



*School of
Computer
Science*

МОДЕЛЬ ПАМЯТИ В PYTHON. УПРАВЛЯЮЩИЕ КОНСТРУКЦИИ IF ПРОГРАММИРОВАНИЕ НА PYTHON

Лекции для IT-школы



ВОПРОСЫ ПО ПРОШЛОМУ ЗАНЯТИЮ

1. Как узнать список имен, определенных в текущей области видимости Python?
2. Как узнать список встроенных имен?
3. Как получить список ключевых слов?
4. В чем разница между ключевыми словами и встроенными именами?
5. Как удалить имя в локальном контексте?



ВОПРОСЫ ПО ПРОШЛОМУ ЗАНЯТИЮ

6. Какие кавычки используются для строк в Python? С какой целью?
7. Как размножить содержимое строки N раз?
8. С какой позиции индексируются строки?
9. Как обратиться к символу в строке с заданной позицией?
10. Как получить срез (часть) строки с заданной начальной и конечной позициями?



ВОПРОСЫ ПО ПРОШЛОМУ ЗАНЯТИЮ

Что такое исключение в программировании:

1. Исключительно хороший код, которому можно только позавидовать
2. Механизм обработки ошибочных ситуаций
3. Ошибка, при возникновении которой программа перестает работать



ВОПРОСЫ ПО ПРОШЛОМУ ЗАНЯТИЮ

– Какие из этих слов применяются при обработке исключений в Python?

- 1) disaster
- 2) try
- 3) problem
- 4) catch
- 5) except
- 6) call_police





ВОПРОСЫ ПО ПРОШЛОМУ ЗАНЯТИЮ

Как называется символьный тип в Python:

1. `char`
2. `varchar`
3. `str`
4. никак не называется



ВОПРОСЫ ПО ПРОШЛЫМ ЗАНЯТИЯМ

Укажите выражения, значения которых равны True:

1. `'239' < '30' and 239 < 30`
2. `'239' < '30' and 239 > 30`
3. `'239' > '30' and 239 < 30`
4. `'239' > '30' and 239 > 30`



МОДЕЛЬ ПАМЯТИ В PYTHON

Пространство имён,
namespace:

Пространство объектов,
heap:

Имя	Ссылка/Адрес	Значение	Тип
name	1237856	→ 'Вася'	str
amount	1237868	→ 5678.78	float
b_25	1243464	→ 4346	int
is_cold	4573735	→ True	bool
...

id(<переменная>) – получение адреса переменной в пространстве объектов



ОПЕРАТОРЫ ПРИСВАИВАНИЯ

1. Обычная ссылка на значение: `var_1 = 1`
2. Присваивание одного значения нескольким переменным: `var_1 = var_2 = var_3 = 8`
3. Множественное присваивание:
`var_1, var_2, var_3 = 5, 3.15, "Вася"`
4. Присваивание, совмещенное с арифметической операцией:
`+=, -=, /=, //=, %=, *=`



КОНСТАНТЫ

В Python нет ключевых слов для описания констант

По общепринятому соглашению константы в Python определяются в верхнем регистре

Пример:

```
>>> HOURS_IN_DAY = 24 # количество часов в сутках
>>>
>>> MINUTES_IN_HOUR = 60 # количество минут в часе
>>>
>>> SECONDS_IN_MINUTE = 60 # количество секунд в минуте
>>>
>>> SECONDS_IN_HOUR = 3600 # количество секунд в часе
```



НЕИЗМЕНЯЕМОСТЬ ПРОСТЫХ ТИПОВ ДАННЫХ

- Типы данных в Python бывают изменяемые (`mutable`) и неизменяемые (`immutable`)
- Простые типы `int`, `float`, `bool`, `str` относятся к НЕизменяемым типам данных
- Значения неизменяемых типов данных не изменяются по ссылке
- Присваивая новое значение переменной неизменяемого типа мы получаем ссылку на НОВЫЙ объект этого типа



УСЛОВНЫЙ ОПЕРАТОР

Отступы
обязательны!

```
if <логическое выражение 1>:  
    ← КОД, выполняемый при True  
    для 1-го логического  
    выражения...  
[elif <логическое выражение 2>:  
    ← КОД, выполняемый при True  
    для 2-го логического  
    выражения...  
...]  
[else:  
    ← КОД, выполняемый, если все  
    условия выше дают False  
]
```



ВАРИАНТЫ ВЕТВЛЕНИЙ

Выражение	Описание
if <условие>: <блок>	Если <условие> истинно, то <блок> выполняется, иначе – пропускается
if <условие>: <блок 1> else: <блок 2>	Условная конструкция с условием else. Если <условие> истинно, выполняется <блок1>, если ложно – <блок2>
if <условие 1>: <блок 1> elif <условие 2>: <блок 2> ... elif <условие N>: <блок N> else: <блок N+1>	Условная конструкция с дополнительными условиями elif и необязательным else в конце. Будет исполнен <u>единственный</u> блок if / elif после <u>первого</u> условия, которое окажется истинным. Если же нет условий принимающих значение True, то будет исполнен блок после заключительного else



