

Федеральное государственное бюджетное образовательное учреждение высшего образования
«Казанский национальный исследовательский технический университет им. А.Н. Туполева – КАИ»
(КНИТУ – КАИ)

ВВЕДЕНИЕ В ПРОГРАММИРОВАНИЕ

Доцент кафедры компьютерных систем,
к.т.н., Классен Роман Константинович

Что такое программирование?

Программирование – процесс создания компьютерных программ.

Программирование – это процесс превращения алгоритма в нотацию, написанную на языке программирования, которая может быть выполнена компьютером.

Программирование – искусство создания компьютерных программ.

Программирование – разработка программ управления компьютером с целью решения различных информационных задач.

Программирование – раздел прикладной математики, изучающий и разрабатывающий методы и средства составления программ для вычислительных машин.

Что такое компьютерная программа?

- 1) комбинация компьютерных инструкций и данных, позволяющая аппаратному обеспечению вычислительной системы выполнять вычисления или функции управления;
- 2) синтаксическая единица, которая соответствует правилам определённого языка программирования, состоящая из определений и операторов или инструкций, необходимых для определённой функции, задачи или решения проблемы.

* *Компьютерная программа // Википедия. [2020]. Дата обновления: 14.10.2020. URL: <https://ru.wikipedia.org/?curid=10732&oldid=109868076> (дата обращения: 14.10.2020).*

Что нужно для программирования?

- Компьютер (или иное устройство с возможностью ввода данных запуска произвольных программ (да, программировать можно даже на телефоне ☺))
- Исполняющее устройство (если программа разрабатывается для другого устройства)
- Текстовый редактор (лучше IDE – Integrated Development Environment)
- Компилятор или интерпретатор
- Базовые знания одного из языков программирования
- Базовые знания в построении алгоритмов
- Базовые знания математики (все школу закончили? ☺)

С чего начать изучение программирования?

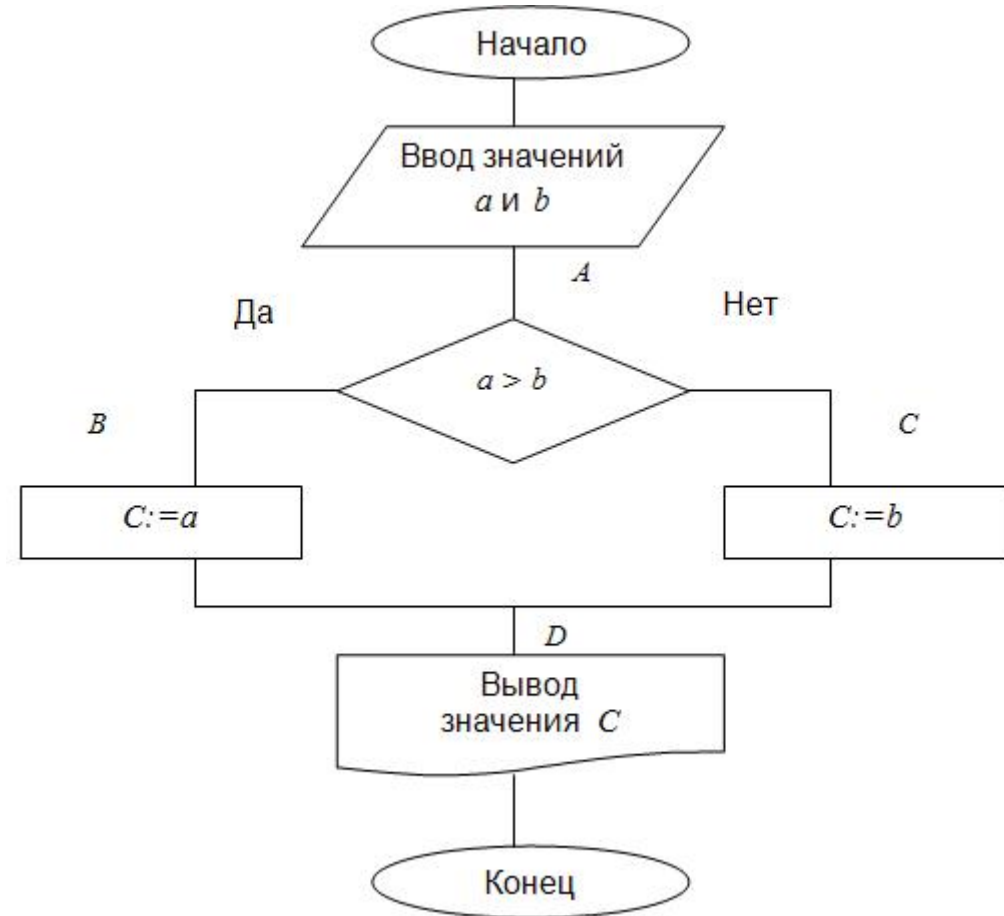
1. Изучить математику, математическую логику, дискретную математику
2. Изучить устройство и принцип работы вычислительных машин
3. Изучить основы построения алгоритмов
4. Выбрать язык программирования
5. Программировать
6. Еще программировать
7. И еще программировать

