

# Методические указания

## Урок 2.2 Типы данных

### Задачи урока:

- Познакомить с основными типами данных в python;
- Разобрать работу с типами на задачах.

## 0. Подготовка к уроку

До урока преподавателю необходимо:

- 1) Прочитать методичку до конца.

## 1. Основные типы данных в языке

**Учитель:** Сегодня мы с продолжим разбирать типы данных языка Python. Для начала повторим типы которые мы знаем с прошлого занятия. *Какие типы вы помните?(спрашивает у учеников)*

1. integer - целое число
2. float - число с плавающей точкой
3. decimal - число с фиксированной точностью
4. string - строка
5. boolean - булев тип. Имеет всего два варианта: правда(True) и ложь(False)
6. list - список
7. tuple - кортеж
8. set - множество
9. frozenset - неизменяемое множество
10. dict - словарь

На текущем занятии мы немного поработаем с типами boolean, string и list. У них мы затронем основные операции проводимые с данными типами. Более подробно мы будем разбирать каждый тип на следующих занятиях, так как почти каждый тип имеет свои возможности (методы), которые облегчают нам написание кода.

**Учитель:** Давайте сегодня напишем простенькую игру угадай слово, а потом порешаем пару задачек.

Для начала давайте подумаем и разобьем на более мелкие задачи написание нашей игры. *Как вы думаете на какие этапы мы можем разбить разработку программы. (спрашивает учеников)*

1. Требуется загадать какое то слово
2. Указать количество попыток
3. Вывести слово заменив буквы какими либо символами
4. Попросить пользователя ввести букву
5. Сравнить букву с буквами в загаданном слове
6. Если указанной буквы нет, то отнять попытку, если пользователь угадал - открыть букву
7. Повторить все пункты с 4 по 6 до победы или поражения
8. Реализовать проверку на победу и поражения

**Учитель:** Ну что ж давайте приступим. Для начала выполним первые два пункта

```
print('Попробуй угадать слово по буквам!')
secret_word = 'машина'
attempts = 5
```

Отлично начала нашей игры положено. Стоит обратить внимание, что пользователь может начать вводить буквы не в том регистре и тогда он не сможет угадать букву. Решение данной проблемы мы рассмотрим на занятии, когда будет более подробно разбирать возможности строк.

Теперь нам необходимо выполнить пункт 3, так как пользователь должен знать как минимум сколько букв в нашем слове, а в дальнейшем и видел, какие буквы он уже отгадал. Для реализации данной затеи нам потребуется цикл. В данной программе, мы воспользуемся вторым циклом - for. Он отличается от цикла while тем, что повторяется ограниченное количество раз и удобен для вывода символов/элементов строки/списка. Детально мы его разберем на уроке посвященном циклам.

Так как слово у нас может быть разной длины, мы укажем циклу повторяться столько раз, сколько символом в нашем слове.

```
print('Попробуй угадать слово по буквам!')
secret_word = 'машина'
```

```
attempts = 5
for i in range(len(secret_word)):
    print("*")
```

Отлично, но есть определенная проблема. Символы у нас выводятся, каждая на новой строке. Все из-за того, что когда мы используем команду `print()`, независимо написано что в скобках или нет, у нас проставляется символ переноса строки `\n`. Решить данную проблему нам поможет возможность изменять конечный символ в `print()` - аргумент `end`. Мы укажем, что вместо переноса строки у нас будет например пробел

```
print('Попробуй угадать слово по буквам!')
secret_word = 'машина'
attempts = 5
for i in range(len(secret_word)):
    print("*", end='')
```

Давайте реализуем ввод и проверку буквы пользователя. Для хранения букв пользователя, так как буква будет не одна, мы будем использовать список (list). Список позволяет хранить множество значений любого типа. А для проверки совпадения буквы мы воспользуемся оператором `in`, который проверяет есть ли элемент в списке/строке и `not in`, который делает противоположное. Добавлять в список буквы мы будем с помощью команды `append`. Она добавляет элемент в конец списка.

```
print("Попробуй угадать слово по буквам!")
secret_word = "машина"
attempts = 5
user_chars = []
for i in range(len(secret_word)):
    print("*", end='')
print()
user_char = input('Введите букву ')
if user_char not in user_chars:
    user_chars.append(user_char)
for char in secret_word:
    if char in user_chars:
        print(char, end='')
    else:
        print("*", end='')
```

Команда else позволяет выполнить какой либо код если условие(if) оказалось неверно и переводится как иначе (if - если, else - иначе)  
Осталось добавить цикл, выводить и отнимать попытки, а также реализовать победу и проигрыш.

```
print("Попробуй угадать слово по буквам!")
secret_word = "машина"
attempts = 5
user_chars = []
for i in range(len(secret_word)):
    print("*", end='')
print()
while True:
    is_win = True
    print('Попыток: ' + str(attempts))

    user_char = input('Введите букву ')
    if user_char not in user_chars:
        user_chars.append(user_char)
    for char in secret_word:
        if char in user_chars:
            print(char, end='')
        else:
            print('*', end='')
            is_win = False
    print()
    if user_char not in secret_word:
        attempts -= 1
    if attempts == 0:
        print('Вы проиграли. Загаданное слово: ' + secret_word)
        break
    if is_win == True:
        print('Вы победили')
        break
```

## 2. Решение задач

Задача 1.

### **Дополнительно**

Если на уроке остается время, то ученикам можно предложить начать прорешивать домашнее задание.

### **Домашняя работа**

В конце урока можно подвести итоги и дать домашнюю работу:

1. Написать программу, запрашивающую на ввод год рождения, которая рассчитывает возраст пользователя