

Введение в Python

№ урока: 1 Курс: Python Starter

Средства обучения: PyCharm

Обзор, цель и назначение урока

После завершения урока обучающиеся будут иметь представление об особенностях языка программирования Python, истории его появления, достоинствах и недостатках, областях применения, научатся использовать интерпретатор и создадут свою первую программу на этом языке.

Изучив материал данного занятия, учащийся сможет:

- Использовать интерпретатор языка Python для запуска скриптов
- Использовать интерпретатор языка Python в интерактивном режиме
- Создавать проекты и запускать код в интегрированной среде разработки

Содержание урока

1. Общие сведения
2. История языка
3. Применение
4. Преимущества и недостатки
5. Установка Python
6. Создание простейших программ на Python

Резюме

Язык программирования Python — это инструмент для создания программ самого разнообразного назначения. Язык с низким порогом вхождения, что делает его отличным выбором в качестве первого языка.

Язык программирования — это формальная знаковая система, что используется для написания компьютерных программ. Язык определяет набор лексических, синтаксических и семантических правил. Это позволяет чётко определить, как нам следует общаться с компьютером.

Python - это высокоуровневый язык программирования. Разработчики языка сделали упор на читабельности и удобстве написания и кода. Сейчас он используется во всех IT отраслях.

История языка

Python разрабатывался с конца 80-х. Его создатель, Гвидо ван Россум, назвал язык в честь британского телешоу “Летающий цирк Монти Пайтона”, а не потому что любил змей.

В то время Россум работал в голландском институте CWI. Создавал Python для студентов, так как языки того времени были сложными для студентов. Взяв лучшие возможности из других языков программирования, Python был представлен в феврале 1991 года.

В 2018 году Гвидо ушел в “постоянный отпуск” из проекта Python, оставив за собой право быть обычным разработчиком. Сейчас Гвидо ван Россум работает в DropBox и среди разработчиков Python известен как “великодушный пожизненный диктатор” проекта, что значит, что он продолжает наблюдать за процессом разработки Python.

Python имел две актуальные ветки: 2.x, 3.x. В начале 2020 года поддержка Python 2.x прекратилась. Новые проекты следует создавать на Python 3 - это гарантирует наличие актуальных библиотек и исправление старых ошибок интерпретатора

Применение

Python применяется для создания скриптов и утилит. Широко известен в научной сфере и используется для анализа данных. Иногда Python можно увидеть среди языков для создания игр. В нашем курсе мы будем рассматривать Python, как язык для создания веб приложений.

Вот ряд компаний, что успешно применили Python в своих продуктах: Dropbox, Blender, Mercurial, Instagram, Pinterest, EVE Online, World of Tanks, Civilisation IV.

К сожалению, Python нельзя применять для низкоуровневого программирования. Это значит, что драйверы на нём не написать.

Особенности

Среди особенностей Python можно выделить следующие: * простота синтаксиса и изучения * поддержка разных парадигм программирования * интерпретируемый * динамическая типизация * открытость * кроссплатформенность * существование альтернативных реализаций, которые решают определённые проблемы (Stackless Python, PyPy) или интегрируют его в определённую платформу (IronPython для .NET, Jython для Java)

Создатель Python - опытный программист. Он постарался устранить все недостатки существовавших тогда языков. Ему это удалось. Python может похвастаться: * простотой и читабельностью * большой стандартной библиотекой и огромным выбором сторонних библиотеки и модулей * краткостью кода и экономией времени разработчика * возможностью связывания с кодом на других языках

Но не обошлось без недостатков: * низкая по сравнению с компилируемыми языками скорость работы * GIL

Чтобы приступить к изучению Python, необходимо выучить понятия трансляции, компиляции и интерпретации.

Трансляция — преобразование программы, что написана на одном языке программирования, в программу на другом языке.

Компиляция — трансляция программы, что написана на языке высокого уровня, в эквивалентную программу на низкоуровневом языке, близком машинному коду.

Интерпретация — построчный анализ, обработка и выполнение кода. Python является интерпретируемым языком.

Интерпретатор — программа (разновидность транслятора), выполняющая интерпретацию.

Установка Python и IDE

Чтобы выполнить наш код, надо установить программу-переводчик (интерпретатор), которая будет объяснять компьютеру, что значит наш код.

Интерпретатор можно скачать с официального сайта языка: <https://python.org>. Выберите подходящий для своей системы установщик и следуйте инструкциям.