«Научиться программировать можно только программируя»

А можно научиться построению алгоритмов в процессе изучения программирования?

Да, обычно это так и происходит 😊

Однако, предварительное изучение основ построения алгоритмов существенно упрощает последующий процесс написания программы и позволяет избежать ошибок.



Какие языки программирования бывают?

Классификация программирования

- Языки низкого и высокого уровня
 - Assembly
 - C
- Безопасные и небезопасные языки
 - C#, Java
 - C, C++
- Компилируемые,
интерпретируемые и встраиваемые
языки
 - C
 - PHP
 - SQL

языков

Парадигмы программирования

- Императивное программирование
 - C, C++, C#, Java, Python и др.
- Декларативное программирование
 - Prolog, Haskell, Lisp
- Метaprogramмирование

Лучшие языки для обучения

1. Pascal, C – просты и не требовательны, позволяют «прочувствовать» каким было программирование 50 лет назад.
2. JavaScript – немного сложнее, но для начала изучения потребуется только браузер.
3. Python – простой и современный язык, идеален для создания прототипов и автоматизации, но имеет специфический синтаксис.
4. C#, Java, PHP, Ruby и др. – «серьезные» языки для решения рабочих задач
5. C++, Rust, Qt и др. – позволяют писать чрезвычайно быстрые программы, но высока цена ошибки
6. Assembly, IL – близкие к «железу» языки

Какие языки сейчас популярны?

TIOBE Index for
January 2021

Jan 2021	Jan 2020	Change	Programming Language	Ratings	Change
1	2	▲	C	17.38%	+1.61%
2	1	▼	Java	11.96%	-4.93%
3	3		Python	11.72%	+2.01%
4	4		C++	7.56%	+1.99%
5	5		C#	3.95%	-1.40%
6	6		Visual Basic	3.84%	-1.44%
7	7		JavaScript	2.20%	-0.25%
8	8		PHP	1.99%	-0.41%
9	18	▲▲	R	1.90%	+1.10%
10	23	▲▲	Groovy	1.84%	+1.23%
11	15	▲▲	Assembly language	1.64%	+0.76%
12	10	▼	SQL	1.61%	+0.10%
13	9	▼▼	Swift	1.43%	-0.36%
14	14		Go	1.41%	+0.51%
15	11	▼▼	Ruby	1.30%	+0.24%
16	20	▲▲	MATLAB	1.15%	+0.41%
17	19	▲	Perl	1.02%	+0.27%

[www.tiobe.com/
tiobe-index/](http://www.tiobe.com/tiobe-index/)

