

## Знакомство с Python (продолжение)

Для написания кода можно воспользоваться онлайн-компилятором по ссылке:  
[https://www.onlinegdb.com/online\\_python\\_compiler](https://www.onlinegdb.com/online_python_compiler)

Необходимо ознакомиться с теорией и выполнить задачи в компиляторе. После выполнения делать скриншот выполненной задачи.

### Именованние переменных

В именах переменных используйте только **латинский алфавит, цифры и подчеркивание**. Остальные символы, в том числе буквы кириллицы, могут привести к ошибкам. Они создают путаницу: например, английская *x* и русская *х* выглядят одинаково. Если название состоит из нескольких слов, разделяйте их символом подчёркивания: `new_message`. Такой стиль написания называется *snake case*, потому что слова могут получаться *очень\_длинные\_как\_змея\_или\_даже\_как\_две*.

Python допускает использование цифр в именах переменных, но не на первой позиции:

```
snake1 = "питон"  
snake2 = "удав"  
long_snake = "длинный змей"  
print(long_snake + " — это может быть " + snake1 + " или " + snake2)
```

#### Важно

Как выбрать имя переменной?

Составляйте названия переменных из английских слов. Когда ваша программа вырастет до многих сотен строк, названия-слова помогут быстро сориентироваться в коде. Сравните сами: переменную с названием урока можно назвать буквой *l*, а можно — словом `lesson`. Второй вариант лучше: ведь то, что когда-то разработчик сократил `lesson` до одной буквы, со временем забудут и он, и его коллеги. Код станет сложнее для чтения. Не рекомендуется использовать русские слова в английской раскладке. Рано или поздно ваш код будет читать человек, не владеющий русским. Он может не понять, что к чему. Сразу называйте переменные по-английски: `child`, а не транслитерацией — `rebyonok`.

### Задача 1

Выберите правильные названия переменных для `imia`, `familia` и переименуйте их.

<p>Код Python Скопировать и выполнить задачу 1</p> <p>Подсказка Если название состоит из нескольких слов, разделяйте их символом нижнего подчёркивания: <code>first_name</code>. Обязательно поменяйте названия переменных везде, где они встречаются в коде.</p>	<pre>name = 'Данил' family = 'Марков' print('Меня зовут ' + name + ' ' + family + '.')</pre>
---	--

## Вывод на экран

Вы уже умеете выводить на экран текст, складывая строки. Но можно напечатать и без оператора `+`: в скобках функции `print()` перечислите через запятую аргументы, которые она должна напечатать:

```
weather = 'облачно'  
print('На улице сейчас', weather)  
# Будет напечатано:  
# На улице сейчас облачно
```

Запятая между аргументами по умолчанию заменяется на пробел.

```
print('За', 'окном', 'метель')  
# Будет напечатано:  
# За окном метель
```

Это значит, что в сообщения Анфисы можно не добавлять пробелы, когда вы составляете сложную фразу. Достаточно разделить аргументы функции `print()` запятой. И, что приятно, количество этих аргументов не ограничено:

```
current_hour = 14  
messages_count = 12  
print('К', current_hour, 'ч.', 'у тебя накопилось', messages_count, 'новых сообщений.')
```

```
# Будет напечатано:  
# К 14 ч. у тебя накопилось 12 новых сообщений.
```

Неожиданно обнаружилось ещё одно приятное свойство функции `print()`: посмотрите на код, там через запятую перечислены и **строки**, и **числа**, и все они печатаются без проблем.

А ведь если бы мы захотели сначала подготовить полную строку для печати, а потом передать её в функцию `print()` — нам пришлось бы преобразовывать число в строку:

```
messages_count = 100500  
# Складывать строки и числа нельзя,  
# потому придётся преобразовать messages_count в строку:  
how_many_messages = 'У тебя ' + str(messages_count) + ' новых сообщений.'  
print(how_many_messages)
```

```
# Будет напечатано:  
# У тебя 100500 новых сообщений.
```

### Задача 2

Начнём учить Анфису разговаривать о погоде.

Напечатайте на экран два сообщения о погоде в следующем формате:

Сегодня ...  
Температура воздуха ... градусов

Код Python Скопировать и выполнить задачу 2	<pre>temperature = -25 weather = 'солнечно'  # напишите ваш код ниже print(...) print(...)</pre>
--	--

## Дробные числа

Числа бывают целыми и дробными. Для десятичных дробей (иначе их называют «числа с плавающей запятой») в Python есть специальный тип данных — **float**.

```
first = 87.2
second = 50.2
third = 50.242
print(first + second + third)
# Будет напечатано: 187.642
```

Дробные числа преобразуют в строки так же, как и целые — функцией `str()`:

```
first = 87.2
second = 50.2
third = 50.242
print(str(first) + str(second) + str(third))
# Будет напечатано: 87.250.250.242
```

