

Python Basics v2.0

модуль I

Условные конструкции

Сергей Колесник

Минск, WGU, 2016

Что было в прошлый раз?

```
python  
or not  
and is-not if  
elif else:  
bool is
```

Разминка

	X	bool(X)
X = str(False)		
X = None		
X = '0'[1:]		
X = '1' + '2'[-1]		
X = type(True)		
X = type(None)		
X = long(0)		
X = float(0.0)		

Разминка

	X	bool(X)
X = str(False)	False'	True
X = None		
X = '0'[1:]		
X = '1' + '2'[-1]		
X = type(True)		
X = type(None)		
X = long(0)		
X = float(0.0)		

Разминка

	X	bool(X)
X = str(False)	False'	True
X = None	None	False
X = '0'[1:]		
X = '1' + '2'[-1]		
X = type(True)		
X = type(None)		
X = long(0)		
X = float(0.0)		

Разминка

	X	bool(X)
X = str(False)	False'	True
X = None	None	False
X = '0'[1:]	'	False
X = '1' + '2'[-1]		
X = type(True)		
X = type(None)		
X = long(0)		
X = float(0.0)		

Разминка

	X	bool(X)
X = str(False)	False'	True
X = None	None	False
X = '0'[1:]	'	False
X = '1' + '2'[-1]	12'	True
X = type(True)		
X = type(None)		
X = long(0)		
X = float(0.0)		

Разминка

	X	bool(X)
X = str(False)	False'	True
X = None	None	False
X = '0'[1:]	'	False
X = '1' + '2'[-1]	12'	True
X = type(True)	bool	True
X = type(None)		
X = long(0)		
X = float(0.0)		

Разминка

	X	bool(X)
X = str(False)	False'	True
X = None	None	False
X = '0'[1:]	'	False
X = '1' + '2'[-1]	12'	True
X = type(True)	bool	True
X = type(None)	NoneType	True
X = long(0)		
X = float(0.0)		

Разминка

	X	bool(X)
X = str(False)	False'	True
X = None	None	False
X = '0'[1:]	'	False
X = '1' + '2'[-1]	12'	True
X = type(True)	bool	True
X = type(None)	NoneType	True
X = long(0)	0L	False
X = float(0.0)		

Разминка

	X	bool(X)
X = str(False)	False'	True
X = None	None	False
X = '0'[1:]	'	False
X = '1' + '2'[-1]	12'	True
X = type(True)	bool	True
X = type(None)	NoneType	True
X = long(0)	0L	False
X = float(0.0)	0.0	False?

Пример

Длина входной строки больше трех?

Input: Строка

Output: При длинной строке: печать ее самой. И определить, была ли строка напечатана.

Пример

Длина входной строки больше трех?

Input: Строка

Output: При длинной строке: печать ее самой. И определить, была ли строка напечатана.

```
printed = False
```

```
if len(raw_input()) > 3:  
    print 'Long string!'  
    printed = True
```

```
print 'Was printed: {}'.format(printed)
```

Практика

Определение названия месяца и времени года с проверкой корректности.

Input: Целое число — номер месяца.

Output: Название месяца и соответствующего времени года.

Практика

Определение названия месяца и времени года с проверкой корректности.

Input: Целое число — номер месяца.

Output: Название месяца и соответствующего времени года.

Сравнение с bool

```
sorting = request.get_param_as_bool('asc_order')  
# True, False
```

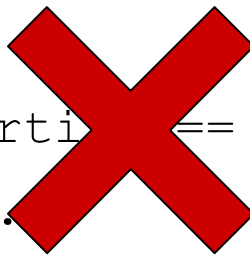

Сравнение с bool

```
sorting = request.get_param_as_bool('asc_order')  
# True, False  
  
if sorting == True:  
    ...
```

Сравнение с bool

```
sorting = request.get_param_as_bool('asc_order')  
# True, False
```

```
if sorting == True:  
    ..
```



```
if sorting is True:  
    ...
```

Сравнение с bool

```
sorting = request.get_param_as_bool('asc_order')  
# True, False
```

```
if sorting == True:  
    ..
```



```
if sorting is True:  
    ...
```

```
if sorting:  
    ...
```

Сравнение с bool

```
sorting = request.get_param_as_bool('asc_order')  
# True, False
```

Сравнение с bool

```
sorting = request.get_param_as_bool('asc_order')  
# True, False  
  
if sorting == False:  
    ...
```

Сравнение с bool

```
sorting = request.get_param_as_bool('asc_order')
```

```
# True, False
```

```
if sorting == False:
```

```
...
```

```
if sorting is False:
```

```
...
```

Сравнение с bool

```
sorting = request.get_param_as_bool('asc_order')
```

```
# True, False
```

```
if sorting == False:
```

```
...
```

```
if sorting is False:
```

```
...
```

```
if not sorting:
```

```
...
```

Практика

Определение поведения сортировки на основе переменной-параметра «`asc_order`», который может принимать значения `True`, `False` или `None`.

