

Почему программирование?

Глава 1

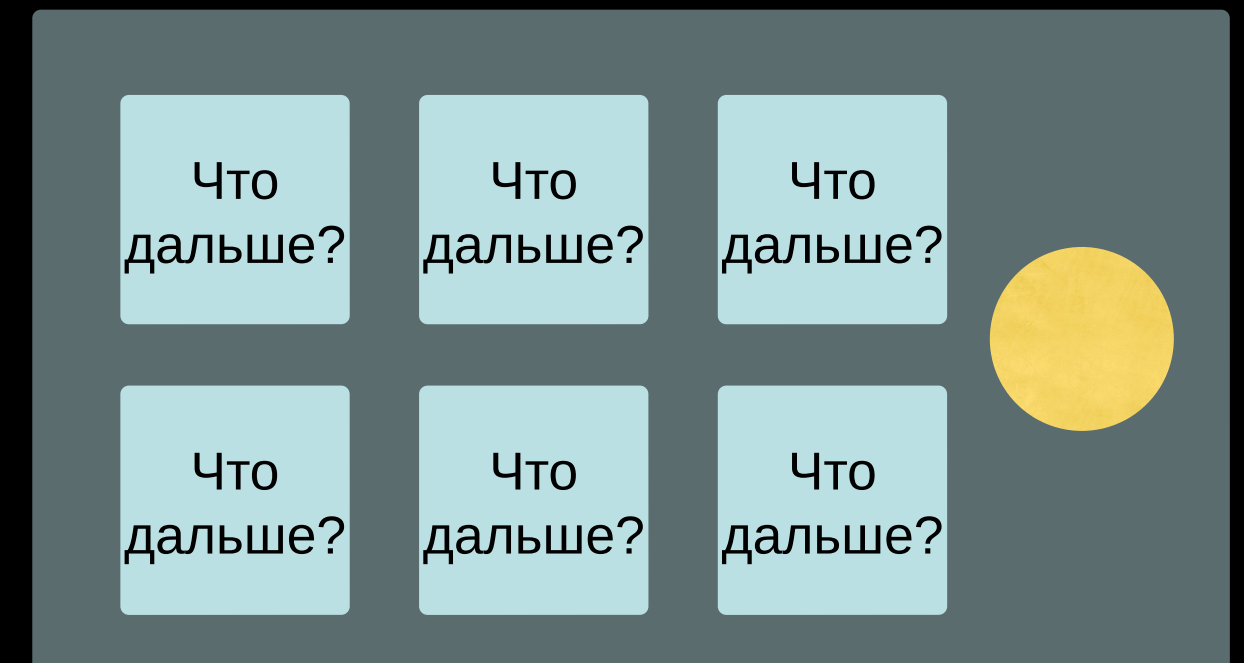


Пайтон для всех
www.py4e.com



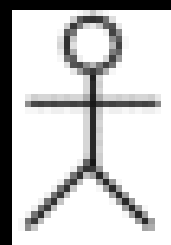
Компьютеры хотят быть полезными...

- Компьютеры созданы с одной целью — быть для нас полезными
- Но нам необходимо говорить на их языке, чтобы описать, чего мы хотим
- К удобству пользователей кто-то уже поместил множество разных программ (инструкций) в компьютер, а пользователям нужно лишь выбрать, какую из них они хотят использовать



Программисты предвидят потребности

- iPhone-приложения — это целый рынок
- iPhone-приложения имеют более 3 миллиардов скачиваний
- Все больше программистов становятся разработчиками iPhone-приложений
- Программисты разбираются в том, как **работает программа**

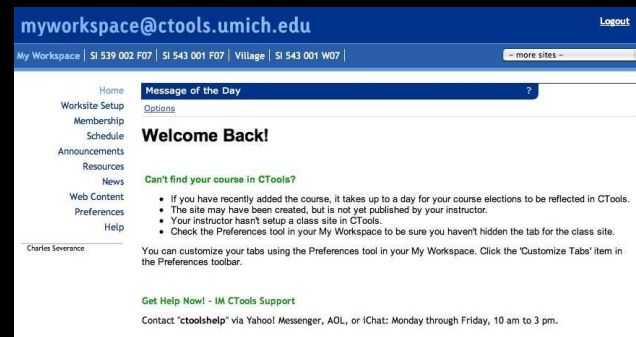


Пользователи vs. Программисты

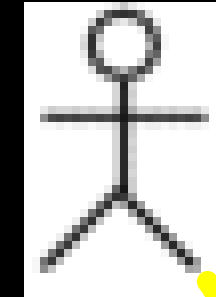
- Пользователи воспринимают компьютер, как набор инструментов — текстовый редактор, электронная таблица, карта, список дел и т.д.
- Программисты изучают как работает компьютер и язык компьютера
- Программисты используют специальный инструментарий, который позволяет им создавать новые инструменты (программы)
- Иногда программисты создают программы для большого количества пользователей, а иногда пишут код для себя в качестве помощника для автоматизации какой-либо задачи

Зачем становиться программистом?

