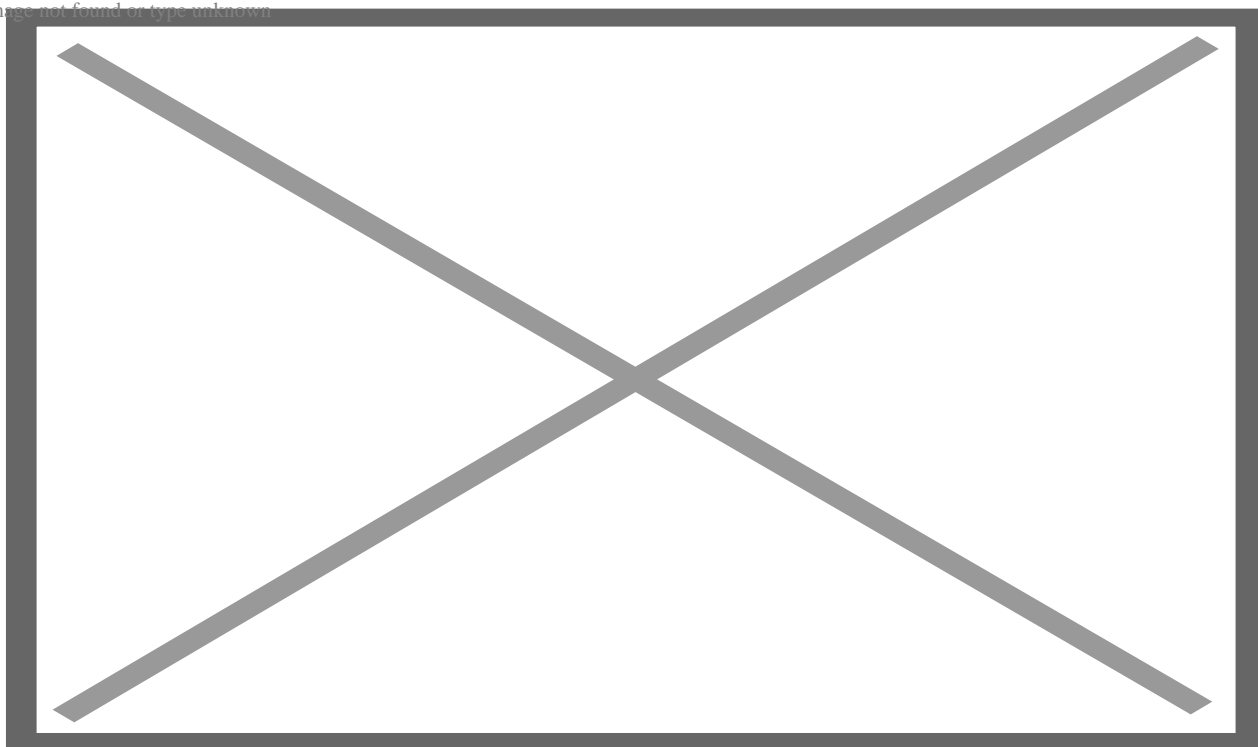
## **С помощью Блокнота**

Создавать сценарии PowerShell в Блокноте очень просто.

Все, что вам нужно сделать, это выполнить следующие шаги:

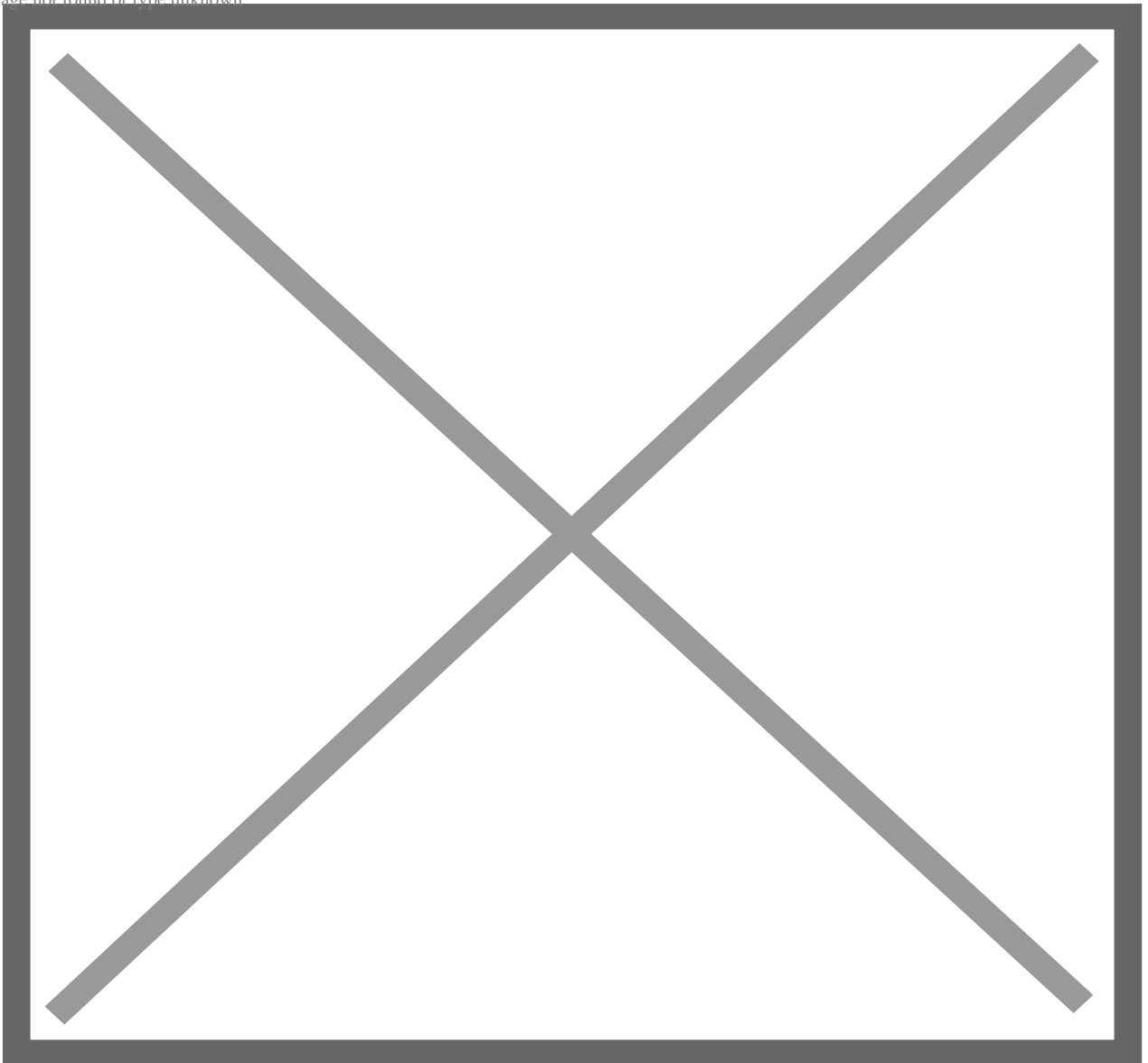
- 1.Откройте “Пуск” и найдите “Блокнот”.
- 2.Щелкните по нему правой кнопкой мыши и выберите “Открыть от имени администратора”.
- 3.Теперь напишите свой скрипт.

Image not found or type unknown



- 4.Для учебника мы напомним: “Это сценарий PowerShell, написанный в Блокноте”.
- 5.Перейдите в Файл в верхнем меню и нажмите “Сохранить как”.

Image not found or type unknown



6. Выберите имя вашего сценария PowerShell и нажмите “Сохранить”.

Вот и все. Вы успешно создали сценарий PowerShell с помощью Блокнота.

