Основные понятия модуля

Python — это высокоуровневый интерпретируемый кроссплатформенный язык программирования.

*Актуальная версия Python — **Python 3***

Документация по Python 👆

Переменная — именованная область памяти компьютера, адрес которой позволяет получить доступ к данным.

Алгоритм — набор последовательных действий, направленных на достижение поставленной цели или решение конкретной задачи.

Функция — фрагмент кода, к которому можно обратиться из любого другого места.

Функции, как правило, возвращают некоторое значение в качестве результата работы.

Аргументы — данные, которые необходимы функции для её работы.

Python — язык с **неявной сильной динамической типизацией**.

Динамическая типизация	**Сильная типизация**	**Неявная типизация**
Тип переменной определяется во время выполнения программы	Нельзя совершать операции над объектами разного типа без приведения их к одному типу	Не надо указывать тип переменной при её объявлении

Строки — неизменяемый тип данных, предназначенный для хранения текстовой информации.

Ввод и вывод информации

`print(аргумент)`	Печатает на экране данные, которые мы передали. Это может быть переменная или выражение.
`print(аргумент_1, аргумент_2, … , аргумент_n)`	Печатает переданные значения через пробел.
`input(подсказка_для_пользователя)`	Функция для ввода информации от пользователя. В круглых скобках можем написать строку-подсказку для пользователя о том, какую именно информацию мы ожидаем от него получить.

Codeboard

RUN — Запускает выполнение кода. Так вы можете проверить, как работает ваш код до отправки решения.

TEST — Запускает тесты для вашего решения. Таким образом проверяется логика решения.

SUBMIT — Отправляет код на проверку и запускает дополнительные тесты => результат и начисление баллов (при верном решении).

Присваивание

`a = 5`	Переменной `a` присвоили значение 5.
`b = a`	Переменной `b` присвоили значение переменной `a`.
`a, b = 5, 6`	Множественное присваивание: переменной `a` присвоили значение 5, переменной `b` присвоили значение 6.

Правила именования переменных

- Название переменной должно состоять только из букв, цифр и знаков подчёркивания _.
- Название переменной не может начинаться с цифры.

Типы данных

Тип данных	Изменяемость	Класс	Пример
Целые числа	-	`int`	`73` `0`
Числа с плавающей точкой	Нет	`float`	`3.14` `-2.79`
Строки	Нет	`str`	`"Hello, world!"` `"5"`
Логические переменные	Нет	`bool`	`True` `False`
Списки	Да	`list`	`[1,2,3,4]`
Кортежи	Нет	`tuple`	`('a','b','c')`
Словари	Да	`dict`	`{'a' : 1, 'b' : 2}`
Множества	Да	`set`	`{'a', 1, 'b', 2}`

Определение типа переменной и идентификатора объекта

- `type(n)` — тип переменной `n`.
- `id(n)` — уникальный идентификатор объекта, который хранится в переменной `n`.

Операции с целыми и вещественными числами

Сложение	+	7+5 = 12 3.14+1 = 4.14
Вычитание	-	7-5 = 2 3.14-1 = 2.14
Умножение	*	7*5 = 35 3.14*2 = 6.28
Возведение в степень	**	7**5 = 16807 3.14**2 = 9.8596
Деление	/	5/2 = 2.5 3.14/2 = 1.57
Целочисленное деление	//	7 // 5 = 1 3.14 // 2 = 1.0
Остаток от деления	%	7 % 5 = 2 3.14 % 2 = 1.14

Округление чисел

- `round(значение, количество_знаков_после_запятой)` — округляет число к заданной точности.

Значения логического типа данных

- `True` — Истина
- `False` — Ложь

Строки

`s = "Hello!"`	Задаём строку
`s[начало:конец:шаг]`	Срез строки
`s = "Hel" + "lo!"`	Сложение строк
`s = "Hello!"*n`	Дублирование значения строки n раз
`len(s)`	Длина строки
`find(substr)`	Метод для поиска подстроки в строке *Пример вызова:* `s.find('e')` возвращает индекс символа `'e'` в строке `s`
`isdigit()`	Метод возвращает `True`, если строка состоит только из цифр
`isalpha()`	Метод возвращает `True`, если строка состоит только из букв
`isalnum()`	Метод возвращает `True`, если строка состоит только из букв и цифр
`upper()`	Метод возвращает новую строку в верхнем регистре
`lower()`	Метод возвращает новую строку в нижнем регистре
`split(разделитель)`	Метод разбивает строку на части по **разделителю** (по умолчанию — пробел) и возвращает результат в виде списка
`'строка-разделитель'.join(список)`	Метод объединяет элементы списка в строку, вставляя между ними строку-разделитель

Форматирование строк

> Форматирование строк используется, когда нам необходимо вставлять в шаблон строки разные данные.
> Это можно сделать и с помощью соединения частей строк и данных, но с помощью приёмов форматирования делать это можно гораздо удобнее.

Способ создания форматированной строки	Пример задания шаблона
Метод `format()`	`'The {} currency rate on the date {} is {:.3f}'.format(currency, cur_date, rate)`
f-строки	`f'The {currency} currency rate on the date {cur_date} is {rate:.3f}'`