- Чтобы решить некую задачу: в этом случае мы и пользователь, и программист в одном лице
 - Например, очистить данные опроса
- Чтобы создавать нечто, чем смогут пользоваться другие: профессия программиста
 - Устранить проблему с производительностью в программном обеспечении Sakai
 - Добавить на веб-сайт гостевую книгу



Пользователь



Компьютер

Аппаратное обеспечение + Программное
обеспечение



Программист

Данные

Информаци
я

...

Сети

С точки зрения создателя программного обеспечения, мы создаем программное обеспечение. Конечные пользователи — наши хозяева, им мы хотим угодить. Они платят нам, когда удовлетворены. Но данные, информация и сети — наша проблема, нам приходится решать ее от их имени. Оборудование и программное обеспечение — наши друзья и союзники в этом деле

Что такое код? Программное обеспечение? Программа?

- **Последовательность сохраненных инструкций**
 - Это маленькая частичка нашего интеллекта в компьютере
 - Мы что-то узнаем, затем кодируем и отдаем результат кому-то другому, чтобы он мог сэкономить свои силы и время
- **Произведение искусства**, особенно, если мы хорошо поработали над пользовательским интерфейсом (чтобы сделать опыт пользователя приятным)

Программы для людей...



<https://www.youtube.com/watch?v=XiBYM6g8Tck>

Программы для людей...

Пока играет музыка:

Левая рука вперед и вверх
Правая рука вперед и вверх
Перевернуть левую руку
Перевернуть правую руку
Левую руку к правому плечу
Правую руку к левому плечу
Левую руку на затылок
Правую руку на затылок
Левую руку на правое ведро
Правую руку на левое ведро
Левую руку на левую ягодицу
Правую руку на правую ягодицу
Покачивание
Покачивание
Прыжок



<https://www.youtube.com/watch?v=XiBYM6g8Tck>

Программы для людей...

Пока играет музыка:

Левая рука вперед и вверх

Правая рука вперед и вверх

Перевернуть левую руку

Перевернуть правую руку

Левую руку к правому плечу

Правую руку к левому плечу

Левую руку к затылку

Правую **муку** на затылок

Левую руку на правое **ведро**

Правую руку на левое **ведро**

Левую руку на левую ягодицу

Правую руку на правую ягодицу

Покачивание

Покачивание

Прыжок



<https://www.youtube.com/watch?v=XiBYM6g8Tck>

Программы для людей...

Пока играет музыка:

Левая рука вперед и вверх
Правая рука вперед и вверх
Перевернуть левую руку
Перевернуть правую руку
Левую руку к правому плечу
Правую руку к левому плечу
Левую руку на затылок
Правую **руку** на затылок
Левую руку на правое **бедро**
Правую руку на левое **бедро**
Левую руку на левую ягодицу
Правую руку на правую ягодицу
Покачивание
Покачивание
Прыжок



<https://www.youtube.com/watch?v=XiBYM6g8Tck>

Программы на Пайтон...

клоун побежал за машиной,
машина врезалась в палатку,
и палатка упала на клоуна и машину



Изображение: https://www.flickr.com/photos/allan_harris/4908070612/ Attribution-NoDerivs 2.0 Generic (CC BY-ND 2.0)

Программы на Пайтон...



Изображение: https://www.flickr.com/photos/allan_harris/4908070612/ Attribution-NoDerivs 2.0 Generic (CC BY-ND 2.0)

```
name = input('Введите имя файла:')
handle = open(name)

counts = dict()
for line in handle:
    words = line.split()
    for word in words:
        counts[word] = counts.get(word, 0) + 1

bigcount = None
bigword = None
for word, count in counts.items():
    if bigcount is None or count > bigcount:
        bigword = word
        bigcount = count

print(bigword, bigcount)
```