## **С помощью PowerShell ISE**

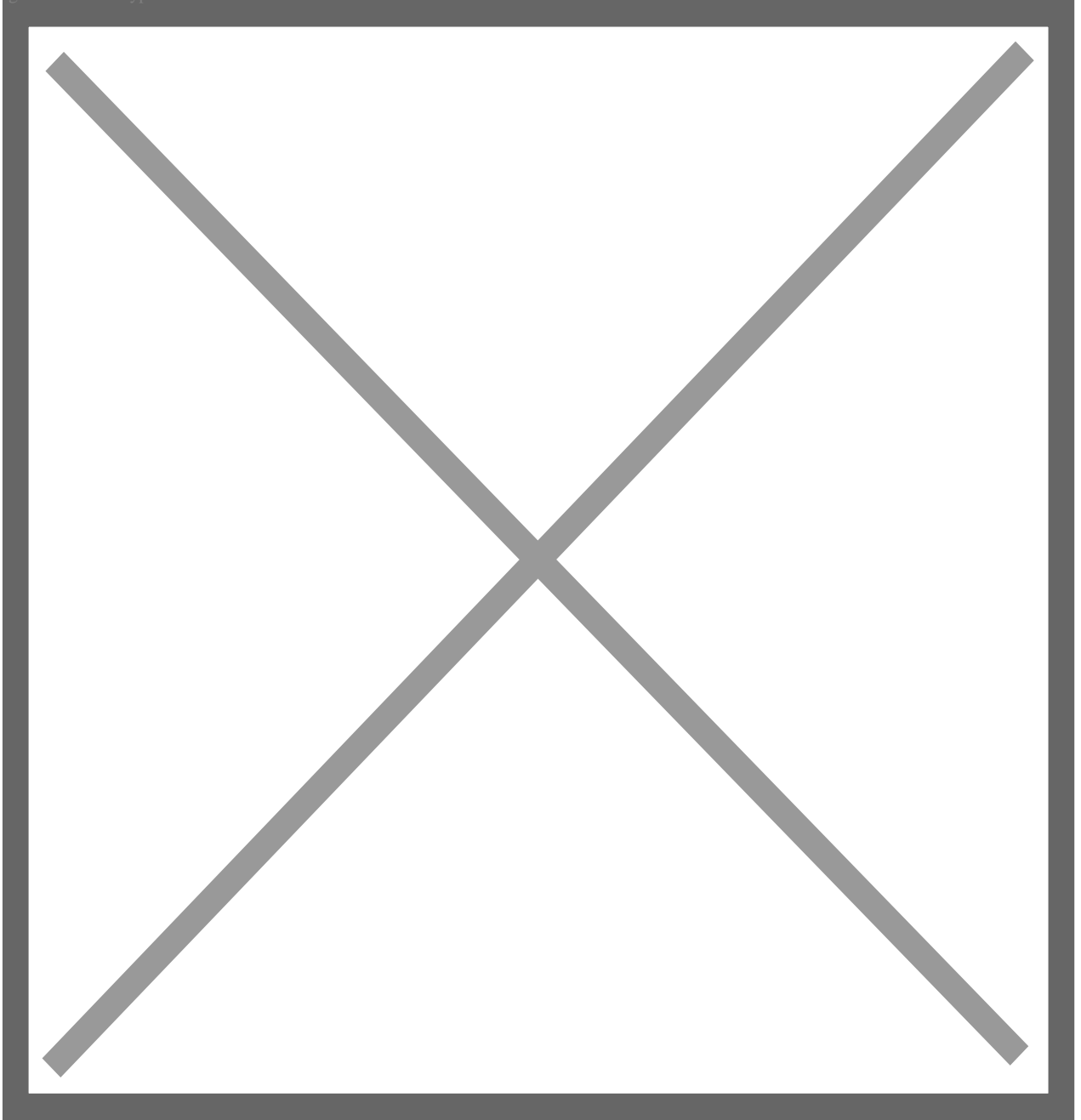
PowerShell в Windows также предлагает свою собственную интегрированную среду сценариев. Вы можете использовать ее для создания своих сценариев, не прибегая к помощи сторонних редакторов кода или текстовых редакторов.

Для успешного использования вам потребуется выполнить следующие шаги:

1. Откройте Пуск.

