

Python основы

Краткий конспект на основе прохождения курсов

Основные типы

type(arg) – функция получения типа аргумента *arg*

int, float, str – основные типы

int – 1, 20, 34,

float – 1.2, 3.0, 156.823244...

str – 'hello'

конкатенация

1 + '1' – неправильно

1 + int('1') – правильно

'hello' + 1 – неправильно

'hello' + str(1) – правильно

но

'abc' * 5 = 'abcabcabcabcabc', потому что * это перегруженный оператор

len()

len(*arg*) – функция получения количества символов аргумента *arg*. В Python используется Unicode.

Ввод данных от пользователя

`a = input('Введите ваше имя:')` - получение ввода от пользователя и присвоение значения в переменную *a*

Вывод данных

`print('Hello')` - выведет *Hello*

`a = 'Hello Pete!'`

`print(a)` - выведет *Hello Pete!*

`name = 'Pete'`

`age = 17`

`a = 'Hello {}. You are {} age old.'.format(name, age)` – выведет *Hello Pete. You are 17 age old.*

`a = 'Hello {0}. You are {1} age old.'.format(name, age)` – тоже выведет *Hello Pete. You are 17 age old.*

`a = 'Hello {1}. You are {0} age old.'.format(name, age)` – тоже выведет *Hello 17. You are Pete age old.*

`a = f'Hello {name}. You are {age} age old.'` – тоже выведет *Hello Pete. You are 17 age old.* (Python 3.6+)

Условие

```
if a == b:
    #...do something1
    #...do something2
    #...do something3
elif a > b:
    #...do something else 1
else:
    #...do some other1
    #...do some other2
```

блоки кода обозначаются отступами, никаких скобок как в Си-подобных языках.

Если надо оставить (например на время) тело условия пустым, используется **pass**

```
if a == b:
    pass
else:
    #...do some other1
```

Цикл while

```
i = 0  
while i < 5:  
    #...do something1  
    i += 1
```

Python Zen

Явное лучше неявного.