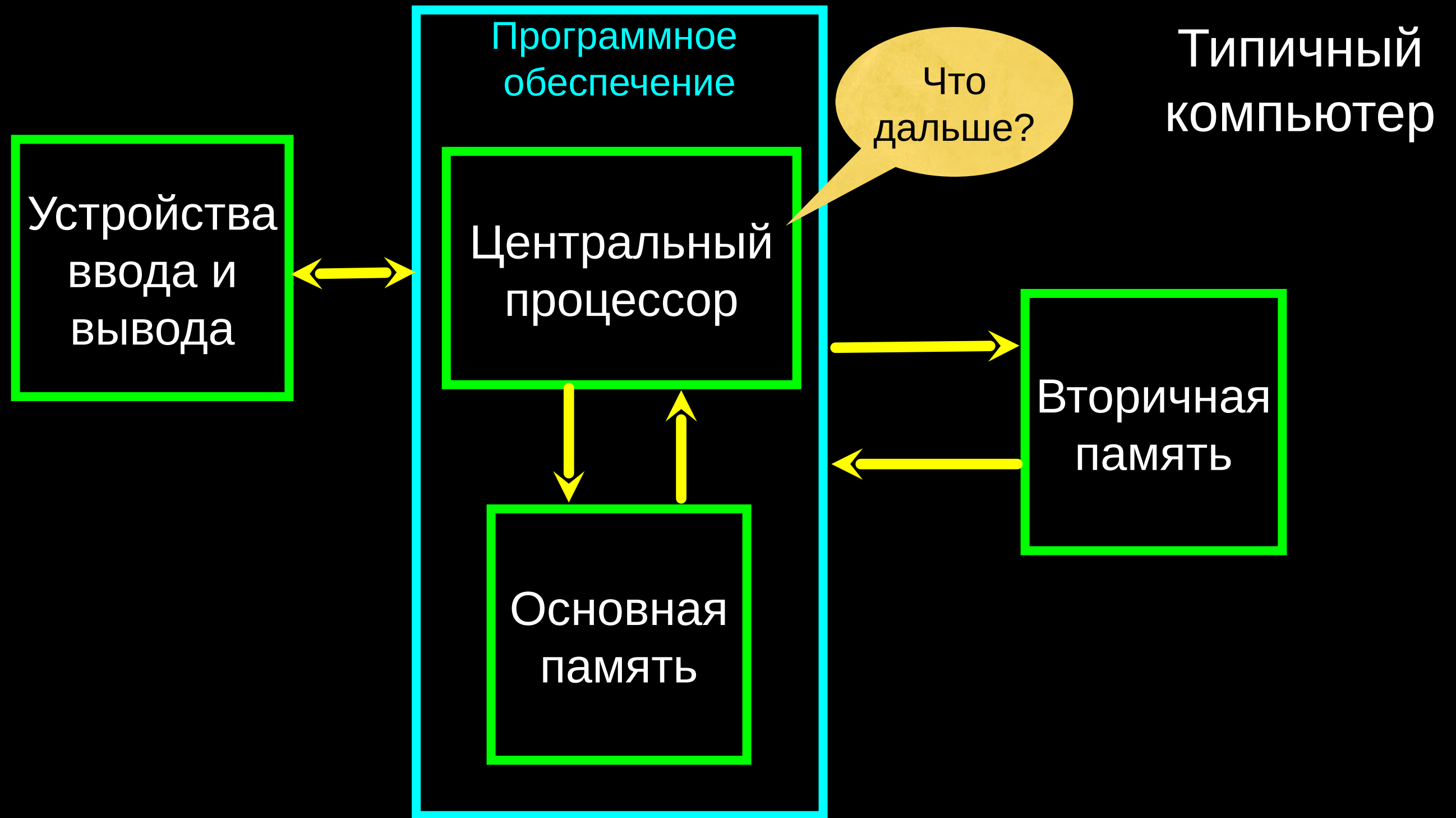
```
python words.py
Введите имя файла:
words.txt
to 16
```

```
python words.py
Введите имя файла:
clown.txt
the 7
```

Аппаратная Архитектура



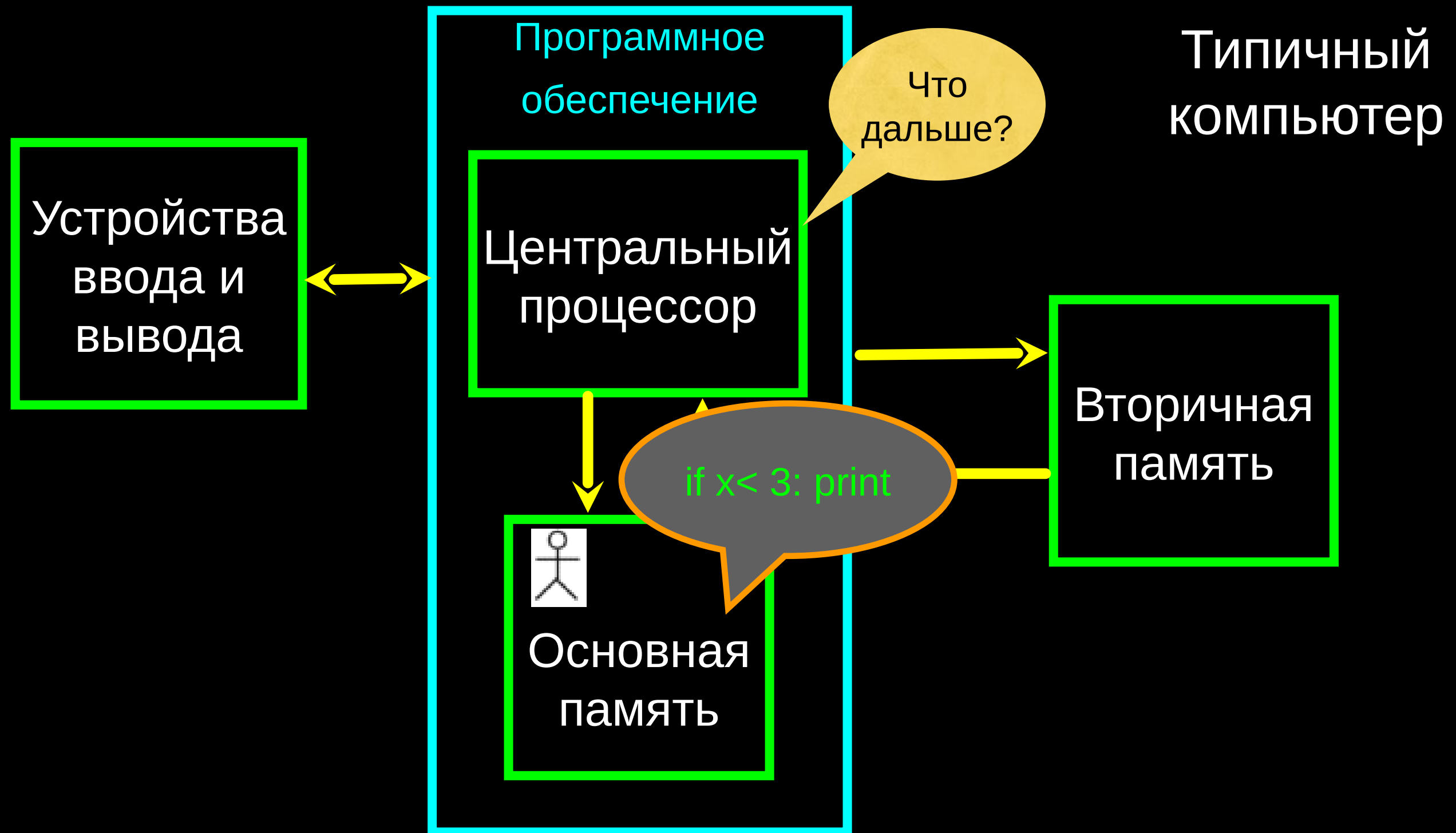
<http://upload.wikimedia.org/wikipedia/commons/3/3d/RaspberryPi.jpg>