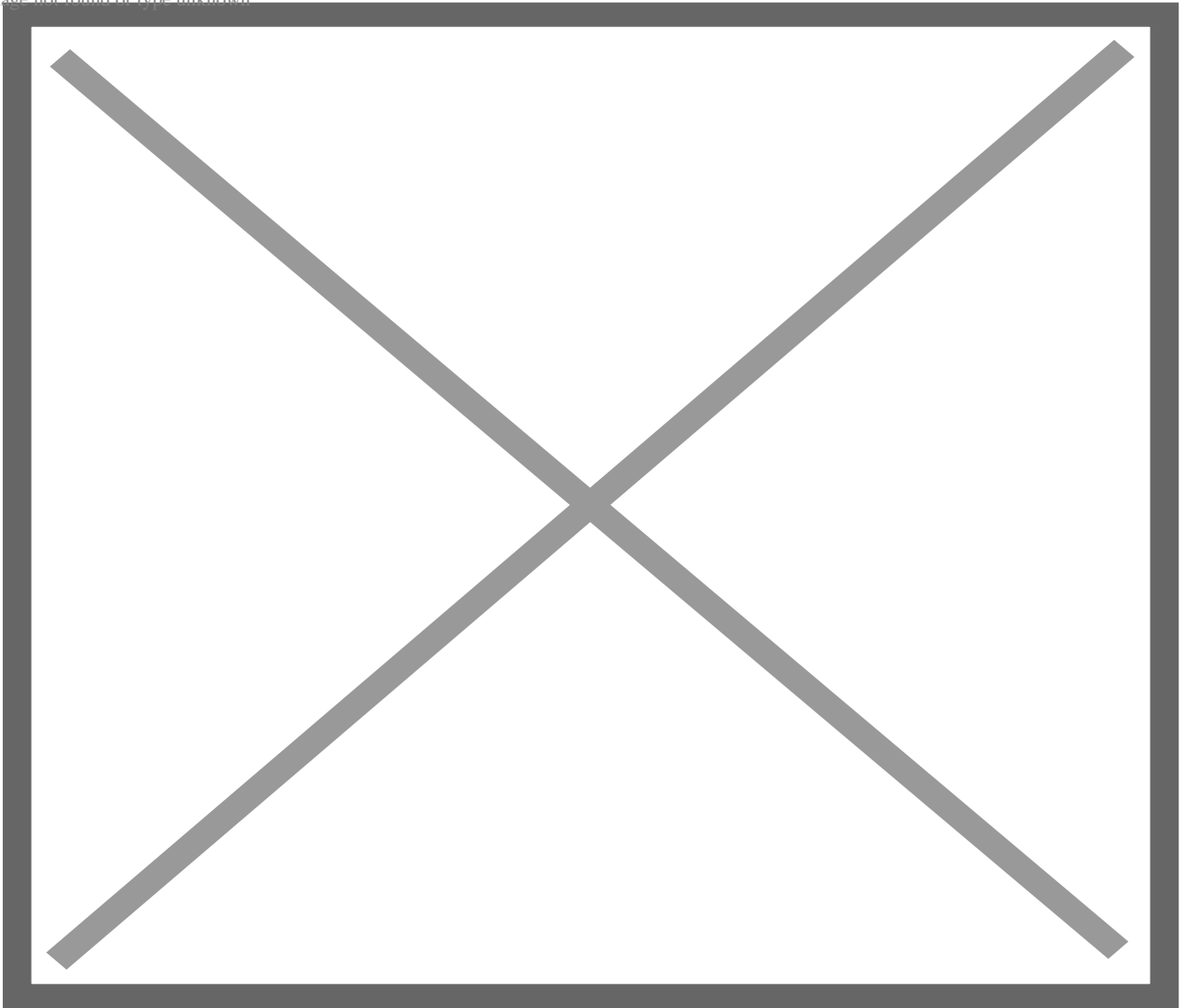
2. Теперь найдите ключевое слово “Windows PowerShell ISE”.