ОПЕРАТОР ВЫБОРА

СМ. `INLINE_IF.PY`

блок-True **if** (условие) **else** блок-False

аналог оператора `? :` в C++ и Java

Примеры:

```
max_var = var1 if (var1 > var2) else var2
```

```
print("Even" if (var_int % 2 == 0) else "Odd")
```



УСЛОВНЫЙ ОПЕРАТОР С IF. ПРИМЕР

```
pwd = input("Введите ваш пароль: ")
```

```
if pwd == "secret":  
    print("Доступ предоставлен")  
    <действия после предоставления  
    доступа>
```



УСЛОВНЫЙ ОПЕРАТОР С IF-ELSE.

ПРИМЕР

```
pwd = input("Введите ваш пароль: ")

if pwd == "secret":
    print("Доступ предоставлен")
    <действия после предоставления
    доступа>
else:
    print("Доступ запрещен")
    <действия после запрещения
    доступа>
```




УСЛОВНЫЙ ОПЕРАТОР С ELIF.

ПРИМЕР

```
weight = int(input('Введите ваш вес:'))

if weight > 100:
    print('Вам пора в спортзал!')
elif weight < 50:
    print('Вам надо лучше питаться')
else:
    print('У вас нормальный вес')
```



ПРАКТИЧЕСКОЕ ЗАДАНИЕ №1. IF-ELSE

См. [dice_guess_template.py](#)

Написать программу, которая запрашивает целое число от 1 до 6.



Если введено число больше 6 или меньше 1, то программа должна вывести текст «Вы ввели неверное число».

Если число совпало с тем, что на кубике, то выводится текст «Вы угадали!», иначе выводится текст «Вы не угадали. На кубике выпало X»



СОВЕТЫ ПО ИСПОЛЬЗОВАНИЮ IF-ELIF-ELSE

- В сложных ветвлениях используйте ветку `else` даже если она вам сейчас не требуется:
 - Чтобы определить когда «что-то пошло не так»
 - Пример в [mood_detector.py](#)
- `if-elif-elif` не то же самое, что последовательность операторов `if-if-if`:
 - Среди **`if-elif-elif...`** выполнится **только один блок**
 - Пример в [structured_if.py](#)
- Вложенный `if` можно убрать в пользу внешнего `if` с объединением условий по `and`:
 - Пример в [structured_if_with_and.py](#)



Переменные `grade1` и `grade2` представляют баллы за 2 курса. Переменная `num_passed` равна нулю. После каких фрагментов кода `num_passed` будет содержать количество пройденных экзаменов, по которым достигнут или превышен проходной балл (50 или более)?

1)

```
if grade1 >= 50:
    num_passed = num_passed + 1
if grade2 >= 50:
    num_passed = num_passed + 1
```

2)

```
if grade1 >= 50:
    num_passed = num_passed + 1
elif grade2 >= 50:
    num_passed = num_passed + 1
```

3)

```
if grade1 >= 50 and grade2 >= 50:
    num_passed = 2
if grade1 >= 50:
    num_passed = 1
if grade2 >= 50:
    num_passed = 1
```

4)

```
if grade1 >= 50 and grade2 >= 50:
    num_passed = 2
elif grade1 >= 50:
    num_passed = 1
elif grade2 >= 50:
    num_passed = 1
```



Рассмотрите этот блок кода:

```
if temperature > 28:
    if money >= 30:
        print("Я покупаю мороженку")
    else:
        print("Где можно найти тенёк?")
```

1)

```
if temperature > 28 and money >= 30:
    print("Я покупаю мороженку")
elif temperature > 28 and money < 30:
    print("Где можно найти тенёк?")
```

2)

```
if temperature > 28 and money >= 30:
    print("Я покупаю мороженку")
elif temperature > 28:
    print("Где можно найти тенёк?")
```

3)

```
if temperature > 28 and money >= 30:
    print("Я покупаю мороженку")
else:
    print("Где можно найти тенёк?")
```

Какие из приведенных кусков кода 1,2,3 ему идентичны?

Какое из решений выглядит наиболее элегантно?

Какой(ие) вариант(ы) отбросили и почему?



ПРАКТИЧЕСКОЕ ЗАДАНИЕ №2. РАСЧЕТ ШТРАФА

- На участке дороги установлено ограничение скорости в 40 км/ч
- При скорости 140 км/ч и выше водитель лишается прав
- Если скорость в 40 км/ч превышена то за каждые лишние 20 км/ч к штрафу доначисляется 500 штрафных рублей и определяется общая сумма штрафа
- Программа должна принимать на вход скорость автомобиля в км/ч и выдавать сумму штрафа, либо сообщения «Штрафа нет» / «Лишение прав!»



СПАСИБО ЗА ВНИМАНИЕ !
ВОПРОСЫ ?



*School of
Computer
Science*