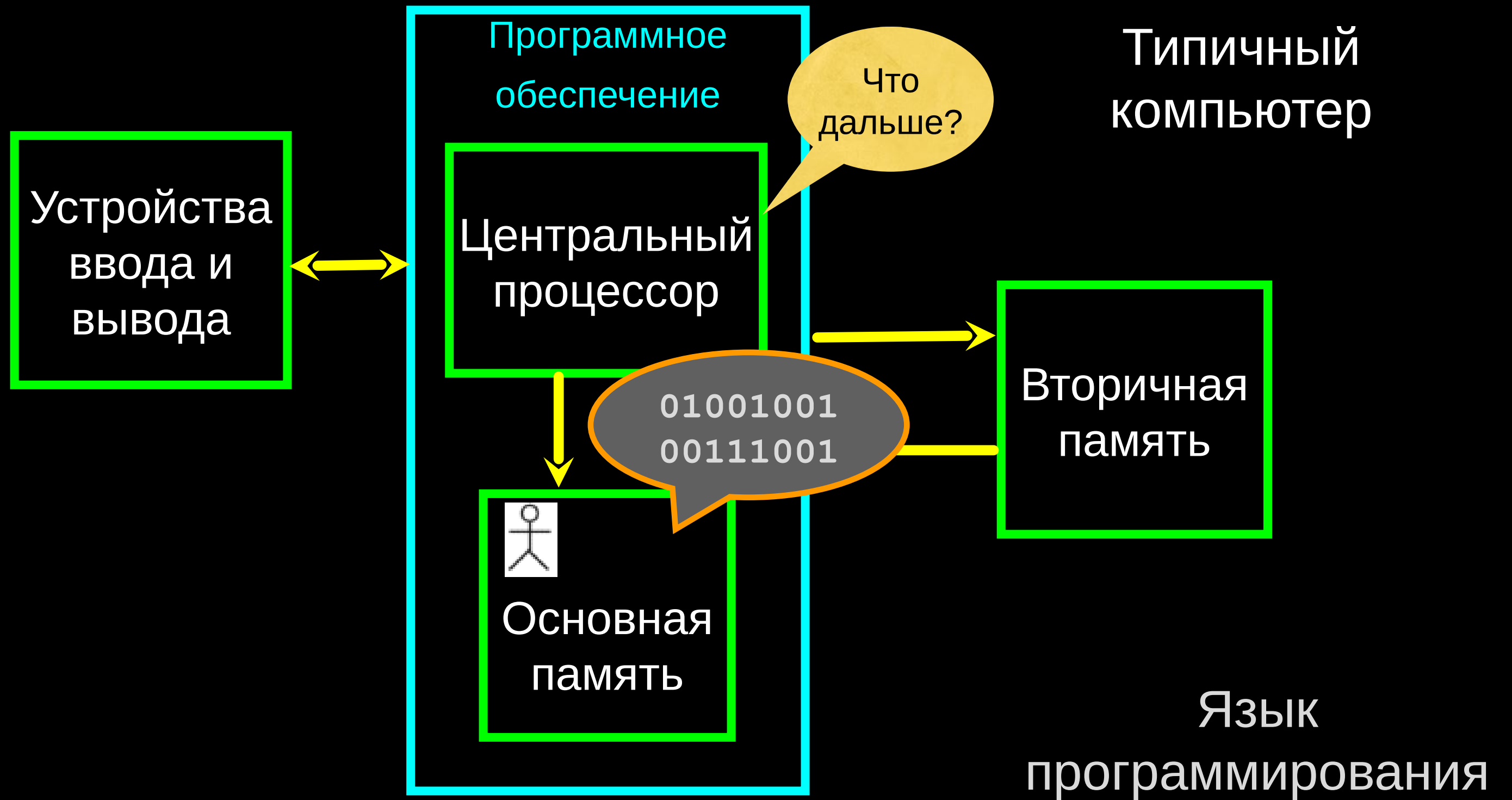


Определения

- **Центральный процессор:** Запускает программу. ЦП всегда задается вопросом: «Что дальше?». Не является мозговым центром – глупый, но очень быстрый;
- **Устройства ввода:** Клавиатура, Мышь, Сенсорный экран;
- **Устройства вывода:** Экран, Колонки, Принтер, Записывающее DVD-устройство;
- **Основная память:** Быстрое маленькое временное хранилище. Данные теряются после перезагрузки – он же RAM (ЗУПВ);
- **Вторичная память:** Более медленное большое постоянное хранилище. Остается, пока не будет удалено – диск / карта памяти;







Горячий Центральный Процессор



<http://www.youtube.com/watch?v=y39D4529FM4>

Жесткий диск в работе



<http://www.youtube.com/watch?v=9eMWG3fwiEU>

Пайтон как язык программирования

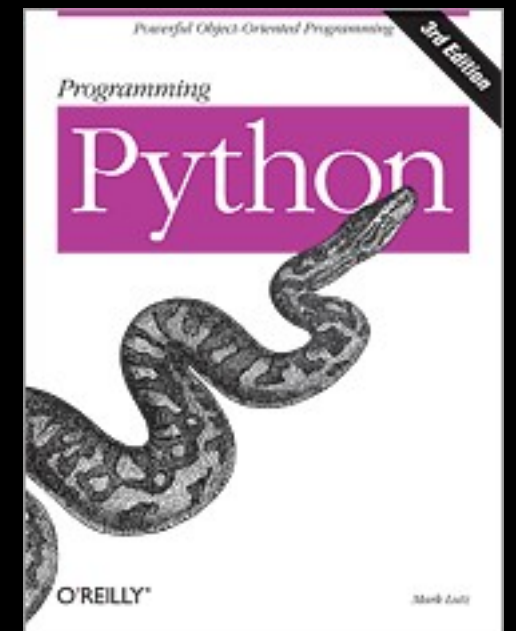
Змеиный язык или Парселтанг — магический язык змей и тех, кто способен с ними общаться. Человека, говорящего на **Парселтанге**, называют **Змееуст**. Это очень необычный навык и он может передаваться по наследству. Практически все известные **Змееусты** произошли от [Салазара Слизерина](#).



<http://harrypotter.wikia.com/wiki/Parseltongue>

Пайтон — язык интерпретатора Пайтон и тех, кто может с ним общаться. Человека, владеющего **Пайтон**, называют **Пайтонист**.

Это очень необычный навык и он может передаваться по наследству. Практически все **Пайтонисты** используют программное обеспечение разработанное **Гвидо Ван Россумом**.



Начало обучения: Синтаксические ошибки

