

Python

01

Востребованность Python

Программист Python Django

 Мильжикс

Описание вакансии:

Приглашаем в нашу компанию программиста back-end Python

Требования:- опыт работы с Python 3.X — от 1-го года- знание фреймворка Django / DRF- опыт работы с системами семейства Linux- понимание особенностей web-разработки, HTTP-протокола- опыт работы с SQL, GIT- понимание процессов развертывания Python-приложения на рабочих серверах

Программист Python

 Совалабс

Описание вакансии:

Обязанности:

- Участие в разработке backend монолита и микросервисов.
- Оптимизация кода для высокопроизводительных вычислений на облачных платформах.

Требования:

- Отличное знание Python3, Django, Django REST Framework
- Знание Docker
- Знание Git
- Опыт работы с системами семейства Linux

Разработчик Python Full Stack (Django)



ООО БелТрансБилет

от 850 руб

Город: Минск

Занятость: Полная

Опыт работы: Не имеет значения

Обязанности

Разработка новых сервисов компании
Поддержка текущих проектов компании

Требования

Опыт работы с реляционными базами данных (PostgreSQL, MySQL);
Понимание принципов построения API в Интернете DRF (REST API);
Опыт интеграции вертки на Jinja2, Django Templates;
Знание основ работы с Git;

Будет плюсом:

Знание ООП и шаблонов проектирования; знание стандартов кодирования и передовой практики на языке Python;
Знание ОС на базе Unix (Ubuntu);
Проведение code review на проектах;
Опыт работы в Docker;
Опыт разработки проектов с использованием одного из MV*-фреймворков (Vue.js, React, Angular);

Образование

Среднее специальное

Программист Django/Python

 ВайсВеб

Описание вакансии:

Обязанности:

- Разработка и сопровождение программного обеспечения.

Требования:

- Уверенное знание Python/Django.
- Работа с БД postgresql.
- Представление о JS, HTML и CSS.
- Дополнительным плюсом будут знания и опыт работы с Linux (Ubuntu, Centos).

Junior Python Developer

з/п не указана

iTechArt Group ✓

Минск, улица Толстого, 10



Откликнуться



Требуемый опыт работы: не требуется

Полная занятость, полный день

iTechArt Group в связи с расширением приглашает в свою команду **Junior Python Developer**.

Компания iTechArt одна из самых быстро растущих компаний в сегменте разработки инженерных решений для ведущих технологических, финансовых, страховых и фармацевтических компаний.

Требования:

- Знакомство с **Python**, **Django**
- Понимание основных концепций реляционных БД
- Общее представление о разработке UI
- Понимание принципов ООП
- Английский язык на уровне Pre-Intermediate и выше

Программист Python



Телеком-ВИСТ

от 2 000 руб

Город: Минск

Занятость: Полная

Опыт работы: 2 года

Обязанности

Формирование технических заданий, проверка качества кода, оценка работы, написание кода

Требования

уверенное знание языка программирования Python, хорошее знание Django, хорошее знание SQL (PostgreSQL), Основы HTML, CSS, JavaScript, опыт работы с Linux; понимание конфигураций nginx, uWSGI;

Python программист



ВиАссист

от 1 500 руб

Город: Минск

Занятость: Полная

Опыт работы: 2 года

Обязанности

Создание высокотехнологичного продукта для банковской сферы с использованием языка программирования Python.

Требования

Опыт backend разработки от 2 лет;
Python, Django;
PostgreSQL, MongoDB, Redis;
Работа с Docker;
Свободное владение инструментами разработки – git, выбранными OS и IDE;

Знание языков

Английский

Основные требования

- Python
- Django
- Linux
- Git
- Docker
- Django REST Framework
- English
- SQL
- OOP
- HTML & CSS
- JS
- IDE

Описание языка Python

Python — *“высокоуровневый язык программирования, главной особенностью которого является читабельность и синтаксис, с помощью которого можно легко описать любую концепцию в нескольких строках кода.”*

© Гвидо Ван Россум(создатель языка)

Linux как среда разработки на python.

