FAQ по частым ошибкам интерпретатора Python для начинающих

Этот конспект поможет вам разобраться с распространёнными ошибками, которые возникают при написании кода на Python. Каждый вопрос содержит развёрнутое объяснение и примеры.

Содержание:

SyntaxError: invalid syntax

Вопрос: почему появляется эта ошибка и как её исправить?

Ответ: ошибка возникает, когда Python не понимает ваш код из-за синтаксической ошибки.

Например:

```
if x = 5:  # Ошибка: вместо == использовано =     print(x)
```

Исправление: проверьте, нет ли опечаток, забытых скобок, двоеточий или неверных операторов.

IndentationError: expected an indented block

Вопрос: почему Python ругается на отступы?

Ответ: в Python отступы (пробелы или табы) определяют блоки кода. Ошибка появляется, если их нет:

```
if True:
print("Hello")  # Ошибка: нет отступа
```

Исправление: добавьте 4 пробела (рекомендуется) или таб после двоеточия.

NameError: name 'x' is not defined

Вопрос: почему Python не видит мою переменную?

Ответ: переменная не была объявлена до использования:

```
print(x)  # Ошибка: x не существует
```

Исправление: убедитесь, что переменная определена перед использованием.

```
x = 10
print(x)  # Ошибка: x не существует
```

TypeError: can only concatenate str (not "int") to str

Вопрос: почему нельзя сложить строку и число?

Ответ: Python не преобразует типы автоматически:

```
print("Возраст: " + 25)  # Ошибка
```

Исправление: преобразуйте число в строку:

```
print("Возраст: " + str(25))
```

IndexError: list index out of range

Вопрос: почему программа падает при обращении к списку?

Ответ: вы пытаетесь получить элемент за пределами списка:

```
lst = [1, 2, 3]
print(lst[3])  # Ошибка: индексы от 0 до 2
```

Исправление: проверьте длину списка (len(lst)) перед доступом.

KeyError: 'key_name'

Вопрос: почему словарь выдаёт ошибку при доступе по ключу?

Ответ: ключа не существует в словаре:

```
d = {"name": "Alice"}
print(d["age"])  # Ошибка
```

Исправление: используйте d.get("age", "default") или проверьте ключ через if "age" in d.

AttributeError: 'str' object has no attribute 'append'

Вопрос: почему нельзя добавить элемент в строку как в список?

Ответ: у строк нет метода append(), он есть только у списков:

```
s = "Hello"
 s.append("!")  # Ошибка
```

Исправление: используйте конкатенацию: s += "!".

ValueError: invalid literal for int() with base 10: 'abc'

Вопрос: почему int() не преобразует строку в число?

Ответ: функция int() ожидает цифры, а не буквы:

```
int("abc")  # Ошибка
```

Исправление: проверьте, что строка содержит только числа: "123".

ZeroDivisionError: division by zero

Вопрос: почему программа падает при делении на ноль?

Ответ: делить на ноль математически невозможно:

```
print(10 / 0)  # Ошибка
```

Исправление: добавьте проверку:

```
if b != 0:     print(a / b)
```

UnboundLocalError: local variable 'x' referenced before assignmen

Вопрос: почему переменная не видна внутри функции?

Ответ: локальная переменная объявлена после использования или конфликтует с глобальной:

```
x = 10
def func():
    print(x)  # Ошибка, если ниже есть x = 5     x = 5
```

Исправление: используйте global x или переименуйте переменную.

ModuleNotFoundError: No module named 'module_name'

Вопрос: почему Python не находит модуль?

Ответ: модуль не установлен или имя написано с ошибкой:

```
import some_unknown_module  # Ошибка
```

Исправление: установите модуль (pip install module_name) или проверьте имя.

TypeError: 'NoneType' object is not iterable

Вопрос: почему нельзя итерировать None?

Ответ: функция вернула None, а вы пытаетесь её перебрать:

```
lst = None

for item in lst:  # Ошибка     print(item)
```

Исправление: проверьте, что переменная является списком/итерируемым объектом.

SyntaxError: EOL while scanning string literal

Вопрос: почему Python ругается на строку?

Ответ: вы забыли закрыть кавычки:

```
s = "Hello  # Ошибка
```

Исправление: добавьте закрывающие кавычки.

TypeError: 'int' object is not callable

Вопрос: почему число нельзя вызвать как функцию?

Ответ: вы переопределили имя функции числом:

```
len = 10

print(len("abc"))  # Ошибка: len теперь int
```

Исправление: не используйте имена встроенных функций для переменных.

TabError: inconsistent use of tabs and spaces in indentation

Вопрос: почему смешивать табы и пробелы плохо?

Ответ: Python требует единообразия в отступах:

```
if True:    print("A")  # 4 пробелаprint("B")  # Таб → Ошибка
```

Исправление: используйте только пробелы (рекомендуется 4).

TypeError: missing 1 required positional argument: 'self'

Вопрос: почему метод класса требует self?

Ответ: вы вызываете метод класса без экземпляра:

```
class MyClass:
        def method(self):
                print("Hello")
MyClass.method()  # Ошибка
```

Исправление: создайте объект:

```
obj = MyClass() obj.method()
```

RecursionError: maximum recursion depth exceeded

Вопрос: почему функция вызывает саму себя бесконечно?

Ответ: рекурсия не имеет условия выхода:

```
def infinite():
        infinite()  # Бесконечная рекурсия
```

Исправление: добавьте базовый случай:

```
def finite(n):
        if n == 0:
                return    finite(n - 1)
```

FileNotFoundError: [Errno 2] No such file or directory

Вопрос: почему Python не находит файл?

Ответ: файл не существует или путь указан неверно:

```
open("missing.txt")  # Ошибка
```

Исправление: проверьте путь и имя файла.

KeyError: 0 при работе со словарём

Вопрос: почему словарь не может найти ключ 0?

Ответ: ключа 0 нет в словаре:

```
d = {1: "one"}
print(d[0])  # Ошибка
```

Исправление: используйте d.get(0, "default").

TypeError: 'tuple' object does not support item assignment

Вопрос: почему кортеж нельзя изменить?

Ответ: кортежи (tuple) неизменяемы:

```
t = (1, 2)
 t[0] = 3  # Ошибка
```

Исправление: используйте список (list), если нужно изменять элементы.

Эти ошибки — часть процесса обучения. Чем чаще вы их встречаетесь, тем быстрее научитесь их исправлять. Если застряли — читайте сообщение интерпретатора, проверяйте типы и структуры данных. Удачи в изучении Python!