

Дзен Python на русском

Разработчики языка Python придерживаются определённой философии программирования, называемой «The Zen of Python» («Дзен Питона», или «Дзен Пайтона»). Её текст выдаётся интерпретатором Python по команде `import this` (работает один раз за сессию).

В целом она подходит к программированию на любом языке.

Текст философии

- Красивое лучше, чем уродливое.
- Явное лучше, чем неявное.
- Простое лучше, чем сложное.
- Сложное лучше, чем запутанное.
- Плоское лучше, чем вложенное.
- Разреженное лучше, чем плотное.
- Читаемость имеет значение.
- Особые случаи не настолько особые, чтобы нарушать правила.
- При этом практичность важнее безупречности.
- Ошибки никогда не должны замалчиваться.
- Если они не замалчиваются явно.
- Встретив двусмысленность, отбрось искушение угадать.
- Должен существовать один и, желательно, только один очевидный способ сделать это.
- Хотя он поначалу может быть и не очевиден, если вы не голландец [^1].
- Сейчас лучше, чем никогда.
- Хотя никогда зачастую лучше, чем прямо сейчас.
- Если реализацию сложно объяснить – идея плоха.
- Если реализацию легко объяснить – идея, возможно, хороша.
- Пространства имён – отличная штука! Будем делать их больше!

Автор этой философии – Тим Петерс. Оригинал на английском:

```
λ python
```

```
Python 3.6.2 (v3.6.2:5fd33b5, Jul 8 2017, 04:14:34) [MSC  
v.1900 32 bit (Intel)] on win32
```

```
Type "help", "copyright", "credits" or "license" for more  
information.
```

```
>>> import this
```

```
The Zen of Python, by Tim Peters
```

Beautiful is better than ugly.

Explicit is better than implicit.

Simple is better than complex.

Complex is better than complicated.

Flat is better than nested.

Sparse is better than dense.

Readability counts.

Special cases aren't special enough to break the rules.

Although practicality beats purity.

Errors should never pass silently.

Unless explicitly silenced.

In the face of ambiguity, refuse the temptation to guess.

There should be one-- and preferably only one --obvious way to
do it.

Although that way may not be obvious at first unless you're
Dutch.

Now is better than never.

Although never is often better than **right** now.

If the implementation is hard to explain, it's a bad idea.

If the implementation is easy to explain, it may be a good
idea.

Namespaces are one honking great idea -- let's do more of
those!