Input: -

Output:

- `desc`, когда параметр задан как `False`
- `asc`, когда параметр задан как `True` или не задан вообще

Условные выражения

Условные выражения

Через инструкцию:

```
if user.has_mouse():  
    with_mouse = 'yes'  
else:  
    with_mouse = 'no'
```

Условные выражения

Через выражение:

```
with_mouse = 'yes' if user.has_mouse() else 'no'
```

Условные выражения

- Общий синтаксис:

`<result-if-true> if <expression> else <result-if-false>`

Условные выражения

- Общий синтаксис:

```
<result-if-true> if <expression> else <result-if-false>
```

- Запись в несколько строчек

```
result = (  
    <result-if-true>  
    if <expression>  
    else <result-if-false>  
)
```

Оператор `not`

- Логическое отрицание
- Результат: противоположенное значение

not	False	True
	1	0

- Note: числа только для краткости. Реальный результат операций `not`, `and`, `or` — `True` или `False`.

Оператор and

- Логическое умножение
- Результат: True, только когда оба True

and	False	True
False	0	0
True	0	1

Оператор `or`

- Логическое сложение
- Результат: `False`, только когда оба `False`

or	False	True
False	0	1
True	1	1

Результат операции or

- Во что вычисляется $A \text{ or } B$?

Результат операции `or`

- Во что вычисляется `A or B`?
- Не в `bool`

Результат операции `or`

- Во что вычисляется `A or B`?
- Не в `bool`
- Либо в `A`, либо в `B`

Результат операции `or`

- `A or B or C or ... or Z`
- Вычисляется
 - в первый объект, приведение которого к `bool` дает `True`
 - либо в последний

Результат операции `or`

- `A or B or C or ... or Z`
- Вычисляется
 - в первый объект, приведение которого к `bool` дает `True`
 - либо в последний (когда все до него приводились в `False`)

Результат операции `or`

```
sorting = request.get_param_as_bool('order')  
sorting = sorting or <default>
```

Результат операции `or`

```
sorting = (  
    request.get_param_as_bool('order')  
    or <default>  
)
```

Результат операции and

- A and B and C and ... and Z
- Вычисляется
 - в первый объект, приведение которого к bool дает False
 - либо в последний

Результат операции and

- `A and B and C and ... and Z`
- Вычисляется
 - в первый объект, приведение которого к `bool` дает `False`
 - либо в последний (когда все до него приводились в `True`)

Результат операции `or`

```
sorting = request.get_param_as_bool('order')  
ascending = sorting and 'yes' or 'no'
```

Практика

Определение корней квадратного уравнения
 $ax^2 + bx + c = 0$

Input: Вещественные числа a , b , c —
коэффициенты.

Output:

- Один или два вещественных корня уравнения.
- Либо решение линейного уравнения.
- Либо сообщение о некорректных параметрах.

Практика

Оформить ситуацию по доставке обедов в python-коде.

Если обед привезут вовремя и он будет вкусным, то можно оставить хороший отзыв компании по доставке обедов. Если привезут не вовремя, то отзыв за транспортировку оставим негативный, и независимо от вкустности еды, не будем оставлять положительный про еду. Однако, если она нам не понравится, то в следующий раз не закажем еще.

Если вообще не привезут, то надо позвонить поставщику и поругаться на него. Если его реакция будет такова, что он извиняется и доставит обеды немедленно, то мы согласимся, но отзывов оставлять не будем. Во всех остальных случаях нужно поступить в соответствии с логичными побуждениями оставить отзыв хороший или нет или не оставлять вовсе.

Практика

Input:

- in_time — 'yes', 'no' или ''
- В зависимости от in_time:
 - provider_response — 'sorry', 'new', ''
 - is_tasty — 'yes' или 'no'

Output:

- delivery_review — 'ok', 'fail' или отзыва нет
- food_review — 'ok', 'fail' или отзыва нет
- order_again — 'yes' или 'no'

Спасибо за внимание