

Perl. Описание



- **Perl** — Practical Extraction and Reporting Language — высокоуровневый язык общего назначения.
- Первоначально разработал Larry Wall в 1987 в США для обработки отчетов
- Предоставляет широкие возможности для обработки текстов
- Ориентирован на решение практических задач, нежели красоту/изысканность синтаксических конструкций
- Процедурный, с поддержкой объектно-ориентированной и функциональной парадигм программирования
- Автоматическое управление памятью. Сборщик мусора
- Интерпретируемый. В Perl 6 появится возможность компилирования в байт-код
- Специальные синтаксические конструкции для работы с регулярными выражениями и потоками ввода/вывода
- Девизы:
 - «There's more than one way to do it» (TMTOWTDI) — «Есть больше одного способа сделать это»
 - «Easy things should be easy and hard things should be possible» — «Простые вещи должны быть простыми, а сложные вещи — возможными»

Perl. Синтаксис



- Унаследовал общую структуру синтаксиса от языка Си
- Возможно создавать сложные структуры данных, как массивы/хеши ссылок на более простые типы
- Возможность создания модулей
- Примеры.

```
sub GCD {  
    my ($a, $b) = @_;  
    my $c = $b;  
    while ($a > 0)  
    {  
        $c = $a;  
        $a = $b % $a;  
        $b = $c;  
    }  
    return $c;  
}  
print GCD(42, 56);
```

```
$number = 5;  
$string = "String";  
$multilined_string = <<EOF;  
Multilined string,  
terminating with the word "EOF".  
EOF
```

```
@array = (1, 2, "three");  
print $array[1]; # "2"
```

```
%hash = ('one' => 1, 'two' => 2);  
print $hash{'one'}; # "1"
```

```
$ref = \$number;  
print $ref; # SCALAR(0x14ef640)  
print $$ref; # 5
```

```
$circle={  
    center=>{x=>210, y=>297},  
    radius=>53,  
};
```

Perl. Особенности



- Встроенная в синтаксис поддержка регулярных выражений
 - `$x =~ m/abc/; # Истина, если $x содержит «abc»`
 - `$x =~ s/abc/def/; # Заменит «abc» на «def» в $x`
 - `$x =~ m/(\d+)\.(\d+)\.(\d+)\.(\d+)\./ # Найдёт первое вхождение IP-адреса, и сохранит его части в переменных $1, $2, $3, $4`
 - `@a = ($x =~ m/(\d+)/g); # Сохранит в массив @a все найденные в строке числа`
 - `$x =~ s/(\d+) б /$1 Б/g # Заменит «X б» на «X Б»`
- Регулярные выражения Perl удобны и включены во многие другие языки программирования
- Пример. Печать простых чисел:

```
perl -wle '(1 x $_) !~ /^(11+)\1+$/ && print while ++ $_'
```
- Встроенные в синтаксис возможности для работы с потоками ввода/вывода

```
while ( <> ){ # построчное чтение из выбранного потока в переменную $_
  if ( /print/ ){ # если строка в $_ содержит «print» (регулярное выражение)
    print;      # печатаем содержимое переменной $_
  }
}
```

Perl. Применение



- Применение Perl:
 - Разработка инструментов системного администрирования
 - Обработка почты
 - CGI-сценарии
 - Обработка текстов
 - Работа с базами данных
 - Разработка средств тестирования
- Perl используют:
 - Amazon — онлайн сервисы
 - The BBC — обработка данных
 - Ebay — онлайн сервисы
 - Yahoo! — онлайн сервисы
 - NASA — скриптование в различных системах расчета

Python. Описание



- **Python** — высокоуровневый язык общего назначения
- Первоначально разработал Guido van Rossum в 1990 в нидерландском CWI
- Акцент на производительность разработчика и читаемость кода
- Объектно-ориентированный, с поддержкой процедурной и функциональной парадигм
- Минималистичный синтаксис
- Динамическая типизация
- Автоматическое управление памятью. Сборщик мусора
- Интерпретируемый и/или компилируемый в байт-код
- Полная интроспекция
- Механизм обработки исключений
- Интерактивный режим выполнения

Python. Синтаксис



- Отступы определяют разделение программы на блоки
- Возможность написания классов
- Поддержка разделения на модули
- Переменные — имена ссылок на значения
- Коллекции (контейнеры):
 - Списки
 - Кортежи
 - Словари
 - Множества
- Передаются по ссылке
- Примеры:

```
def GCD(a, b):  
    while b > 0:  
        a,b = b,a%b  
    return a  
  
print GCD(42, 56)
```

```
number = 5  
complex_number = 1.5 + 0.5j  
print complex_number.imag # "0.5"  
big_number = 2**1000  
print len(str(big_number)) # "302"
```

```
string = "String"  
multiline_string = """  
Multilined string,  
Yep.  
"""  
unicode_string = u"Уникод"
```

```
list_var = [1, 2, 'three']  
tuple_var = (1, 2, 'three')  
print list_var[1] # "2"
```

```
dictionary = {'one': 1, 'two': 2}  
print dictionary['one'] # "1"  
set_var = set([1, 2, 'five'])  
print 'five' in set_var # "True"
```

Python. Особенности



- Полная интроспекция
- Поддержка ООП
 - Абстракция — утиная типизация
 - Инкапсуляция — ограниченная ввиду интроспекции
 - Множественное наследование
 - Полиморфизм — все функции виртуальные
- Механизм обработки исключений
- Концепция итераторов
 - Единая схема итерирования
 - Лёгкое создание генераторов
 - Мощная библиотека для работы с итераторами
- Встроенная в синтаксис поддержка парадигм функционального программирования
 - Функции высшего класса
 - Лямбда-функции
 - List comprehension

Python. Применение



- Применение Python:
 - Часто используется как основной язык программирования
 - Быстрое прототипирование
 - Разработка инструментов для системного администрирования
 - Веб-программирование
 - Встроенный скриптовый язык
- Python используют:
 - Как встроенный скриптовый язык
 - 3D редакторы: Maya, MotionBuilder, Softimage, Blender
 - 2D редакторы: GIMP, Inkscape, Scribus, and Paint Shop Pro
 - ГИС решение ESRI ArcGIS
 - Скриптование в играх: Civilization IV, EVE Online
 - YouTube — для системного многограммирования
 - Для решения широкого спектра задач: Google, Yahoo!, CERN, NASA