Image not found or type unknown



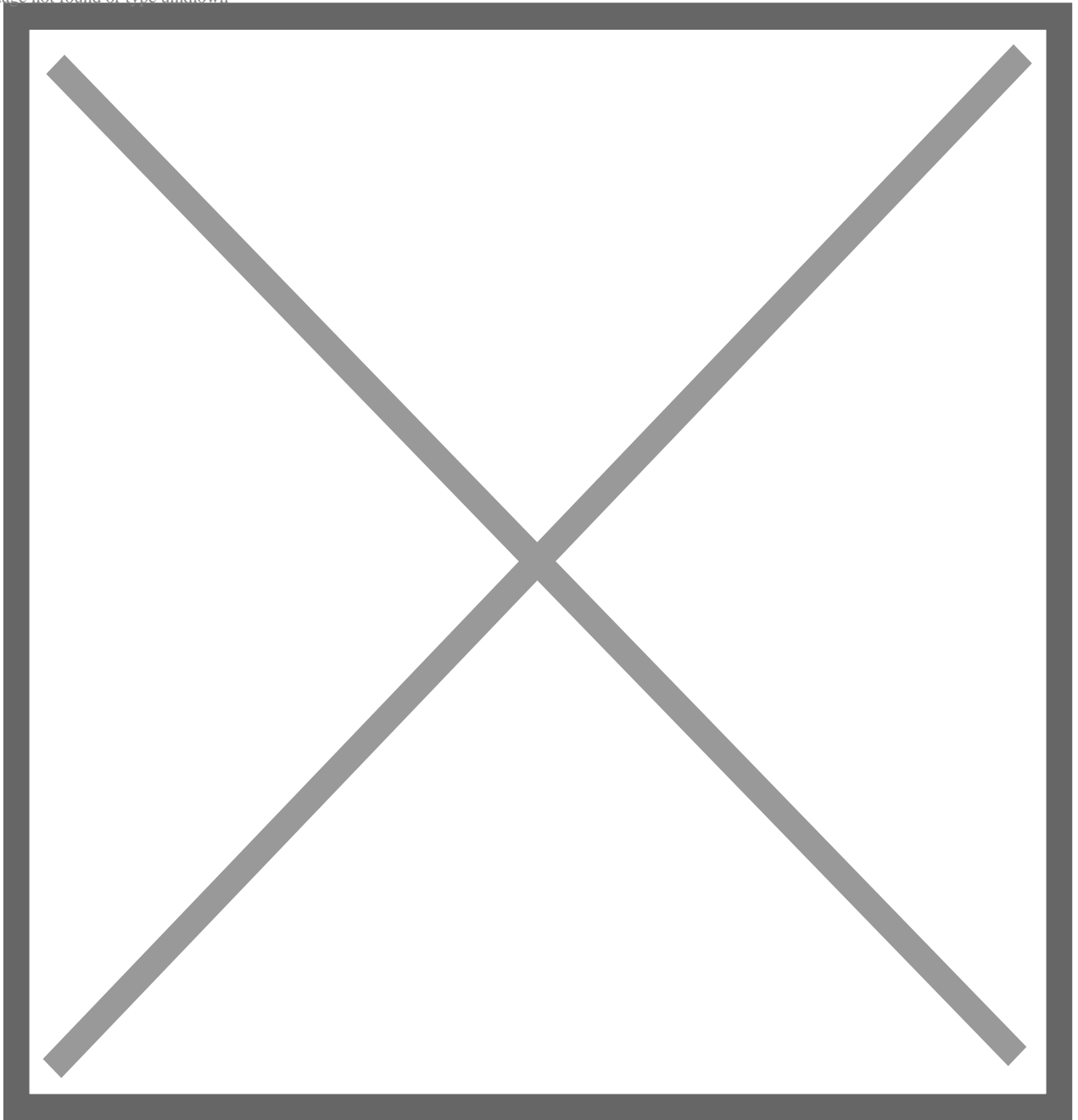
3. Выберите первый результат, щелкните правой кнопкой мыши, а затем левой кнопкой “Запуск от имени администратора”. Откроется окно PowerShell ISE.

Image not found or type unknown



4. Теперь нажмите на Файл и выберите “Новый”. Откроется новое окно файла, как показано ниже.

Image not found or type unknown



7. Введите желаемый сценарий. В данном учебнике мы напишем: “Это сценарий PowerShell, написанный с помощью PowerShell ISE”.

8. После этого перейдите в меню Файл и Сохранить как.

9. Здесь выберите имя файла и нажмите кнопку Сохранить.

Вы также можете запустить сценарий непосредственно из PowerShell ISE. Для этого нажмите кнопку RUN в верхней части среды. Если при этом возникает ошибка

безопасности, читайте ниже, как ее решить.

## Как устранить ошибку безопасности сценария PowerShell и запустить его

Как упоминалось ранее, Windows имеет строгую политику безопасности при выполнении сценариев. Чтобы обойти ее, вам нужно разрешить выполнение сценариев на машине Windows.