Для программиста главная цель разработки на linux - гарантировать одинаковое выполнение программы на сервере и на компьютере, на котором происходила разработка.

Linux

Основные команды:

cd [path] - переход в другую категорию

ls - содержимое текущей категории

mkdir - создание папки

pwd - текущее месторасположение

touch [filename] - создание файла

cat [filename] - отображение содержимого текстового файла

nano [filename] - редактирование файла из консоли

rm [filename] - удаление файла

rm -rf [dir_name] - удаление папки

Virtualenv

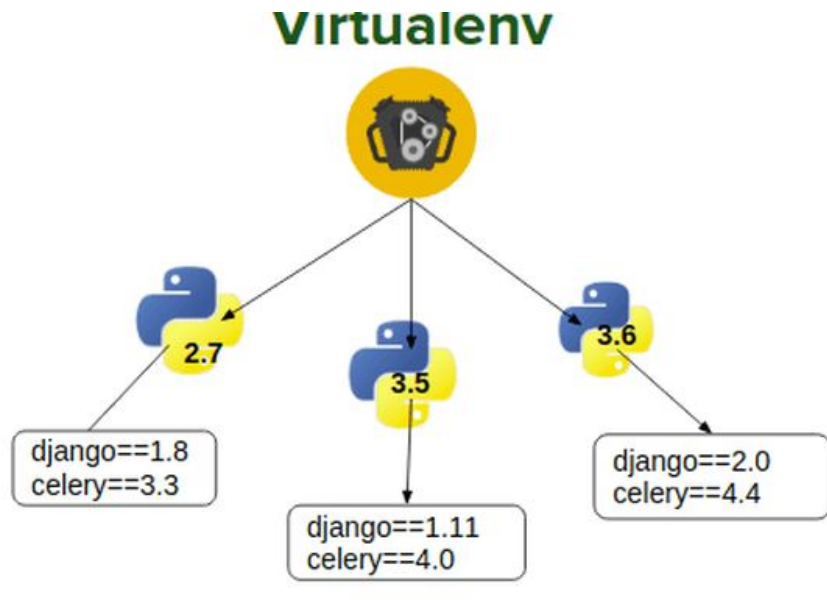
virtualenv - инструмент для создания изолированного окружения Python.

Основные команды:

`virtualenv -p pythonX.X [dirname]` -
создание окружения

`source [virtualenvpath]/bin/activate` -
активация окружения

`deactivate`



Pip

pip - менеджер пакетов для Python

Основные команды:

pip install [packagename] - установка пакета

pip install -r [path_to_requirements] - установка перечня пакетов

pip freeze - отображение установленных пакетов

pip uninstall [packagename] - удаление пакетов

«Hello world» на python

Запуск простейшей программы «Hello world»
используя Pycharm и интерпретатор

Основные типы данных

1. int - целые числа
2. float - числа с плавающей точкой
3. str - строки
4. bool - логические переменные True или False

print(4)

print(5.5)

print('Jason Statham')

print(True)

Переменные

Объявление переменной:

`a = 5`

`b = 4.8`

`c = 'test string'`

`d = True`

Имена переменных и ключевые слова

Хорошо

summ
summ_of_variables
_new_sum
is_payable
__another_variable

Плохо

Summ
summOfVariables
New_sum
isPayable
__anotherVariable

Нельзя

Sum%m
\$summOfVariable\$
42New_sum
isP*yable
__anotherV@riable

Правила для именования переменных

Правила для именования переменных:

- Имя переменной должно начинаться с буквы или нижнего подчеркивания
- Имя не может начинаться с цифры
- Имя переменной может содержать только буквы английского алфавита, цифры и нижние подчеркивания (A-z, 0-9, and _)
- Имена переменных чувствительны к регистру (age, Age и AGE три разные переменные)

Соглашение для именования переменных:

- При наличии нескольких слов в переменной в качестве разделителя использовать нижнее подчеркивание `sum_of_cats`
- Не использовать в переменных буквы верхнего регистра ~~FirstName = 'Jason'~~ `first_name = 'Jason'`
- Использовать осознанные имена для переменных: ~~a = 'Jason Statham'~~ `fullname = 'Jason Statham'`

Задание 1.1

Определить переменные всех 4 базовых типов и вывести их на экран

```
>>> a = 5
```

```
>>> print(a)
```

Операторы и операнды

Операторы (operators) – специальные символы, которые представляют вычисления, наподобие сложения и умножения. Значения, к которым применяется оператор, называются операндами (operands).