Превратив числа в строки, мы получили один из IP-адресов серверов Яндекса *ya.ru*.

```
D:\Users\Anfisa>ping ya.ru

Pinging ya.ru [87.250.250.242] with 32 bytes of data:
Reply from 87.250.250.242: bytes=32 time=5ms TTL=57
```

Можно преобразовать строку в дробное число, для этого есть функция float():

```
first = '87.2' # Строка
second = '50.2' # Тоже строка
third = '50.242' # И это строка
print(float(first) + float(second) + float(third)) # А в итоге получится число!
# Будет напечатано: 187.642
```

Дробные числа приводят к целым функцией int(). Обратите внимание: int() **не округляет** числа по правилам арифметики, а просто отбрасывает дробную часть.

*# Функция int() просто убирает всё, что после запятой. Вместе с запятой:*

```
a = int(3.14)
print(a)
# Будет напечатано: 3
```

---

```
a = int(2.72)
```

```
print(a)
# Будет напечатано: 2
```

---

*# Функция int() одинаково работает с положительными и отрицательными числами*

```
a = int(-3.14)
print(a)
# Будет напечатано: -3
```

Можно сделать несколько преобразований в одной строке: сначала превратить дробь в целое число, а затем преобразовать в строку:

```
fraction = 1.5 # Дробь
print("Целая часть = " + str(int(fraction)))
# Вернётся строка, представляющая собой целочисленную часть дроби.
```

Получится так:

Целая часть = 1

Обратите внимание: чтобы не сбиться в количестве скобок (закрывающих должно быть столько же, сколько и открывающих), в тренажёре пары скобок подсвечиваются, когда рядом оказывается курсор:

```
print("Целая часть = " + str(int(fraction)))
```

↑  
↑  
так помогает выделить парные скобки редактор кода

Для операций с целыми и дробными числами приводить их к одному типу не нужно, тут Python разберётся сам:

```
a = 16 * 2.2 + 7 - 0.2
```

```
print(a)
```

```
# Будет напечатано: 42.0
```

Что значит «плавающая запятая» и куда она плывет?

Дробные числа называют «числами с плавающей запятой», потому что запятая «плавает» по числу, когда его представляют в виде произведения значащей части и степени.

Например, число 3,14159 можно записать следующим образом:

$314,159 \cdot 10^{-2}$  —  $314,159 \cdot 10^{-2}$

$0,0314159 \cdot 10^2$  —  $0,0314159 \cdot 10^2$

$314159,0 \cdot 10^{-5}$  —  $314159,0 \cdot 10^{-5}$

В англоязычной литературе запятая называется точкой, *floating point*, потому что десятичные дроби пишутся через точку.

### Задача 3

Скорость света равна 1 079 252 848,8 километров в час. Научите Анфису переводить скорость в километры в секунду.

На основе переменной `speed_kmh` (скорость света в км/ч) вычислите значение переменной `speed_kms` (скорость света в км/с).

Значение переменной `speed_kms` приведите к целому типу: отбросьте дробную часть.

Код Python

Скопировать и выполнить задачу 3

Подсказка

Чтобы получить скорость света в км/с, разделите скорость в км/ч на количество секунд в часе.

В часе 60 минут, в минуте 60 секунд, значит, в часе 3600 секунд ( $60 \cdot 60$ ).

```
speed_kms = speed_kmh / 3600
```

```
speed_kmh = 1079252848.8
```

```
# переменную speed_kms сделайте типа int
```

```
speed_kms = ...
```

```
print('Скорость света равна', speed_kms, 'км/с')
```

Чтобы превратить дробное число в целое — оберните ваши вычисления в функцию <code>int()</code> .	
--	--

#### Задача 4

Длина Питона из известного советского мультфильма равна 38 попугаев и одно попугайское крылышко. Будем считать, что крылышко равно 0.2 попугая.

В железнодорожный вагон можно уложить по длине шесть с половиной Питонов.

Помогите Анфисе выяснить, какое количество попугаев можно уложить вдоль вагона.

Попугаи бывают только целые, это важно. Поэтому переменная `result` должна быть типа `int`.

Код Python	<pre>snake = '38.2' # Длина Питона в попугаях length = 6.5 # Количество Питонов,                которых можно уложить вдоль вагона.  # Напишите формулу для вычисления # количества попугаев result = ...  print('В вагоне можно положить в ряд',       result, 'попугаев')</pre>
------------	---

Скопировать и выполнить задачу 4

Подсказка

- преобразуйте строку `snake` в число функцией `float()`;
- длина вагона в попугаях — это количество Питонов `length`, умноженное на `float(snake)`;
- превратите результат в целое число функцией `int()`.