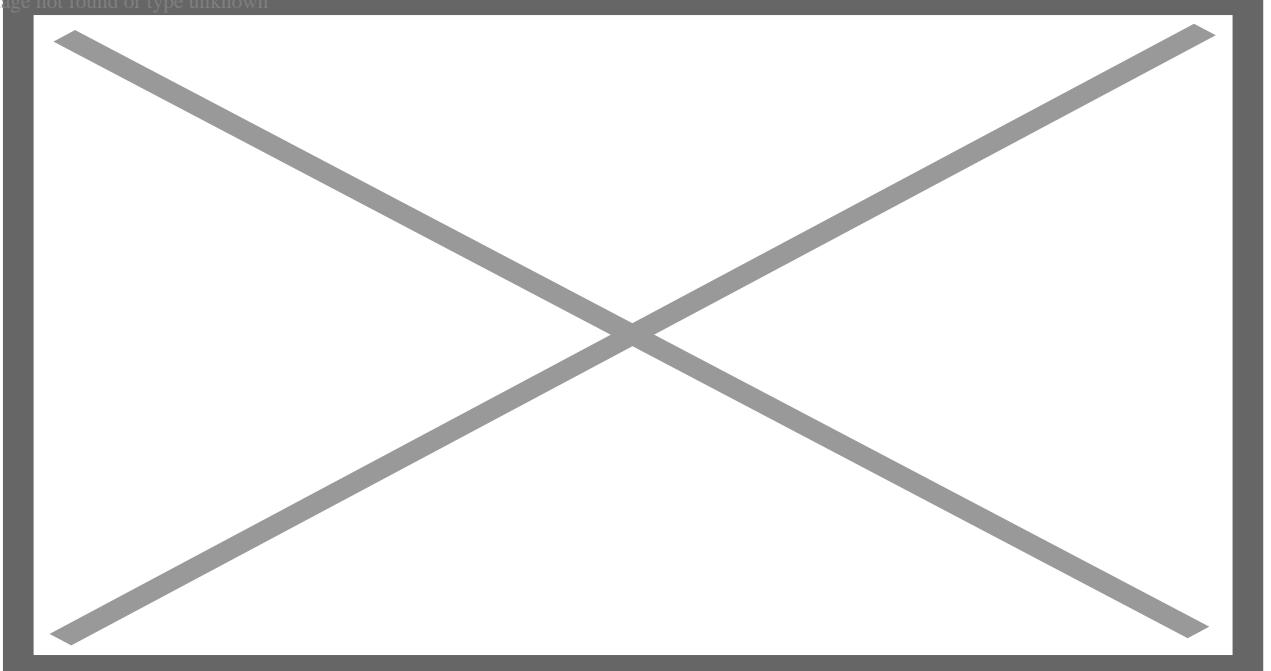
Для этого выполните следующие действия:

1. Откройте меню Пуск Windows.
2. Найдите PowerShell.
3. Теперь щелкните правой кнопкой мыши на приложении PowerShell и нажмите на него.
4. Выберите опцию “Запуск от имени администратора”.
5. Когда откроется PowerShell, введите команду “Set-ExecutionPolicy RemoteSigned”.
6. Нажмите Enter, после чего появится запрос на подтверждение. Нажмите A, чтобы включить его.

Вот и все; теперь вы наконец-то можете запускать все свои сценарии!

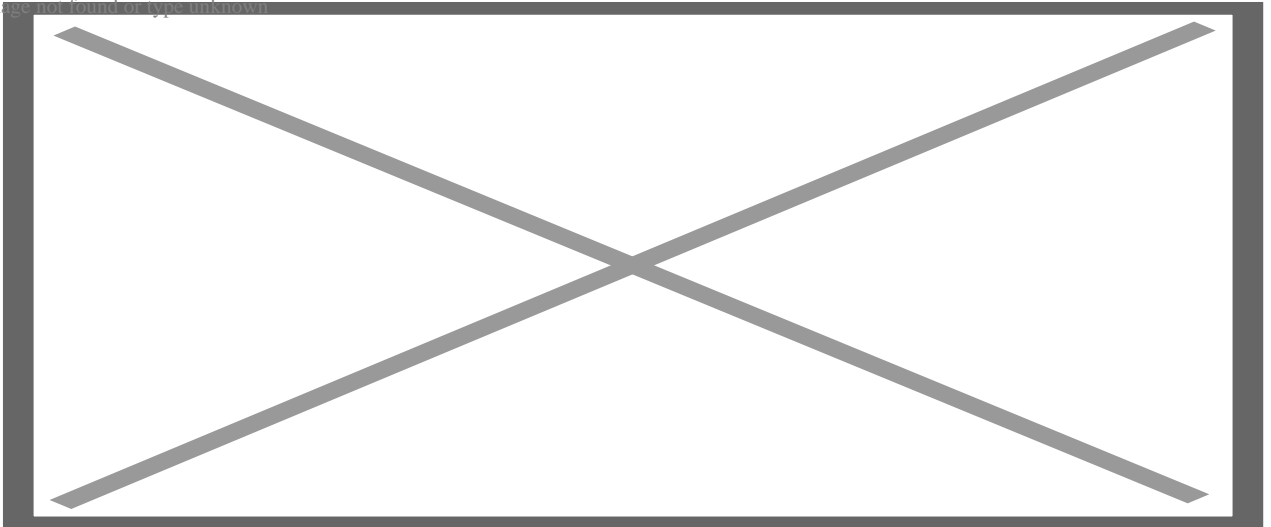
Для PowerShell ISE это выглядит следующим образом.

