

Строки

План урока

- Строки

СТРОКИ

это упорядоченные коллекции символов в кодировке Unicode

они являются неизменяемым типом данных и подобны кортежам, но имеют методы для работы со строками, такие как:

- поиск подстроки
- изменение регистра символов

Работа со строками

ooo

код

```
s = "Abcdefgh"  
print(s, type(s))  
  
print(s[0], s[1], s[-1])  
  
print(len(s))  
  
print(s[:6:2])
```

Строка s из
первых восьми
букв английского
алфавита

ooo

вывод

```
>>> Abcdefgh <class 'str'>  
>>> A b h  
>>> 8  
>>> Ace
```

Работа со строками

ооо | код

```
s = "Abcdefgh"  
print(s, type(s)) ←  
print(s[0], s[1], s[-1])  
  
print(len(s))  
  
print(s[:6:2])
```

Выводим
строку
s и её тип

ооо | вывод

```
>>> Abcdefgh <class 'str'> ←  
>>> A b h  
>>> 8  
>>> Ace
```

Работа со строками

ooo | код

```
s = "Abcdefgh"  
  
print(s, type(s))  
  
print(s[0], s[1], s[-1]) ←  
  
print(len(s))  
  
print(s[:6:2])
```

Выводим
символы строки,
используя
различные
индексы

ooo | вывод

```
>>> Abcdefgh <class 'str'>  
>>> A b h ←  
>>> 8  
>>> Ace
```

Работа со строками

ооо | код

```
s = "Abcdefgh"  
  
print(s, type(s))  
  
print(s[0], s[1], s[-1])  
  
print(len(s)) ←  
  
print(s[:6:2])
```

Выводим
длину
строки
используя
метод len

ооо | вывод

```
>>> Abcdefgh <class 'str'>  
>>> A b h  
>>> 8 ←  
>>> Ace
```

Работа со строками

ооо | код

```
s = "Abcdefgh"  
  
print(s, type(s))  
  
print(s[0], s[1], s[-1])  
  
print(len(s))  
  
print(s[:6:2]) ←
```

Выводим срез

ооо | вывод

```
>>> Abcdefgh <class 'str'>  
>>> A b h  
>>> 8  
>>> Ace ←
```

Неизменяемый тип

ооо | код

```
s = "Abcdefgh"
```

```
s[1] = 'e' ←
```

ооо | вывод

```
>>> TypeError: 'str' object  
does not support item assignment ←
```

При попытке
модификации
по индексу
возникает ошибка

Строка

ooo | код

```
s = "Abcdefgh"  
  
for ch in s:  
    print(ch, end=' ')  
print()
```

ooo | вывод

```
>>> A b c d e f g h
```

Перебираем
символы строки
с помощью цикла for

Сложение строк

ооо	код
<pre>s1 = "cool" s2 = "feature" s3 = s1 + s2 ← print(s3)</pre>	

ооо	вывод
<pre>>>> coolfeature</pre>	

Объединяем две
строки, используя
конкатенацию

Сложение строк

ооо | код

```
s1 = "cool"  
s2 = "feature"  
s3 = s1 + s2  
  
print(s3)
```

ооо | вывод

```
>>> coolfeature <
```

Объединенная
строка без
пробела

Сложение строк

ооо

код

```
s1 = "cool"  
s2 = "feature"  
s3 = s1 + " " + s2  
  
print(s3)
```

ооо

вывод

```
>>> cool feature ←
```

Объединенная
строка с
пробелом

МЕТОД JOIN

применяется к строке-разделителю и
объединяет список строк, проставляя между
ними указанный разделитель

Метод join

ooo | код

```
s1 = "cool"  
s2 = "feature"  
  
s3 = ' '.join([s1, s2]) ←  
  
print(s3)
```

ooo | вывод

```
>>> cool feature
```

Применяем метод join к строке
с одним пробелом и принимаем
в качестве аргумента список
из строк S1 и S2

Метод join

ooo | код

```
words = [  
    "another",  
    "brick",  
    "in",  
    "the",  
    "wall"  
]  
  
phrase = ' '.join(words)  
print(phrase)
```

ooo | вывод

```
>>> another brick in the wall
```

Метод join

ooo | код

```
phone_number_parts = [  
    "123",  
    "45",  
    "63"  
]  
  
phone_number =  
    '-'.join(phone_number_parts)  
print(phone_number)
```

ooo | вывод

```
>>> 123-45-63
```

Другие методы строк

ооо

код

```
print("AbcDEfgh".lower())
```

```
print("AbcDEfgh".upper())
```

```
print("AbcDEfgh".capitalize())
```

ооо

вывод

```
>>> abcdefgh
```

```
>>> ABCDEFGH
```

```
>>> Abcdefgh
```

Метод lower

ооо | Код

```
print("MisterSmith@Matrix.com".lower())
```

ооо | Вывод

```
>>> mistersmith@matrix.com
```

Метод count

ооо | код

```
s = "man dog cat sea fly dog mouse cat dog ice"  
print(s.count("dog"))
```

ооо | вывод

```
>>> 3
```

Ищем число вхождения строки
«dog» в исходную строку

python.org

F-СТРОКИ

позволяют вставлять выражения или переменные любого типа, которые могут быть преобразованы в строки

ooo | код

```
print(f"{2 * 37}")
```

ooo | вывод

```
>>> 74
```

Пример

ooo | код

```
name = "Joe"  
age = 23  
  
s = f"{name} is {age} years old"  
  
print(s)
```

ooo | вывод

```
>>> Joe is 23 years old
```

В качестве аргументов
передаются {name} и {age}

Пример

ooo | код

```
x = 1  
y = 3
```

```
print(f"x / y = {(x / y):.3f}")
```

ooo | вывод

```
>>> x / y = 0.333
```

Unicode

ооо

код

```
print("Привет")
print("你好世界")
```

ооо

вывод

```
>>> Привет
>>> 你好世界
```