- Нам нужно изучить **язык Пайтон**, чтобы передавать инструкции Пайтон-интерпретатору. В начале обучения мы будем делать множество ошибок и нести белиберду, словно маленькие дети.
- Когда мы делаем ошибку, компьютеру это не кажется забавным. Он выдает сообщение: **“синтаксическая ошибка”**, как-бы подчеркивая, что он знает язык, а вы его еще только учите. Кажется, Пайтон жесток и бесчувственен.
- Но вы должны помнить, что люди умны и наделены способностью учиться. Компьютер простой и очень быстрый, но он не может учиться. Поэтому для вас **освоить Пайтон — гораздо проще, чем компьютеру выучить английский.**

Общение с Пайтоном

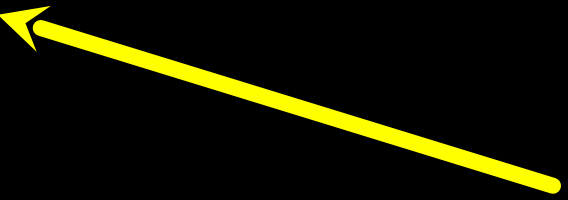
```
csev$ python3
```

```
Python 3.5.1 (v3.5.1:37a07cee5969, Dec 5 2015, 21:12:44)
```

```
[GCC 4.2.1 (Apple Inc. build 5666) (dot 3)] on darwinType
```

```
"help", "copyright", "credits" or "license" for more information.
```

```
>>>
```



Что дальше?

```
csev$ python3
```

```
Python 3.5.1 (v3.5.1:37a07cee5969, Dec 5 2015, 21:12:44)
```

```
[GCC 4.2.1 (Apple Inc. build 5666) (dot 3)] on darwinType
```

```
"help", "copyright", "credits" or "license" for more information.
```

```
>>> x = 1
```

```
>>> print(x)
```

```
1
```

```
>>> x = x + 1
```

```
>>> print(x)
```

```
2
```

```
>>> exit()
```

Это отличный тест, чтобы убедиться, что среда разработки на Пайтон установлена корректно. Обратите внимание, что команда `quit()` также выполняет завершение интерактивного сеанса.