Image not found or type unknown



Для Visual Studio Code это будет выглядеть следующим образом.

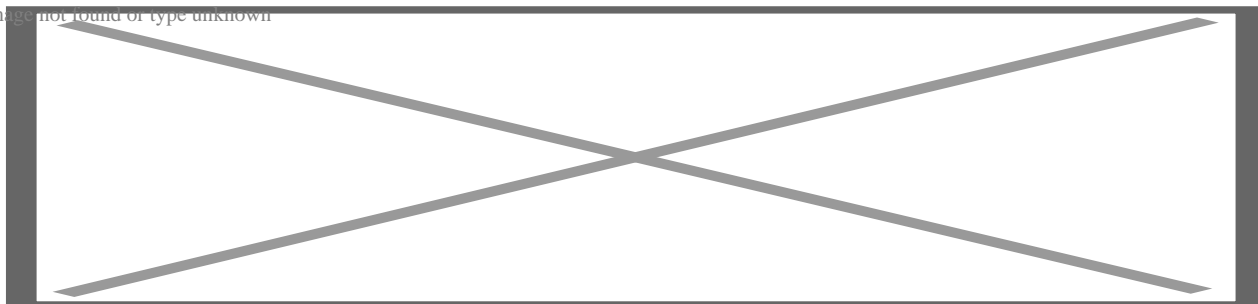
Image not found or type unknown



А для Notepad вам нужно будет запустить его непосредственно из командной строки PowerShell.



Image not found of type unknown



## Лучшие практики написания и выполнения сценариев PowerShell

При работе со сценариями PowerShell следует придерживаться лучших практик. К ним относятся:

- Используйте полное имя команды, а не ее псевдоним. Псевдоним удобен только при взаимодействии с консолью. Псевдоним повышает скорость взаимодействия, но может испортить сценарий.
- Лучше не использовать команду Write-Host, поскольку она обладает ограниченной функциональностью. Например, она только отправляет вывод на экран. Однако ее можно использовать для отображения текста определенным цветом. Вместо этого вы можете использовать команду Write-Output.
- Всегда используйте существительное команды в единственном числе, а не во множественном.
- Используйте утвержденный глагол для команды.
- Указывайте расширение приложения всегда, когда это возможно. Это устраняет путаницу и улучшает читаемость.
- Используйте специальный редактор кода PowerShell, такой как VS Code или PowerShell ISE. Они предлагают отличные возможности для написания и выполнения сценариев PowerShell. Вы можете использовать Notepad или Notepad++, но тогда вы ограничиваете себя.
- Старайтесь комментировать, но не переусердствуйте, так как это может испортить читаемость и сопровождаемость кода.
- Убедитесь, что вы установили правильный параметр безопасности для запуска сценариев PowerShell.

## **Заключение**

Вот и все! Теперь вы можете создавать и запускать сценарии PowerShell с помощью блокнота, VS Code и PowerShell ISE. Кроме того, вы можете использовать любой из перечисленных инструментов для написания и выполнения сценариев PowerShell. Однако, как уже говорилось, лучше избегать использования Notepad или Notepad++ для написания сценариев PowerShell, так как вы упускаете многие возможности, предлагаемые специальными редакторами кода VS Code и PowerShell ISE.

### **Дата Создания**

22.05.2023