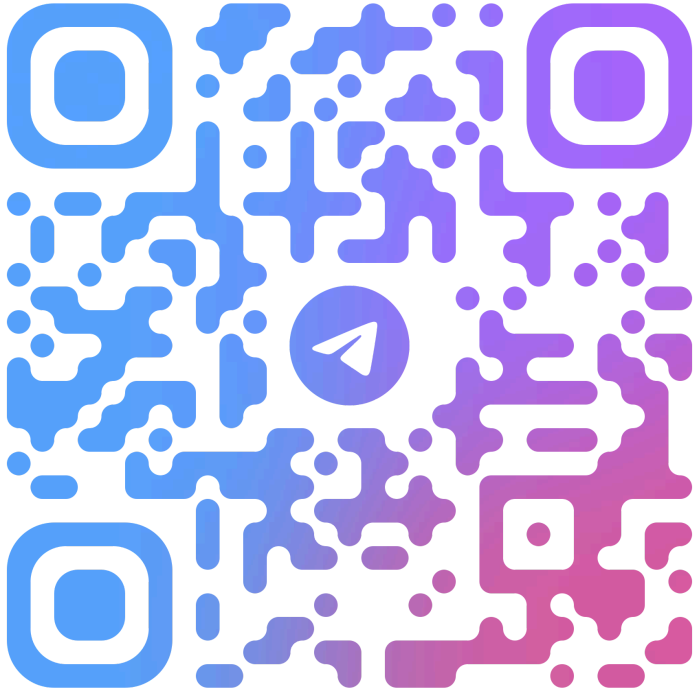
Университет Сириус Колледж

Основы программирования
Лекция 1. Введение

Введение. Ввод-вывод. Переменные. Типы данных. Операторы. Строки.

Университет Сириус Колледж

О курсе



Университет Сириус Колледж

Система оценивания и структура курса

Университет Сириус Колледж

Система оценивания и структура курса

- бально-рейтинговая система

Университет Сириус Колледж

Система оценивания и структура курса

- бально-рейтинговая система
- максимум 10 баллов

Университет Сириус Колледж

Система оценивания и структура курса

- бально-рейтинговая система
- максимум 10 баллов
  - Неудовлетворительно: 0-3

Университет Сириус Колледж

Система оценивания и структура курса

- бально-рейтинговая система
- максимум 10 баллов
  - Неудовлетворительно: 0-3
  - Удовлетворительно: 4-5

Университет Сириус Колледж

Система оценивания и структура курса

- бально-рейтинговая система
- максимум 10 баллов
  - Неудовлетворительно: 0-3
  - Удовлетворительно: 4-5
  - Хорошо: 6-7

Университет Сириус Колледж

Система оценивания и структура курса

- бально-рейтинговая система
- максимум 10 баллов
  - Неудовлетворительно: 0-3
  - Удовлетворительно: 4-5
  - Хорошо: 6-7
  - Отлично: 8-10

Университет Сириус Колледж

Система оценивания и структура курса

- бально-рейтинговая система
- максимум 10 баллов
  - Неудовлетворительно: 0-3
  - Удовлетворительно: 4-5
  - Хорошо: 6-7
  - Отлично: 8-10
- 50% практические работы (~14 работ)

Университет Сириус Колледж

Система оценивания и структура курса

- бально-рейтинговая система
- максимум 10 баллов
  - Неудовлетворительно: 0-3
  - Удовлетворительно: 4-5
  - Хорошо: 6-7
  - Отлично: 8-10
- 50% практические работы (~14 работ)
- 50% зачет

Университет Сириус Колледж

Язык Python

- Создан в 1991 г.
- Создатель: Гвидо Ван Россум
- Мультипарадигменный
  - ООП
  - структурное
  - функциональное,
  - метапрограммирование
- Интерпретируемый (cpython)
- Использует динамическую типизацию
- Официальный репозиторий: github.com/python
- Официальная документация: docs.python.org

Университет Сириус Колледж

Рейтинг TIOBE

Aug 2025	Aug 2024	Change	Programming Language	Ratings	Change
1	1		Python	26.14%	+8.10%
2	2		C++	9.18%	-0.86%
3	3		C	9.03%	-0.15%
4	4		Java	8.59%	-0.58%
5	5		C#	5.52%	-0.87%