Для комфортной разработки, нам понадобится **IDE — интегрированная среда разработки**. Другими словами, это система программных средств, используемая программистами для разработки программного обеспечения. Среда разработки обязательно включает в себя: текстовый редактор * компилятор и/или интерпретатор или средства интеграции с ним * средства автоматизации сборки * отладчик

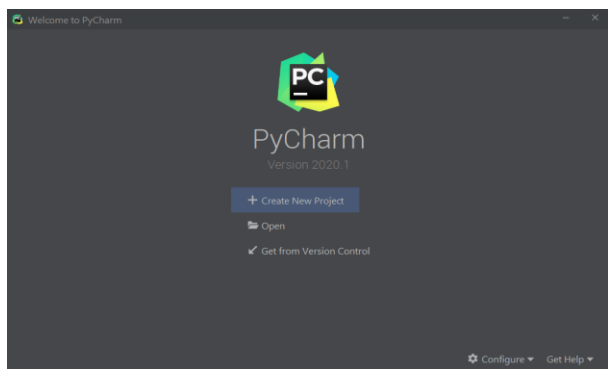
Также IDE может включать: * средства интеграции с системами контроля версий * инструменты конструирования графического интерфейса пользователя

Официальный сайт IDE PyCharm (рекомендуется): <http://www.jetbrains.com/pycharm/PyCharm>

Практика

Давайте создадим наш первый Python проект. В примерах будет использоваться среда разработки PyCharm.

В стартовом диалоге необходимо выбрать "Create New Project", выбрать тип проекта "Pure Python" и указать путь к папке, где он будет храниться.



Стартовое диалоговое окно

Затем создадим первый файл исходного кода. Для этого нужно выбрать команду меню File → New, нажать правую кнопку мыши в дереве проекта слева и выбрать “New”, после чего выбрать Python File и ввести его имя.

Программа на Python, которая выводит приветствие “Hello, World!”:

```
print("Hello, World!")
```

Hello, World!

Здесь *print* – функция, а “Hello, World!” – строка.

Более подробно эти понятия будут рассмотрены на следующих уроках. Для первого запуска скрипта нужно нажать Alt+Shift+F10 и выбрать из списка нужный файл исходного кода. В последующие разы просто Shift+F10.

Можно вывести несколько значений:

```
print("first", "second")
```

first second

И [присваивать значениям имена](#):

```
name = "Alex"
```

```
print(name)
```

Alex

Также можно [ввести значения с клавиатуры](#):

```
text = input("Enter some text : ")
```

```
print(text)
```

Enter some text : CyberBionic Systematics

CyberBionic Systematics

Любой файл исходного кода на Python – это обычный текстовый файл с расширением.py. Скрипты можно запускать при помощи командной строки: `python file_name.py`.

Для экспериментов и обучения удобно использовать интерпретатор в интерактивном режиме. Для этого можно открыть командой “python” интерпретатора командной строке или, что более удобно, воспользоваться графическим интерфейсом к нему, запустив приложение IDLE.

Закрепление материала

- Что такое язык программирования?
- Что такое интерпретация?

- Что такое компиляция?
- Язык программирования Python интерпретируемый или компилируемый?
- Для чего используют язык программирования Python? Для чего его нельзя использовать?
- Какие преимущества и недостатки Python вы знаете?
- Что такое интегрированная среда разработки?

Дополнительное задание

Задание

Создайте новый проект в интегрированной среде разработки PyCharm. Создайте файл исходного кода и напишите программу, которая выводит ваше имя. Запустите его. Создайте второй файл с кодом и напишите программу, которая спрашивает у пользователя, как его зовут, и здоровается с ним. Запустите его. Переключитесь на первый скрипт и запустите его. Переключитесь обратно на второй скрипт.

Самостоятельная деятельность учащегося

Задание 1

Установите Python и PyCharm (или другую IDE с поддержкой Python, которая вам удобна). Исследуйте настройки, настройте среду разработки для себя: выберите цветовую схему и шрифт редактора, которые вам нравятся, включите или отключите отображение номеров строк, подсветку текущей строки, отображение разделителей между участками кода и т.п.

Задание 2

Создайте скрипт на языке Python, используя обычный текстовый редактор (можно использовать редактор кода, такой как, например, Sublime Text). Запустите его при помощи консоли. Запустите его двойным щелчком в проводнике Windows. Придумайте, как, используя то, что вы уже выучили, сделать так, чтобы окно консоли не закрывалось сразу же после запуска скрипта двойным щелчком.

Задание 3

Откройте IDLE (под Windows и OS X это приложение устанавливается вместе с Python). Поэкспериментируйте с обычными арифметическими выражениями. Попробуйте задать имя значению какого-либо выражения. Попробуйте вывести значение выражения с поясняющим текстом при помощи функции `print`, используя как имена, так и непосредственно выражения в качестве параметров функции.

Рекомендуемые ресурсы

Официальный сайт Python

<https://python.org>

Документация по Python

<https://docs.python.org/3/>

Статьи в Википедии о ключевых понятиях, рассмотренных на этом уроке

<https://ru.wikipedia.org/wiki/Python>

https://ru.wikipedia.org/wiki/Язык_программирования

<https://ru.wikipedia.org/wiki/Интерпретатор>

https://ru.wikipedia.org/wiki/Интегрированная_среда_разработки