

Самые нужные функции Python

добавлено: 12:48 16 фев 2011 | добавил: Andre-4667 | Программирование для смартфонов

Многие спецы или просто знающие программисты прочитали не мало литературы при изучении языка программирования. И по началу, забывав некоторые функции, искали опять это в литературе. Так вот, для новичков собрал самые важные функции без которых в программах почти не обойтись 😊

```
# Модуль
abs()
# Сравнивает числа one и two [one two|1]
cmp(x, y)
# Возвращает минимальное число
min(a, b, c, ...)
# Возвращает максимальное число
max(a, b, c, ...)
# Преобразует число в целое
int()
# Преобразует целое число в дробное
float()
# Преобразует список и кортеж в строку
str()
# Преобразует строку и кортеж в список
list()
# Преобразует строку и список в кортеж
tuple()
# Преобразует в строку
repr()
# Преобразует строку в целый вид
eval()
# Получает информацию о переменной
type()
# Создает список от x до y, не включая y
range(x, y)
# Возвращает длину списка, кортежа, строки
len()
# Функция запроса [Func{code, пароль|number, целые числа|date, дата|time,
время|float, вещественные числа|query, вопрос}]
appuifw.query('Name', 'Func', 'Start Set')
# Функция оповещения [Func{info, информация|error, ошибка|conf, завершение}]
appuifw.note('Text', 'Func')
# Функция создания меню
appuifw.app.menu=[('Name', func)]
# Окно выбора варианта из списка
appuifw.popup_menu(list, 'name')
# Функция запроса 2-х строк в кортеж
appuifw.multi_query(name_1, name_2)
# Функция запроса варианта из списка
appuifw.selection_list()
# Функция запроса нескольких вариантов из списка
appuifw.multi_selection_list()
# Возвращает список шрифтов установленных на смартфоне
appuifw.available_fonts()
# Устанавливает подпись приложения
appuifw.app.title=""
# Устанавливает функцию на правую софт клавишу
appuifw.app.exit_key_handler=name
# Устанавливает функцию на левую софт клавишу
appuifw.app.menu_key_handler=name
# Устанавливает текст над правой софт клавишей
appuifw.app.exit_key_text=""
# Устанавливает текст над левой софт клавишей
appuifw.app.menu_key_text=""
# Цикл
```

```

1 | while condition:
2 |     body
3 | [else]:
4 |     body_2]

```

```

# Возвращает 1 если x ложно, иначе 0
not x
# Возвращает x, если x ложно, иначе у
x and y
# Возвращает у, если x ложно, иначе x
x or y
# Проверяет есть/нет ли в последовательности условия
in/not in
# Проверяет ссылаются/не ссылаются ли две переменные на одно условие
is x not is
# Цикл

```

```

1 | for a in b:
2 |     body
3 | [else]:
4 |     body_2]

```

```
# Добавление русского языка в программу через Lambda функцию
ru=lambda x: x.decode('u8')
```

```
# Функция пропуска
```

```
pass
```

```
# Функция исключения
```

```

1 | try:
2 |     body
3 | [except:
4 |     body_2]
5 | [finally:
6 |     body_2]

```

```
# Выводит список имен файлов и папок в папке
os.listdir()
```

```
# Создает папку
```

```
os.mkdir()
```

```
# Создает путь папок
```

```
os.makedirs()
```

```
# Удаляет файл
```

```
os.remove()
```

```
# Удаляет папку
```

```
os.rmdir()
```

```
# Удаляет путь папок (вместе с пустыми)
```

```
os.removedirs()
```

```
# Переименовывает папку из name в name_2
```

```
os.rename(name, name_2)
```

```
# Возвращает кортеж (Остальной путь, файл)
```

```
os.path.split()
```

```
# Возвращает кортеж (Имя диска, остальной путь)
```

```
os.path.splitdrive()
```

```
# Возвращает кортеж (Остальной путь, расширение файла)
```

```
os.path.splitext()
```

```
# Возвращает путь
```

```
os.path.dirname()
```

```
# Возвращает файл
```

```
os.path.basename()
```

```
# Объединяет путь
```

```
os.path.join()
```

```
# Возвращает 1 если если файл/папка существует, иначе 0
```

```
os.path.exists()
```

```
# Возвращает 1 если указан на файл, иначе 0
```

```
os.path.isfile()
```

```
# Возвращает 1 если указан на папку, иначе 0
```

```
os.path.isdir()
```

```
# Возвращает время последнего доступа к файлу/папке
```

```
os.path.getatime()
```

```
os.path.getatime()  
# Возвращает время изменения файла/папки  
os.path.getmtime()  
# Возвращает размер файла/папки  
os.path.getsize()  
# Возвращает файл [mode{r, для чтения|w, для записи|a, для дописывания}]  
open(name, mode)  
# Возвращает всё содержимое файла в виде строки  
open(name, 'r').read()  
# Возвращает всё содержимое файла в виде строки разделяя каждую строку  
# символом \n  
open(name, 'r').readline()  
# Возвращает всё содержимое файла в виде списка  
open(name, 'r').readlines()  
# Записывает строку в файл  
open(name, 'w').write()  
# Записывает строки из списка в файл  
open(name, 'w').writelines()  
# Усекает файл до заданного размера  
open(name, 'w').truncate()  
# Закрывает файл  
open(name, mode).close()  
# Преобразует строку чисел в список чисел  
str="  
list=[int(x) for x in str]  
# Преобразует список чисел в список сканкодов клавиш  
list=[]  
skancodes=[ord(str(x)) for x in list]  
# Преобразует список чисел в строку чисел  
list=[]  
b=""  
for x in list: b+=str(x)  
# Преобразует список сканкодов клавиш в список чисел  
list=[eval(chr(x)) for x in skancodes]
```

Цитата: Символы:
<p>1) \\" – Символ обратно косой черты; 2) \' – Символ одинарной кавычки; 3) \" – Символ двойной кавычки; 4) \\r – Символ возврата каретки; 5) \\t – Символ горизонтальной табуляции; 6) \\v – Символ вертикальной табуляции. 6) \\n – Символ перехода на новую строку.</p>