Университет Сириус Колледж

Zen of Python

Дзен языка Python всегда можно посмотреть, вызвав интерпретатор и выполнив код `import this`

Университет Сириус Колледж

Zen of Python

1. Beautiful is better than ugly.
2. Explicit is better than implicit.
3. Simple is better than complex.
4. Complex is better than complicated.
5. Flat is better than nested.
6. Sparse is better than dense.
7. Readability counts.
8. Special cases aren't special enough to break the rules.
9. Although practicality beats purity.
10. Errors should never pass silently.
11. Unless explicitly silenced.
12. In the face of ambiguity, refuse the temptation to guess.
13. There should be one-- and preferably only one --obvious way to do it.
14. Although that way may not be obvious at first unless you're Dutch.
15. Now is better than never.
16. Although never is often better than *right* now.
17. If the implementation is hard to explain, it's a bad idea.
18. If the implementation is easy to explain, it may be a good idea.
19. Namespaces are one honking great idea – let's do more of those!

Университет Сириус Колледж

Zen of Python

1. Красивое лучше, чем уродливое.
2. Явное лучше, чем неявное.
3. Простое лучше, чем сложное.
4. Сложное лучше, чем запутанное.
5. Плоское лучше, чем вложенное.
6. Разреженное лучше, чем плотное.
7. Читаемость имеет значение.
8. Особые случаи не настолько особые, чтобы нарушать правила.
9. При этом практичность важнее безупречности.
10. Ошибки никогда не должны замалчиваться.
11. Если они не замалчиваются явно.
12. Встретив двусмысленность, отбрось искушение угадать.
13. Должен существовать один и, желательно, только один очевидный способ сделать это.
14. Хотя он поначалу может быть и не очевиден, если вы не голландец.
15. Сейчас лучше, чем никогда.
16. Хотя никогда зачастую лучше, чем *прямо* сейчас.
17. Если реализацию сложно объяснить — идея плоха.
18. Если реализацию легко объяснить — идея, возможно, хороша.
19. Пространства имён — отличная штука! Будем делать их больше!

Код на Python

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

Вывод:

```
Hello, World!
```

Код на Python

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

Вывод:

```
File "", line 1
  print(Hi there!)
                  ^
SyntaxError: invalid syntax
```

Код на Python

```
1  print("Добро пожаловать на курс Основы программирования!")
2  print("Для начала научимся работать с командой print.")
3  print("Это программа выведет на экран 3 строки.")
```

Вывод:

```
Добро пожаловать на курс Основы программирования!
Для начала научимся работать с командой print.
Это программа выведет на экран 3 строки.
```

Университет Сириус Колледж

Арифметические операции

```
1  print(2 + 5)
2  print(3 * 3)
3  print(2 + 2 * 10)
```

Вывод:

```
7
9
22
```

Университет Сириус Колледж

Арифметические операции

```
1  print(2 + 2 * 10)
2  print("2 + 2 * 10")
```

Вывод:

```
22
2 + 2 * 10
```

Университет Сириус Колледж

Комментарии

```
1  print("Часов в году:")
2  # в году 365 дней, в каждом дне 24 часа
3  print(365 * 24)
```

Вывод:

```
Hours in a year:
8760
```

Университет Сириус Колледж

Комментарии

```
1  print("Часов в году:")
2  print(365 * 24) # в году 365 дней, в каждом дне 24 часа
```

Вывод:

```
Hours in a year:
8760
```

Университет Сириус Колледж

Ввод

```
1  name = input("Введите ваше имя: ")
2  print("Привет," + name + "!")
```

Вывод:

```
Введите ваше имя: Поликарп
Привет, Поликарп!
```

Университет Сириус Колледж

Ввод

```
1  name = input("Введите ваше имя: ")
2  print("Привет," + name + "!")
```

Вывод:

```
Введите ваше имя: 123
Привет, 123!
```

Ввод

```python
name = input("Введите ваше имя: ")
print("Привет," + name + "!")
```