Операторы и операнды

a равно 10 и b равно 20

+ сложение	Складывает значение переменных слева и справа от оператора	$a + b = 30$
- вычитание	Вычитает из левого операнда правый	$a - b = -10$
* умножение	Перемножает левый и правый операнды	$a * b = 200$
/ деление	Делит левый операнд на правый	$b / a = 2$
% остаток от деления	Делит левый операнд на правый и возвращает остаток от деления	$b \% a = 0$ и $42 \% 10 = 2$
** возведение в степень	Возводит левый операнд в степень равную правому оператору	$5^{**2} = 25$ и $2^{**3} = 8$
// целочисленное деление	Делит левый операнд на правый и оставляет только целую часть.	$9 // 2 = 4$ and $9.0 // 2.0 = 4.0$, $-11 // 3 = -4$, $-11.0 // 3 = -4.0$

Выражения

Выражение (expression) – это комбинация значений, переменных и операторов. Значение (или переменная) само по себе является выражением, поэтому все следующие выражения являются корректными (предполагается, что переменной x было присвоено значение):

17

x

$x + 17$

Порядок операций

Если в выражении встречается больше, чем один оператор, то порядок вычислений зависит от правил старшинства (rules of precedence). Для математических операций, Python следует математическим соглашениям. Аббревиатура PEMDAS является простым способом для запоминания правил:

Parentheses - Скобки

Exponentiation - Возведение в степень

Multiplication - Умножение

Division - Деление

Addition - Сложение

Subtraction - Вычитания

Порядок операций

Скобки (Parentheses) имеют наивысший приоритет и могут использоваться для принудительного определения порядка вычислений в выражении. Таким образом, результат выражения $2*(3-1)$ будет равен 4, $(1+1)**(5-2)$ будет равен 8. Вы также можете использовать скобки для упрощения чтения выражений, например, $(minute*100) / 60$, если это не повлияет на результат.

Возведение в степень (Exponentiation) имеет наибольший приоритет, так $2**1+1$ равно 3, а не 4, и $3*1**3$ равно 3, а не 27.

Умножение и деление (Multiplication and Division) имеют одинаковый приоритет, который выше сложения и вычитания (Addition and Subtraction), которые также имеют одинаковый приоритет. Таким образом, $2*3-1$ равно 5, а не 4, и $6+4/2$ равно 8, а не 5.

Операторы с одинаковым приоритетом вычисляются слева направо. Таким образом, $5-3-1$ равно 1, а не 3. За исключением возведения в степень, в таком случае выражения вычисляются справа налево : $3**2**3 = 3**(2**3)$

Задание 1.2

Найти значение выражений

1. $x = 17 / 2 * 3 + 2$

2. $x = 2 + 17 / 2 * 3$

3. $x = 19 \% 4 + 15 / 2 * 3$

4. $x = (15 + 6) - 10 * 4$

5. $x = 17 / 2 \% 2 * 3 ** 3$

Расставить скобки так, чтобы значение выражений не поменялось.

Обновление переменных ($x=x+1$)

1) переменные могут использоваться для установки значений других переменных

```
my_age = 17
```

```
alisa_age = my_age + 4
```

```
fred_age = alisa_age + 1
```

2) "обновление самой себя"

```
my_age = 10
```

```
my_age = my_age + 5
```

```
my_age = my_age + 1
```

```
my_age += 1
```

Задание 1.3

Создать три переменные, содержащие возраст трех ближайших соседей, найти сумму и вывести ее на экран. Создать еще одну переменную равную среднему арифметическому возрасту, и вывести это значение на экран.