

ЛЕКЦИЯ 9

- Условные конструкции
- Циклы
- Функции
- Обработка исключений

```
if <условие>:  
    <действие>  
elif <другое_условие>:  
    <действие>  
else:  
    <действие>
```

УСЛОВНЫЕ КОНСТРУКЦИИ: ПРИМЕР

```
In [1]: age : int = 16
        if age >= 18:
            print('Вы совершеннолетний')
        elif age < 18:
            print('Вы несовершеннолетний')
```

Вы несовершеннолетний

```
In [2]: age : int = 22
        if age >= 18:
            print('Вы совершеннолетний')
        elif age < 18:
            print('Вы несовершеннолетний')
```

Вы совершеннолетний

Цикл *while*

```
while <условие>:  
    <действие>
```

Цикл *for*

```
for<итератор> in <итерируемый_объект>:  
    <действие>
```

ЦИКЛЫ: ПРИМЕР

```
In [13]: # вывести каждый элемент списка
i : int=0
l=[1,2,3,4]
while i < len(l):
    print(l[i],end=' ')
    i+=1
```

1 2 3 4

```
In [12]: # вывести каждый элемент списка
l=[1,2,3,4]
for i in range(len(l)):
    print(l[i],end=' ')
```

1 2 3 4

```
In [15]: # вывести каждый элемент списка
l=[1,2,3,4]
for i in l:
    print(i,end=' ')
```

1 2 3 4

ЦИКЛЫ: RANGE ОБЪЕКТ

```
In [17]: range_object=range(10)
print(range_object)
print(list(range_object))

range(0, 10)
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

ЦИКЛЫ: BREAK И CONTINUE

```
In [19]: i : int = 0
while True:
    if i == 10:
        break
    if i == 8:
        print('->', i, end=' ')
        i += 1
        continue
    print(i, end=' ')
    i += 1
```

0 1 2 3 4 5 6 7 -> 8 9

ФУНКЦИИ: СИНТАКСИС

```
In [20]: # описание функции
def my_foo(param1):
    print(f'Это моя первая функция с параметром {param1}')
# вызов функции
my_foo('параметр')
```

Это моя первая функция с параметром параметр

ФУНКЦИИ: RETURN

По умолчанию возвращается None

```
In [20]: # описание функции
def my_foo(param1):
    print(f'Это моя первая функция с параметром {param1}')
    return
# вызов функции
my_foo('параметр')
```

Это моя первая функция с параметром параметр

Можно вернуть любое значение, например, строку.

```
In [23]: # описание функции
def my_foo(param1):
    return f'Это моя первая функция с параметром {param1}'
# вызов функции
print(my_foo('параметр'))
```

Это моя первая функция с параметром параметр

ФУНКЦИИ: НЕОБЯЗАТЕЛЬНЫЕ ПАРАМЕТРЫ

```
In [24]: # описание функции
def my_foo(param1='необязательный параметр'):
    return f'Это моя первая функция с параметром {param1}'
# вызов функции
print(my_foo())
print(my_foo('параметр'))
```

Это моя первая функция с параметром необязательный параметр
Это моя первая функция с параметром параметр

ФУНКЦИИ: *ARGS & **KWARGS

```
In [25]: def args_and_kwargs(*args,**kwargs):  
          print('args',end=' ')  
          for i in args:  
              print(i,end=' ')  
          print('\n','kwargs',end=' ')  
          for i in kwargs.items():  
              print(i,end=' ')  
args_and_kwargs('arg1','arg2','arg3',arg1=1,arg2=2,arg3=3)
```

```
args arg1 arg2 arg3  
kwargs ('arg1', 1) ('arg2', 2) ('arg3', 3)
```

ФУНКЦИИ: LAMBDA ФУНКЦИИ

```
In [28]: lambda_function=lambda x : x**2  
print(lambda_function(2))
```

4

ИСКЛЮЧЕНИЯ: БЛОК TRY EXCEPT ELSE FINALLY

```
In [33]: try:
          print('Пытаюсь что то сделать')
          value=int(input('Введите число '))
        except:
          value=10
          print('Выполнюсь только если возникло исключение')
        else:
          print('Выполнюсь только если не возникло исключения')
        finally:
          print('Выполнюсь в любом случае')
```

Пытаюсь что то сделать
Введите число
Выполнюсь только если возникло исключение
Выполнюсь в любом случае

```
In [34]: try:
          print('Пытаюсь что то сделать')
          value=int(input('Введите число '))
        except:
          value=10
          print('Выполнюсь только если возникло исключение')
        else:
          print('Выполнюсь только если не возникло исключения')
        finally:
          print('Выполнюсь в любом случае')
```

Пытаюсь что то сделать
Введите число 10
Выполнюсь только если не возникло исключения
Выполнюсь в любом случае