Вывод:

```
Введите ваше имя: !@#$%^&
Привет, !@#$%^&!
```

Университет Сириус Колледж

Ввод

```python
name = input("Введите ваше имя: ")
print("Привет," + name + "!")
print(name + " - отличное имя!")
```

Вывод:

```
Введите ваше имя: Поликарп
Привет, Поликарп!
Поликарп - отличное имя!
```

Ввод

```
1  name = input("Введите ваше имя: ")
2  print("Привет," + name + "! Давай проверим, тебя зовут " + name + "?")
```

Вывод:

```
Введите ваше имя: Поликарп
Привет, Поликарп! Давай проверим, тебя зовут Поликарп?
```

Ввод

```python
name = input("Введите ваше имя: ")
email = input("Введите ваш email: ")
nickname = input("Введите ваш никнейм: ")

print("Привет, " + name + "!")
print("Твой email: " + email)
print("Твой никнейм: " + nickname)
```

Вывод:

```
Введите ваше имя: Поликарп
Введите ваш email: polykarp@mail.com
Введите ваш никнейм: poly
Привет, Поликарп!
Твой email: polykarp@mail.com
Твой никнейм: poly
```

Ввод

```python
address = input("Введите ваш адрес: ")
print("Ваш адрес: " + address)

address = input("Введите новый адрес: ")
print("Ваш адрес: " + address)
```

Вывод:

```
Введите ваш адрес: ул. Ленина, д. 1
Ваш адрес: ул. Ленина, д. 1
Введите новый адрес: ул. Пушкина, д. 2
Ваш адрес: ул. Пушкина, д. 2
```

Ввод

```
1  address = input("Введите ваш адрес: ")
2  address = input("Введите новый адрес: ")
3  print("Ваш адрес: " + address)
```

Вывод:

```
Введите ваш адрес: ул. Ленина, д. 1
Введите новый адрес: ул. Пушкина, д. 2
Ваш адрес: ул. Пушкина, д. 2
```

Университет Сириус Колледж

Переменные

```
1  variable_name = ...
```

Университет Сириус Колледж

Переменные

```
1  name = input("Введите ваше имя: ")
2
3  print("Привет," + name + "!")
```

Университет Сириус Колледж

Переменные

```
1  name = "Поликарп"
2  surname = "Петров"
3
4  full_name = name + " " + surname
5
6  print(full_name)
```

Университет Сириус Колледж

Переменные

```
1  word = "Hello"
2  print(word)
3
4  word = "World"
5  print(word)
6
7  word = "!"
8  print(word)
```

Переменные

```
1  word = "Hello"
2  print(word)
3
4  word = word + ", World!"
5  print(word)
```

Университет Сириус Колледж

Имена переменных

Университет Сириус Колледж

Имена переменных

- Состоят из букв, цифр и символов подчеркивания

Университет Сириус Колледж

Имена переменных

- Состоят из букв, цифр и символов подчеркивания
- Не могут начинаться с цифры

Университет Сириус Колледж

Имена переменных

- Состоят из букв, цифр и символов подчеркивания
- Не могут начинаться с цифры
- Чувствительны к регистру (`Variable` , `variable` и `VARIABLE` - разные переменные)

Университет Сириус Колледж

Имена переменных

- Состоят из букв, цифр и символов подчеркивания
- Не могут начинаться с цифры
- Чувствительны к регистру (`Variable` , `variable` и `VARIABLE` - разные переменные)
- Не могут быть ключевыми словами языка Python (например, `if` , `for` , `while` , `import` и т.д.)

Университет Сириус Колледж

Имена переменных

- Состоят из букв, цифр и символов подчеркивания
- Не могут начинаться с цифры
- Чувствительны к регистру (`Variable` , `variable` и `VARIABLE` - разные переменные)
- Не могут быть ключевыми словами языка Python (например, `if` , `for` , `while` , `import` и т.д.)
- Рекомендуется использовать "говорящие" имена (например, `user_name` вместо `un`)