Как нам общаться?

Элементы языка Пайтон

- **Лексика / Слова** — Переменные и Ключевые слова (Глава 2)
- **Структура предложения** — допустимые синтаксические шаблоны (Главы 3-5)
- **Story структура** — составление программы для определенной цели

```
name = input('Введите имя файла:')
handle = open(name)

counts = dict()
for line in handle:
    words = line.split()
    for word in words:
        counts[word] = counts.get(word, 0) + 1

bigcount = None
bigword = None
for word, count in counts.items():
    if bigcount is None or count > bigcount:
        bigword = word
        bigcount = count

print(bigword, bigcount)
```

Короткая «история»
о том, как подсчитать
слова в файле,
используя Пайтон

```
python words.py
Введите имя файла:
words.txt
to 16
```

Ключевые слова

Не стоит использовать **ключевые слова** в качестве имен переменных и других идентификаторов

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

Предложения или линии кода

`x = 2`



Присвоение значения

`x = x + 2`



Присвоение с выражением

`print(x)`



Функция вывода

Переменная

Оператор

Константа

Функция

Параграфы программы

Пайтон-скрипты

- Интерактивная среда Пайтон отлично подходит для экспериментов и небольших программ длиной в 3-4 строки;
- Большинство программ гораздо длиннее, поэтому мы пишем код в файлах, а затем просим Пайтон выполнить команды из этих файлов;
- Можно сказать, мы даем Пайтону «сценарий»;
- Мы добавляем расширение “.py” к концу файлов, чтобы обозначить, что они содержат команды на Пайтон.

Интерактивная среда или Скрипт

- Интерактивная среда

- Вы вводите непосредственно в Пайтон по одной строке за раз и он отвечает

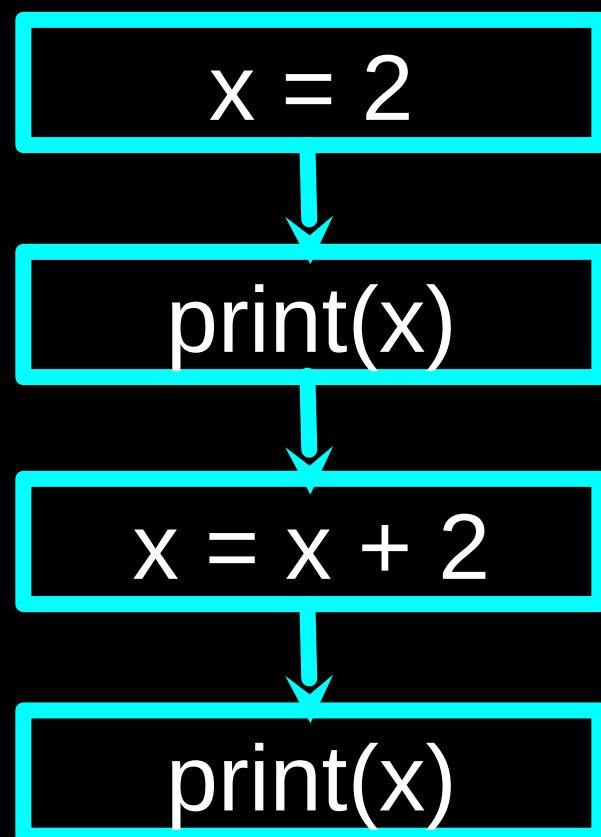
- Скрипт

- Вы вводите последовательность инструкций (строк) в файл, используя текстовый редактор, и поручаете Пайтону выполнить эти инструкции

Шаги программы или ход программы

- Подобно рецепту или инструкции по установке, программа — это **последовательность** шагов, исполняемых в определенном порядке;
- Некоторые шаги могут быть **условными**, их можно пропустить;
- Иногда бывает необходимо **повторить** шаг или группу шагов;
- Иногда мы храним последовательность шагов, чтобы использовать их снова по мере необходимости в разных местах программы (Глава 4).

