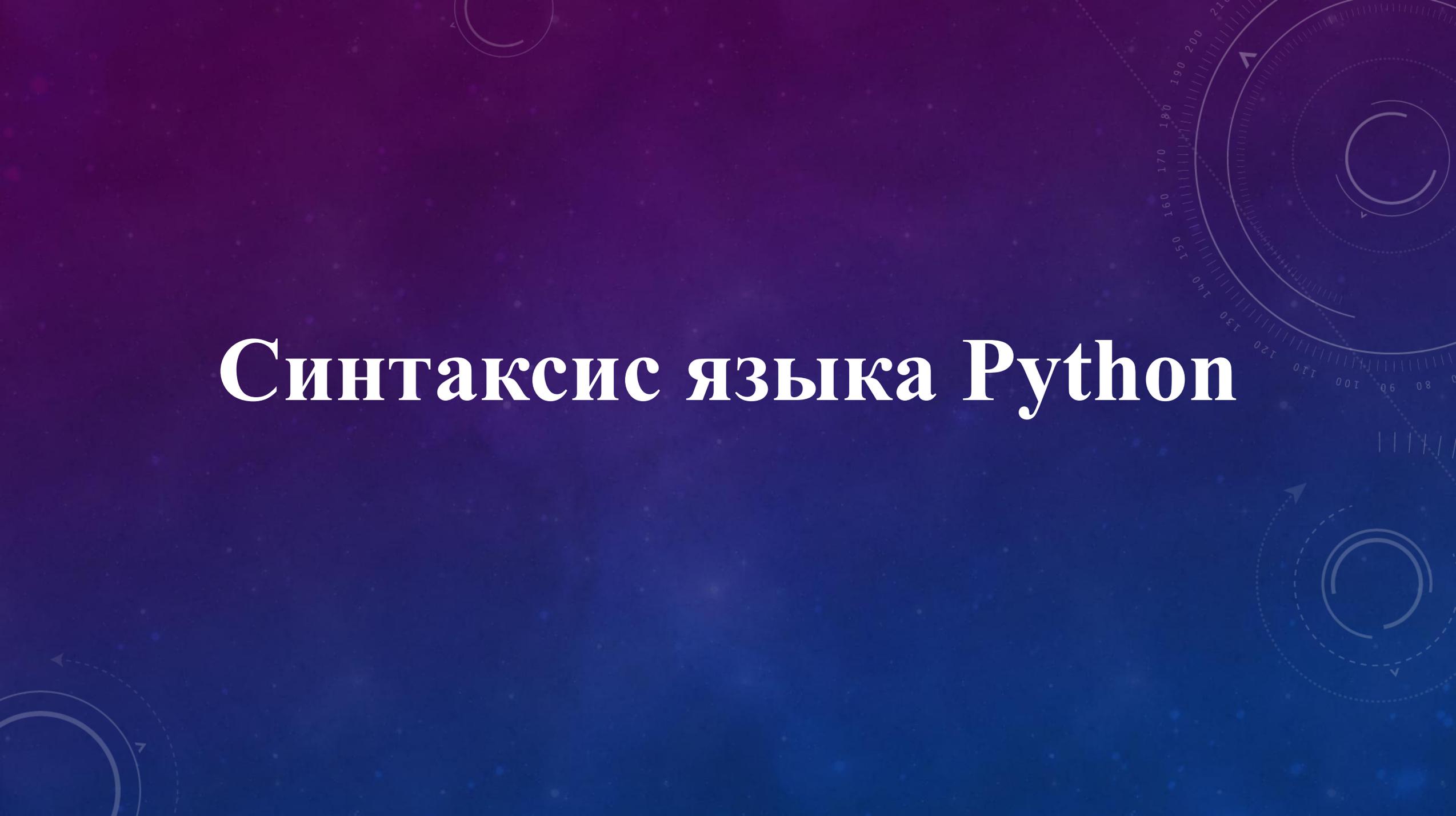


# Синтаксис языка Python

The background is a dark blue gradient with a field of small white stars. Overlaid on this are several faint, light blue technical diagrams. On the right side, there are two circular gauges or dials with numerical scales (100, 110, 120, 130, 140, 150, 160, 170, 180, 190, 200, 210) and arrows. At the bottom, there are dashed circular paths with arrows indicating direction. The overall aesthetic is clean, modern, and technical.

Python не содержит операторных скобок (`begin..end` в pascal или `{..}` в Си), вместо этого блоки выделяются отступами: пробелами или табуляцией, а вход в блок из операторов осуществляется двоеточием.

- ✓ Конец строки является концом инструкции (точка с запятой не требуется).
- ✓ Вложенные инструкции объединяются в блоки по величине отступов. Отступ может быть любым, главное, чтобы в пределах одного вложенного блока отступ был одинаков.
- ✓ Вложенные инструкции в Python записываются в соответствии с одним и тем же шаблоном, когда основная инструкция завершается двоеточием, вслед за которым располагается вложенный блок кода, обычно с отступом под строкой основной инструкции.

# Несколько специальных случаев

Иногда возможно записать несколько инструкций в одной строке, разделяя их точкой с запятой:

```
a = 3; b = 5; print(a, b)
```

Допустимо записывать одну инструкцию в нескольких строках. Достаточно ее заключить в пару круглых, квадратных или фигурных скобок:

```
if (a == 1 and b == 2 and  
    c == 3 and d == 4): # Не забываем про двоеточие  
    print('spam' * 3)
```

Тело составной инструкции может располагаться в той же строке, что и тело основной, если тело составной инструкции не содержит составных инструкций. Например:

```
if x > y: print(x)
```

# Комментирование в Python

Символ решетки (#) в Python обозначает начало комментария. Любые символы после решетки и до конца строки считаются комментариями и игнорируются интерпретатором. Например:

```
# Это комментарий  
print "Hello, Python" # И это комментарий
```

# Кавычки в Python

В Python можно использовать одинарные ( ' ), двойные ( " ) и тройные ( """ или """) кавычки чтобы обозначить строчный тип данных, при этом начинаться и заканчиваться строка должна одинаковыми кавычками. Строка занимающая несколько строк кода должна быть обрамлена тройными кавычками. Например:

```
name = 'wasd'  
description = "Some text"  
biography = """ Some long text for few  
lines of code """
```

# Многострочные выражения

Выражения в Python, как правило, заканчиваются новой строкой. Однако, существует специальный символ переноса строки (`\`), показывающий, что с окончанием строки не заканчивается код. Например:

```
total = item1 + \  
        item2 + \  
        item3
```

# Идентификаторы в Python

Идентификаторы в Python это имена используемые для обозначения переменной, функции, класса, модуля или другого объекта.

Идентификатор должен начинаться с буквы (от а до Z) или со знака подчеркивания ( ), после которых может идти произвольное количество букв, знаков подчеркивания и чисел (от 0 до 9).

В Python недопустимо использование знаков препинания или специальных символов, таких как @, \$ или % в качестве идентификаторов.

Кроме того, Python чувствителен к регистру, то есть cat и Cat это два разных имени.

В Python существует следующая договоренность для названия идентификаторов:

- ✓ Имена классов начинаются с большой буквы, все остальные идентификаторы с маленькой.
- ✓ Использование знака подчеркивания в качестве первого символа идентификатора означает, что данный идентификатор является частным (закрытым от использования вне класса).
- ✓ Если идентификатор начинается и заканчивается двумя знаками подчеркивания (например, `__init__`) это означает, что он является специальным именем, определенным внутри языка.