Университет Сириус Колледж

Имена переменных

- Состоят из букв, цифр и символов подчеркивания
- Не могут начинаться с цифры
- Чувствительны к регистру (`Variable` , `variable` и `VARIABLE` - разные переменные)
- Не могут быть ключевыми словами языка Python (например, `if` , `for` , `while` , `import` и т.д.)
- Рекомендуется использовать "говорящие" имена (например, `user_name` вместо `un`)
- Рекомендуется использовать стиль `snake_case` (слова разделяются символом подчеркивания)

Университет Сириус Колледж

Имена переменных

- Состоят из букв, цифр и символов подчеркивания
- Не могут начинаться с цифры
- Чувствительны к регистру (`Variable` , `variable` и `VARIABLE` - разные переменные)
- Не могут быть ключевыми словами языка Python (например, `if` , `for` , `while` , `import` и т.д.)
- Рекомендуется использовать "говорящие" имена (например, `user_name` вместо `un`)
- Рекомендуется использовать стиль `snake_case` (слова разделяются символом подчеркивания)
- Рекомендуется использовать английские имена (хотя это не обязательно)

Университет Сириус Колледж

Имена переменных

- Состоят из букв, цифр и символов подчеркивания
- Не могут начинаться с цифры
- Чувствительны к регистру (`Variable` , `variable` и `VARIABLE` - разные переменные)
- Не могут быть ключевыми словами языка Python (например, `if` , `for` , `while` , `import` и т.д.)
- Рекомендуется использовать "говорящие" имена (например, `user_name` вместо `un`)
- Рекомендуется использовать стиль `snake_case` (слова разделяются символом подчеркивания)
- Рекомендуется использовать английские имена (хотя это не обязательно)
- Рекомендуется использовать строчные буквы (хотя это не обязательно)

Университет Сириус Колледж

Имена переменных

- Состоят из букв, цифр и символов подчеркивания
- Не могут начинаться с цифры
- Чувствительны к регистру (`Variable` , `variable` и `VARIABLE` - разные переменные)
- Не могут быть ключевыми словами языка Python (например, `if` , `for` , `while` , `import` и т.д.)
- Рекомендуется использовать "говорящие" имена (например, `user_name` вместо `un`)
- Рекомендуется использовать стиль `snake_case` (слова разделяются символом подчеркивания)
- Рекомендуется использовать английские имена (хотя это не обязательно)
- Рекомендуется использовать строчные буквы (хотя это не обязательно)
- Рекомендуется избегать слишком длинных имен (хотя это не обязательно)

Университет Сириус Колледж

Имена переменных

- Состоят из букв, цифр и символов подчеркивания
- Не могут начинаться с цифры
- Чувствительны к регистру (`Variable` , `variable` и `VARIABLE` - разные переменные)
- Не могут быть ключевыми словами языка Python (например, `if` , `for` , `while` , `import` и т.д.)
- Рекомендуется использовать "говорящие" имена (например, `user_name` вместо `un`)
- Рекомендуется использовать стиль `snake_case` (слова разделяются символом подчеркивания)
- Рекомендуется использовать английские имена (хотя это не обязательно)
- Рекомендуется использовать строчные буквы (хотя это не обязательно)
- Рекомендуется избегать слишком длинных имен (хотя это не обязательно)
- Рекомендуется избегать слишком коротких имен (хотя это не обязательно)

Университет Сириус Колледж

Имена переменных

- Состоят из букв, цифр и символов подчеркивания
- Не могут начинаться с цифры
- Чувствительны к регистру (`Variable` , `variable` и `VARIABLE` - разные переменные)
- Не могут быть ключевыми словами языка Python (например, `if` , `for` , `while` , `import` и т.д.)
- Рекомендуется использовать "говорящие" имена (например, `user_name` вместо `un`)
- Рекомендуется использовать стиль `snake_case` (слова разделяются символом подчеркивания)
- Рекомендуется использовать английские имена (хотя это не обязательно)
- Рекомендуется использовать строчные буквы (хотя это не обязательно)
- Рекомендуется избегать слишком длинных имен (хотя это не обязательно)
- Рекомендуется избегать слишком коротких имен (хотя это не обязательно)
- Рекомендуется избегать имен, которые могут быть легко спутаны (например, `l` и `1` , `O` и `0`)

Университет Сириус Колледж

Целые числа (int)

```
1  age = 25
2
3  print(age)
```

Вывод:

25

Университет Сириус Колледж

Целые числа (int)

```
1  number1 = 10
2  number2 = "15"
3
4  print(number1)
5  print(number2)
```

Вывод:

```
10
15
```

Университет Сириус Колледж

Целые числа (int)

```
1  number1 = 10
2  number2 = "15"
3
4  print(number1 + number1)
5  print(number2 + number2)
```

Вывод:

```
20
1515
```

Целые числа (int)

```
1  number = "10"
2
3  print(number / 2)
```

Вывод:

```
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: unsupported operand type(s) for /: 'str' and 'int'
```

Университет Сириус Колледж

Объединение значений при выводе

```
1  result = 10 * 25
2
3  print("Результат равен " + result)
```

Вывод:

```
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: can only concatenate str (not "int") to str
```

Университет Сириус Колледж

Объединение значений при выводе

```
1  result = 10 * 25
2
3  print("Результат равен " + str(result))
```

Вывод:

```
Результат равен 250
```

Университет Сириус Колледж

Интерполяция строк (f-строки)

```
1  result = 10 * 25
2
3  print(f"Результат равен {result}")
```

Вывод:

```
Результат равен 250
```

Университет Сириус Колледж

Интерполяция строк (f-строки)

```
1  name = "Поликарп"
2  age = 25
3  city = "Сириус"
4
5  print("Меня зовут", name, ", мне", age, "лет и я живу в", city)
```

Вывод:

```
Меня зовут Поликарп , мне 25 лет и я живу в Сириус
```

Университет Сириус Колледж

Интерполяция строк (f-строки)

```python
name = "Поликарп"
age = 25
city = "Сириус"

print(f"Меня зовут {name}, мне {age} лет и я живу в {city}")
```

Вывод:

```
Меня зовут Поликарп, мне 25 лет и я живу в Сириус
```

float

```python
number1 = 2.5
number2 = -1.25
number3 = 3.62

mean = (number1 + number2 + number3) / 3
print(f"Mean: {mean}")
```

Вывод:

```
Mean: 1.6233333333333333
```

Университет Сириус Колледж

Арифметические операции

Оператор	Действие	Пример	Результат
+	Сложение	`4 + 2`	`6`
-	Вычитание	`10 - 2.5`	`7.5`
*	Умножение	`-2 * 123`	`-246`
/	Деление	`9 / 2`	`4.5`
//	Целочисленное деление	`9 // 2`	`4`
%	Остаток от деления	`9 % 2`	`1`
**	Возведение в степень	`2**3`	`8`

Университет Сириус Колледж

Порядок выполнения операций

1. Скобки `( )`
2. Возведение в степень `**`
3. Умножение `*`, Деление `/`, Целочисленное деление `//`, Остаток от деления `%`
4. Сложение `+`, Вычитание `-`
5. Операции выполняются слева направо

Университет Сириус Колледж

Приведение типов

```
1  number1 = "10"
2  number2 = 5
3
4  print(number1 + number2)
```

Вывод:

```
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: can only concatenate str (not "int") to str
```

Университет Сириус Колледж

Приведение типов

```
1  number1 = "10"
2  number2 = 5
3
4  print(int(number1) + number2)
```

Вывод:

15

Университет Сириус Колледж

Приведение типов

```
1  number1 = "10.5"
2  number2 = 5
3
4  print(float(number1) + number2)
```

Вывод:

```
15.5
```

Университет Сириус Колледж

Приведение типов

```
1  number1 = 105
2
3  print(str(number1) + " is a number")
```

Вывод:

```
105 is a number
```

Университет Сириус Колледж

Приведение типов

```python
number1 = input("Enter a number: ")
number1 = int(number1)

print(f"You entered: {number1}, which is of type {type(number1)}")
print(number1 + 5)
```

Вывод:

```
Enter a number: 20
You entered: 20, which is of type <class 'int'>
25
```

Университет Сириус Колледж

Приведение типов

```python
number1 = int(input("Enter a number: "))

print(f"You entered: {number1}, which is of type {type(number1)}")
print(number1 + 5)
```

Вывод:

```
Enter a number: 20
You entered: 20, which is of type <class 'int'>
25
```