Последовательные шаги



Программа:

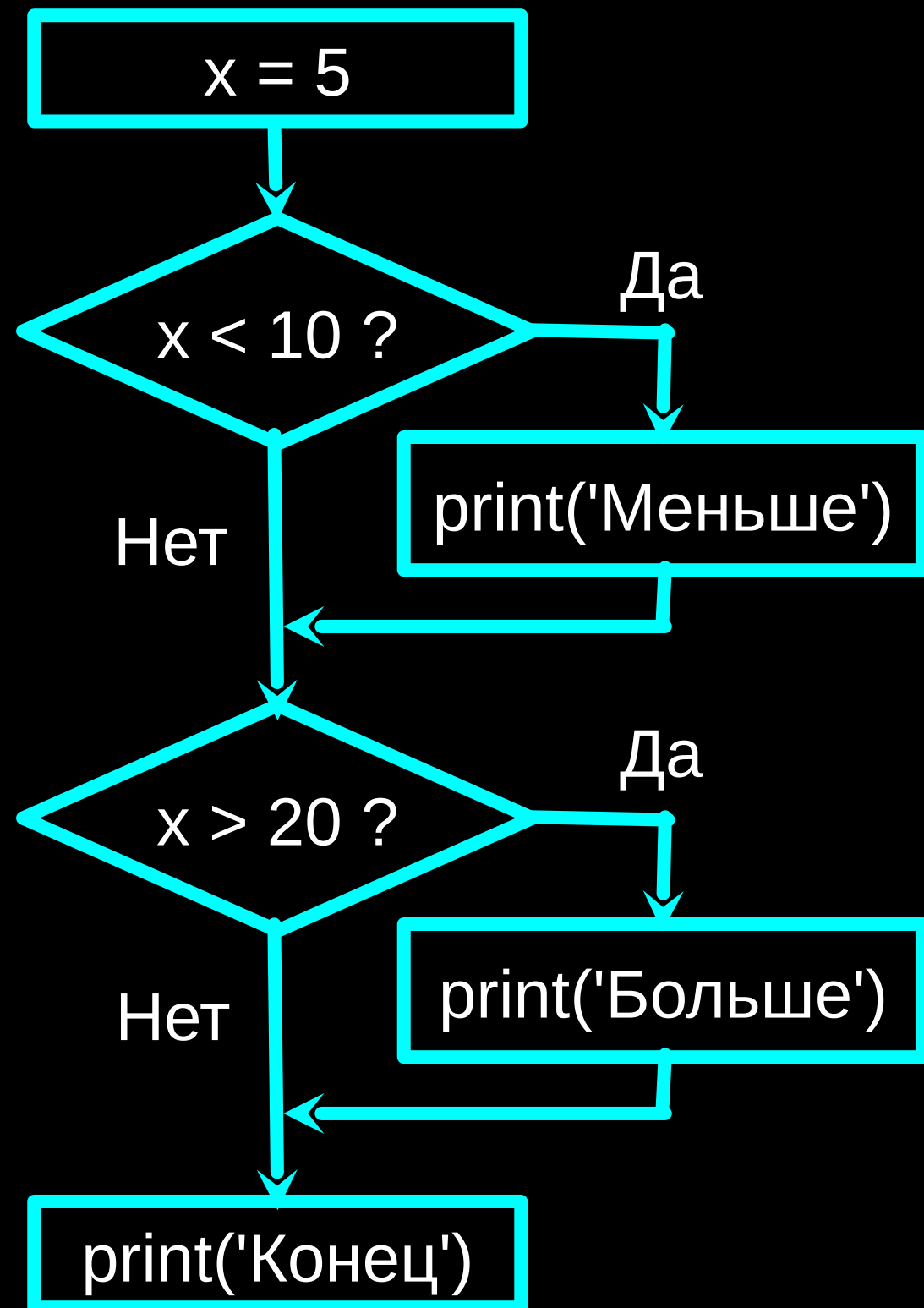
```
x = 2  
print(x)  
x = x + 2  
print(x)
```

Результат:

```
2  
4
```

В процессе выполнения программа движется от одного шага к другому. Как программисты, мы устанавливаем «пути», по которым следует программа.