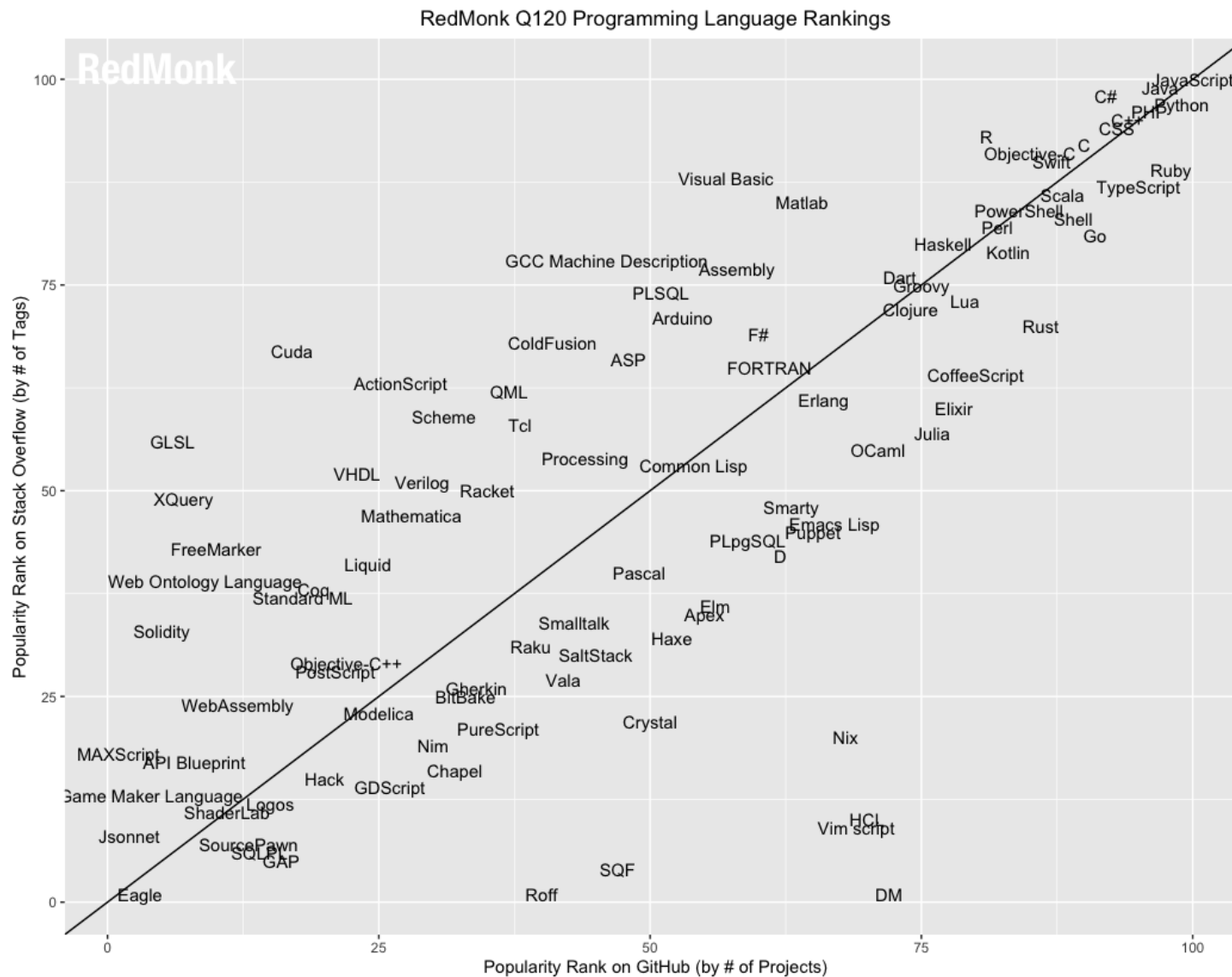
Какие языки сейчас популярны?

PYPL PopularitY of Programming Language

<https://pypl.github.io/PYPL.html>

Rank	Change	Language	Share	Trend
1		Python	30.44 %	+1.2 %
2		Java	16.76 %	-2.0 %
3		JavaScript	8.44 %	+0.3 %
4		C#	6.53 %	-0.7 %
5	↑	C/C++	6.33 %	+0.3 %
6	↓	PHP	6.05 %	-0.2 %
7		R	3.87 %	+0.1 %
8		Objective-C	3.71 %	+1.2 %
9		Swift	2.14 %	-0.3 %
10		TypeScript	1.78 %	-0.0 %
11		Matlab	1.74 %	-0.1 %
12		Kotlin	1.7 %	+0.0 %
13	↑↑	Go	1.33 %	+0.1 %
14	↓	VBA	1.2 %	-0.2 %
15	↓	Ruby	1.12 %	-0.2 %

Какие языки сейчас популярны?



Почему языков программирования так много?

Языки программирования создаются для наиболее эффективного решения некоторого круга задач. Например:

- C# – для разработки Windows приложений и больших приложений
- Java – для разработки универсальных приложений и больших приложений
- PHP – для создания web-сайтов
- JavaScript – для создания динамических web-сайтов
- C++ – для создания больших и быстрых приложений (игры, операционные системы)
- Python (лидер рейтингов последних лет) – для построения прототипов, машинного обучения и автоматизации

Почему Python?

- Легкость чтения исходного кода за счет специфического синтаксиса.
- Продуктивность труда разработчиков – объем исходного кода в 3-5 раз меньше по сравнению с аналогичными программами на C# или Java.
- Переносимость программ – возможность запуска на различных платформах и ОС.
- Интеграция компонентов – возможность коммуникации с внешними системами и использования библиотек других языков.
- Огромный набор готовых библиотек – следует из предыдущего пункта, имеются библиотеки практически для любого случая в жизни.

Кто использует Python?

- Google всесторонне использует Python в своих системах веб-поиска. Популярная пиринговая система обмена файлами BitTorrent начинала свое существование как программа Python.
- Industrial Light & Magic, Pixar и другие компании используют Python в производстве анимационных фильмов.
- ASA использует Python для шифрования и анализа разведывательной информации.
- iRobot применяет Python при разработке коммерческих и военных роботизированных устройств.
- Настраиваемые сценарные события в игре Civilization IV написаны целиком на Python.
- Netflix и Yelp документально подтвердили роль Python в своих инфраструктурах ПО.
- Intel, Cisco, Hewlett-Packard, Seagate, Qualcomm и IBM используют Python для аппаратного тестирования.
- JPMorgan Chase, UBS, Getco и Citadel применяют Python для выработки прогнозов на финансовом рынке.
- NASA, Los Alamos, Fermilab, JPL и другие компании используют Python для решения задач научного программирования.

Что можно делать с помощью Python?

- Писать системное программное обеспечение (системные утилиты)
- Создавать графические пользовательские интерфейсы
- Писать сценарии сценариев для Интернета (передача данных)
- Интегрировать компоненты различных систем (встраивание python в другие системы)
- Выполнять манипуляции с различными СУБД
- Обработать данные экспериментов или выполнять расчет модели
- Выполнять машинное обучение
- Управлять роботом
- И т.д.

Как работает Python?



Перед началом

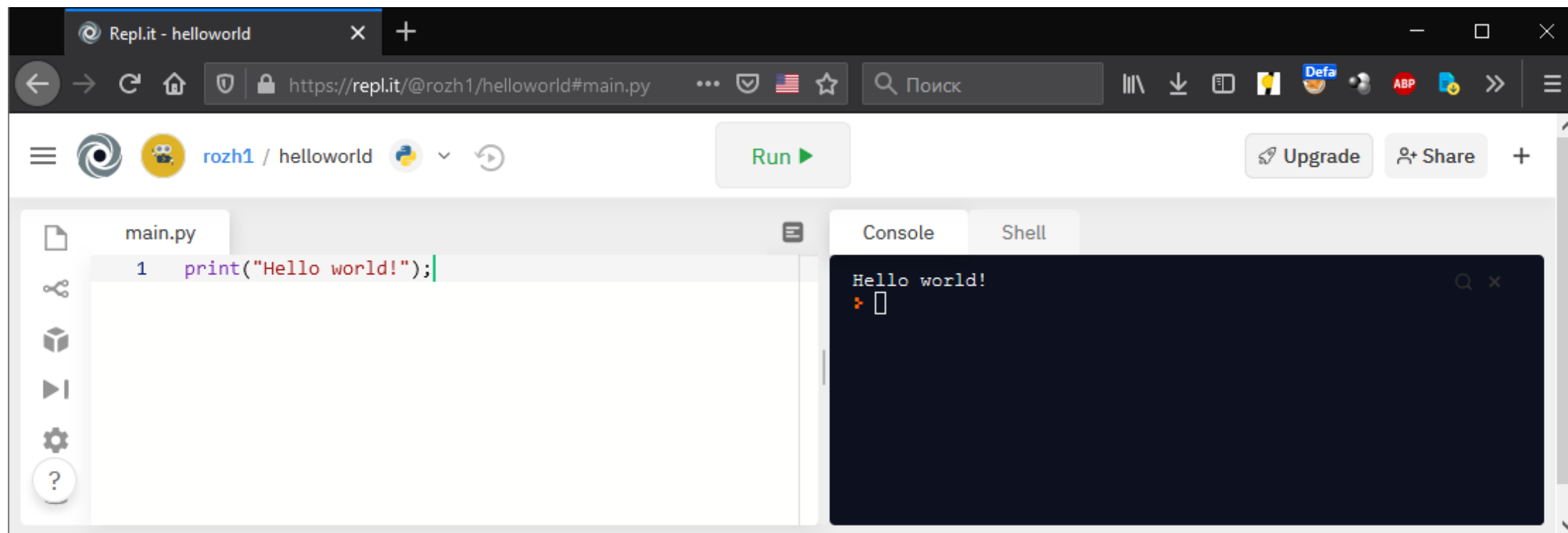
Необходимо:

- Компьютер с ОС Windows, Linux или Mac
- Текстовый редактор (достаточно простого блокнота)
- Компилятор или интерпретатор (<https://www.python.org/downloads/windows/>)

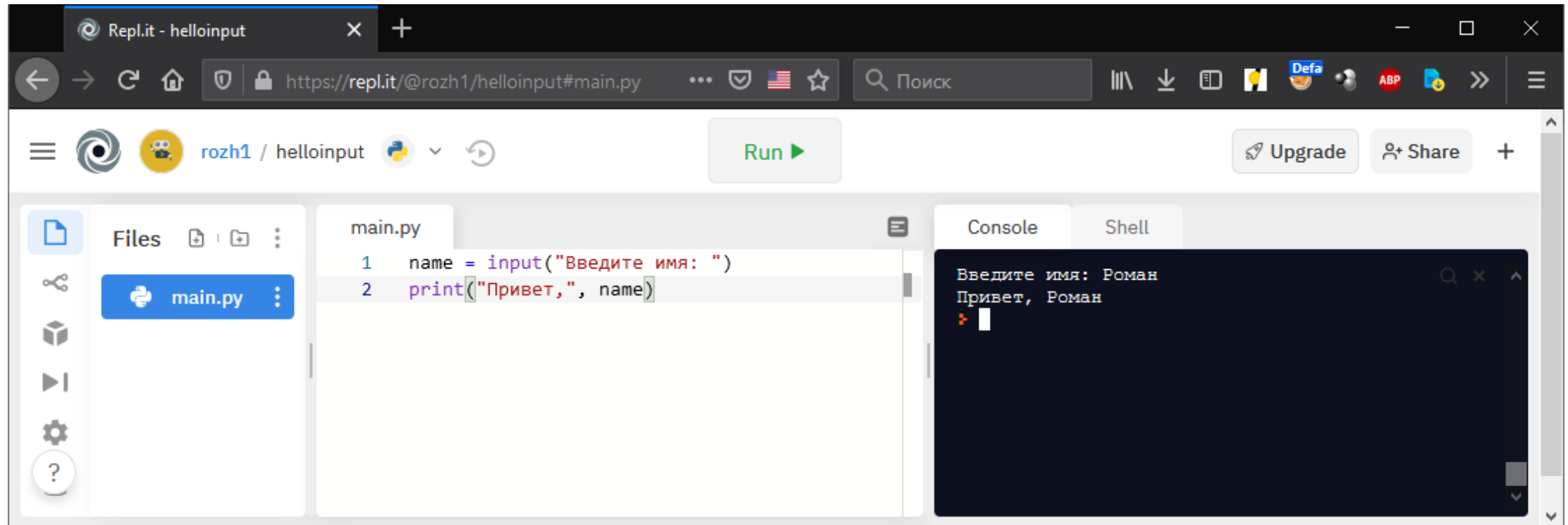
Или воспользоваться одной из online IDE:

- https://www.tutorialspoint.com/execute_python3_online.php
- <https://repl.it/languages/python3>
- <https://ideone.com/>

Первая программа на Python



Вторая программа



The screenshot shows a web browser window with a Repl.it Python REPL. The browser's address bar shows the URL `https://repl.it/@rozh1/helloinput#main.py`. The Repl.it interface includes a 'Run' button, 'Upgrade', and 'Share' options. On the left, a 'Files' panel shows a file named `main.py`. The main editor displays the following Python code:

```
main.py
1 name = input("Введите имя: ")
2 print("Привет,", name)
```

On the right, the 'Console' panel shows the program's output:

```
Введите имя: Роман
Привет, Роман
>
```

Основы Python

Программа на языке Python состоит из набора инструкций. Каждая инструкция помещается на новую строку.

Большую роль в Python играют отступы. Они определяют область видимости подобно {} в C-подобных языках. Неправильно поставленный отступ фактически является ошибкой.

Python – регистрозависимый язык, поэтому выражения print и Print или PRINT представляют разные выражения.

Для отметки, что делает тот или иной участок кода, применяются комментарии.

```
print(2 + 3)
print("Hello")
```

```
print(2 + 3)
    print("Hello")
```

```
Print("Hello World")
```

```
# Комментарий
```

Основные функции ввода/вывода

Основной функцией для вывода информации на консоль является функция **print()**. В качестве аргумента в эту функцию передается строка, которую мы хотим вывести.

```
print("Hello world")
```

Если же нам необходимо вывести несколько значений на консоль, то мы можем передать их в функцию **print** через запятую.

```
print("Name:", "Tom")
```

За ввод информации отвечает **input**. В качестве необязательного параметра эта функция принимает приглашение к вводу и возвращает введенную строку, которую мы можем сохранить в переменную.

```
name = input("Введите имя: ")  
print("Привет", name)
```

Переменные

Переменная хранит определенные данные. Название переменной в Python должно начинаться с алфавитного символа или со знака подчеркивания и может содержать алфавитно-цифровые символы и знак подчеркивания. И кроме того, название переменной не должно совпадать с названием ключевых слов языка Python:

and, as, assert, break, class, continue, def, del, elif, else, except, False, finally, for, from, global, if, import, in, is, lambda, None, nonlocal, not, or, pass, raise, return, True, try, while, with, yield.

```
name = "Tom"  
userName = "Tom"  
user_Name = "Tom"  
A = 2  
a = 1  
_global = True
```

Типы данных

- `boolean` - логическое значение `True` или `False`
- `int` - представляет целое число
- `float` - представляет число с плавающей точкой
- `complex` - комплексные числа
- `str` - строки
- `bytes` - последовательность чисел в диапазоне 0-255
- `byte array` - массив байтов, аналогичен `bytes` с тем отличием, что может изменяться
- `list` - список
- `tuple` - кортеж
- `set` - неупорядоченная коллекция уникальных объектов
- `frozen set` - то же самое, что и `set`, только не может изменяться (`immutable`)
- `dict` - словарь, где каждый элемент имеет ключ и значение

```
bool = True
a = 50
f = 3.14
z = complex(5, 3)
str = "hello"
b = bytes([50, 100])
ba = bytearray([50, 100])

l = [1, 2, 3, 4, 5]
t = (1, 2, 3, 4, 5)
s = {1, 2, 3, 4, 5}
fs = frozenset([1, 2, 3, 4])
d = {'d1': 1, 'd2': 2}
```

Арифметические операции

- + - сложение `print(6 + 2) # 8`
- - - вычитание `print(6 - 2) # 4`
- * - умножение `print(6 * 2) # 12`
- / - деление `print(6 / 2) # 3.0`
- // - целочисленное деление `print(7 / 2) # 3.5`
`print(7 // 2) # 3`
- ** - возведение в степень `print(6 ** 2) # 36`
- % - остаток от деления `print(7 % 2) # 1`