Шаги-условия



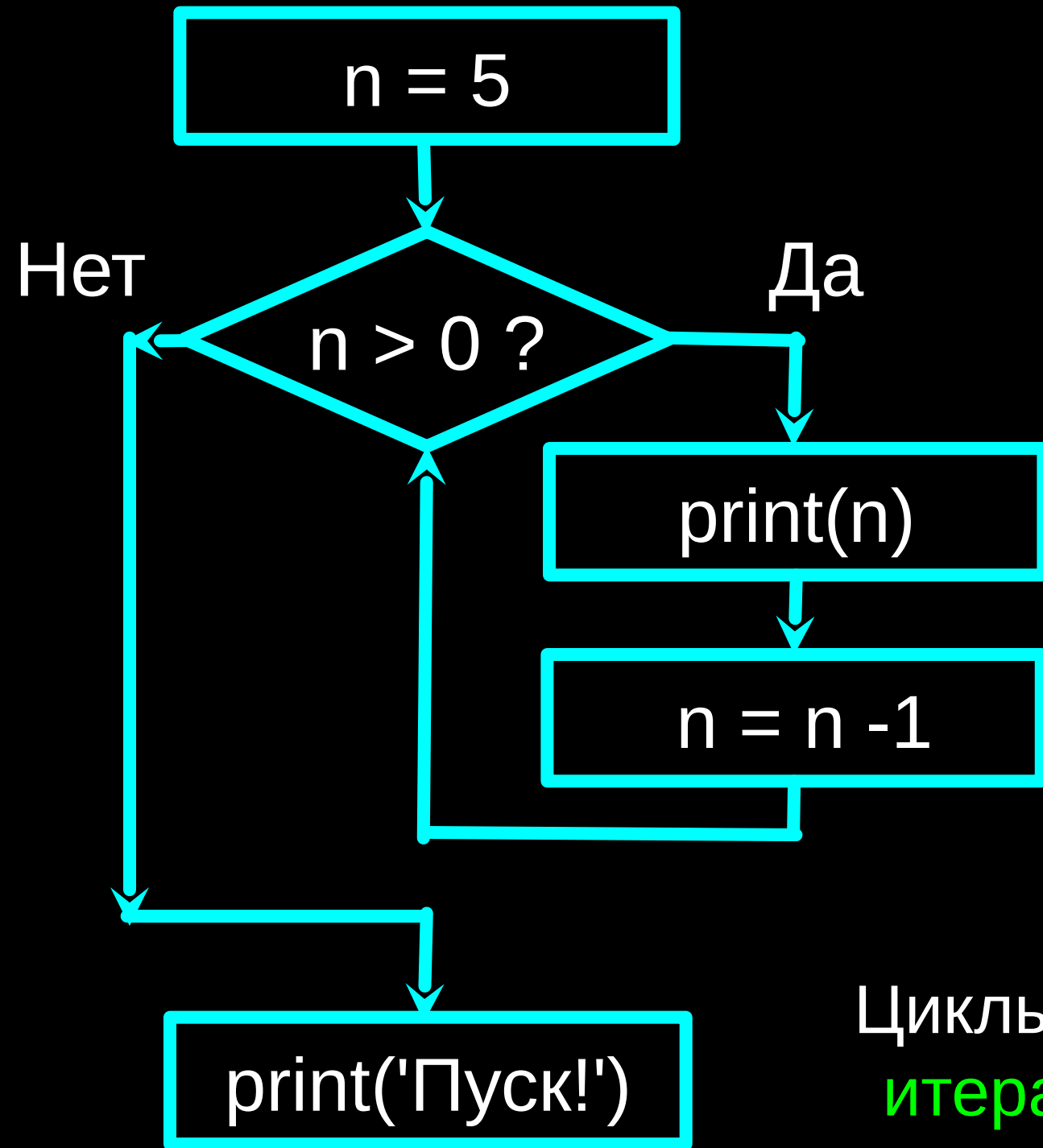
Программа:

```
x = 5
if x < 10:
    print('Меньше')
if x > 20:
    print('Больше')
print('Конец')
```

Результат:

Меньше
Конец

Повторяющиеся шаги



Программа:

```
n = 5
while n > 0 :
    print(n)
    n = n - 1
print('Пуск!')
```

Результат:

5
4
3
2
1
Пуск!

Циклы (повторяющиеся шаги) включают **переменные итерации**, которые изменяются с каждым проходом цикла.

```
name = input('Введите имя файла:')
handle = open(name, 'r')

counts = dict()
for line in handle:
    words = line.split()
    for word in words:
        counts[word] = counts.get(word,0) + 1

bigcount = None
bigword = None
for word,count in counts.items():
    if bigcount is None or count > bigcount:
        bigword = word
        bigcount = count

print(bigword, bigcount)
```

Последовательный

Повторяющийся

Условный

```
name = input('Введите имя файла:')
handle = open(name, 'r')

counts = dict()
for line in handle:
    words = line.split()
    for word in words:
        counts[word] = counts.get(word, 0) + 1

bigcount = None
bigword = None
for word, count in counts.items():
    if bigcount is None or count > bigcount:
        bigword = word
        bigcount = count

print(bigword, bigcount)
```

Короткая «история» о том, как с помощью Python подсчитать количество слов в файле

Слово, используемое для чтения данных пользователя

Предложение, обновляющее значение счетчика слова

Параграф о том, как найти самый длинный элемент в списке

Резюме

- Это быстрый обзор **Главы 1**
- Мы будем возвращаться к этим концепциям на протяжении всего курса
- Сосредоточьтесь на общей картине

Авторы / Благодарности



Авторские права на эти слайды принадлежат Чарльзу Р. Северансу (www.dr-chuck.com), 2010 г., Школа Информации Мичиганского Университета и доступны по лицензии Creative Commons Attribution 4.0 License. Пожалуйста, сохраняйте этот слайд во всех копиях этого документа, в соответствии с требованиями Лицензии. Если вы внесли изменения, добавьте свое имя или организацию в список участников на этой странице.

Исходная разработка: Чарльз Северанс, Школа Информации Мичиганского Университета.

Перевод выполнила Фомкина Виолетта.

... Добавьте сюда новых авторов и переводчиков

Продолжение следует...