Операции сравнения и логические операции

- `==` - равенство

```
print(6 == 5) # False
```

- `!=` - не равенство

```
print(6 != 5) # True
```

- `>` - больше

```
print(6 > 5) # False
```

- `<` - меньше

```
print(6 < 5) # True
```

- `>=` - больше или равно

```
print(6 >= 5) # False
```

- `<=` - меньше или равно

```
print(6 <= 5) # True
```

- `or` – или

```
print(3>2 and 4>3) # True
```

- `and` – и

```
print(3>2 or 4<3) # True
```

- `not` – отрицание

```
print(not 4<3) # True
```

Условная конструкция if

```
if логическое_выражение:
```

```
    инструкции
```

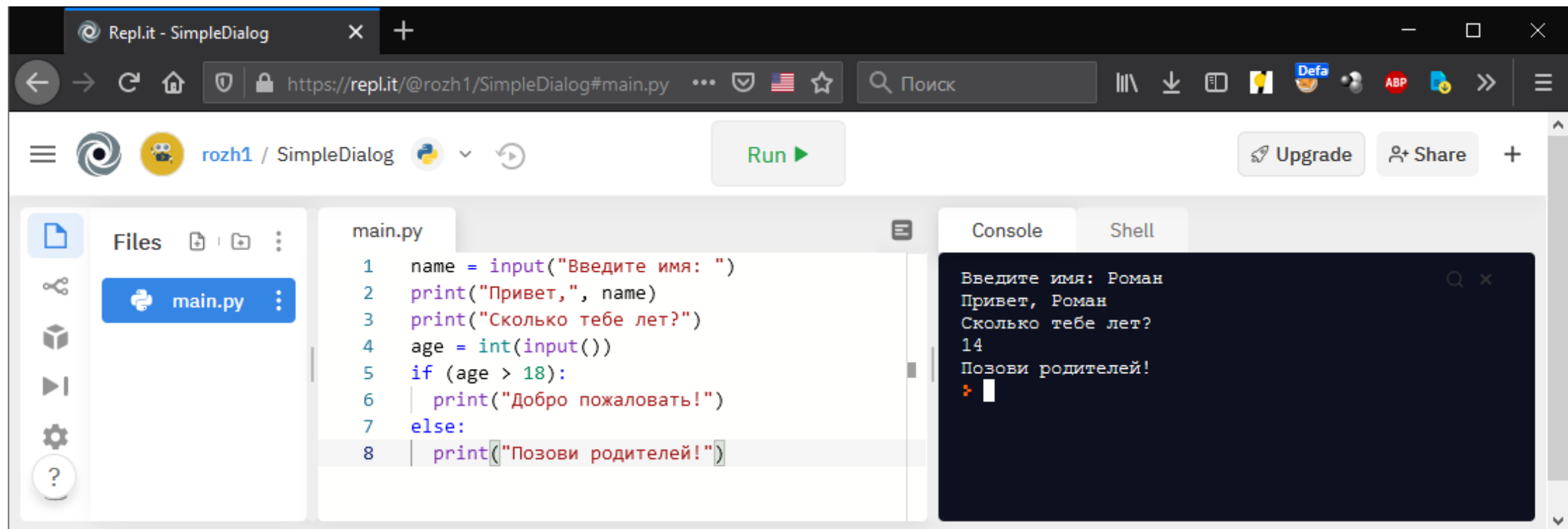
```
[elif логическое выражение:
```

```
    инструкции]
```

```
[else:
```

```
    инструкции]
```

Простое диалоговое приложение



The screenshot shows a web browser window with a Repl.it Python REPL. The URL is `https://repl.it/@rozh1/SimpleDialog#main.py`. The code in `main.py` is as follows:

```
1 name = input("Введите имя: ")
2 print("Привет,", name)
3 print("Сколько тебе лет?")
4 age = int(input())
5 if (age > 18):
6     print("Добро пожаловать!")
7 else:
8     print("Позови родителей!")
```

The console output shows the program's execution with the user input "Роман" and "14":

```
Введите имя: Роман
Привет, Роман
Сколько тебе лет?
14
Позови родителей!
>
```